



## **BGP EVPN VXLAN Configuration Guide, Cisco IOS XE Amsterdam 17.2.x (Catalyst 9300 Switches)**

**First Published:** 2020-03-30

### **Americas Headquarters**

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

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

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

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

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

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

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### CHAPTER 1

#### **BGP EVPN VXLAN Overview 1**

BGP EVPN VXLAN 1

The Evolution of BGP EVPN VXLAN 1

Benefits of Deploying an Overlay-Underlay Architecture using BGP EVPN VXLAN 2

Fundamental Concepts of BGP EVPN VXLAN 2

VXLAN Overlay 2

Virtual Tunnel End Points 3

Overlay Multicast 4

Underlay 4

EVPN Control Plane 5

Route Target 5

EVPN Route Types 6

EVPN Instance 6

Ethernet Segment 6

EVPN Multihoming 6

Stretched VLAN and Subnet 6

Spine Leaf Architecture 7

Integrated Routing and Bridging 8

VXLAN Gateways 8

Layer 2 Virtual Network Instance 9

Layer 3 Virtual Network Instance 9

Mobility 9

---

### CHAPTER 2

#### **Configuring EVPN VXLAN Layer 2 Overlay Network 11**

Information About EVPN VXLAN Layer 2 Overlay Network 11

Broadcast, Unknown Unicast, and Multicast Traffic 11

Underlay Multicast	12
Ingress Replication	12
Flooding Suppression	13
How to Configure EVPN VXLAN Layer 2 Overlay Network	14
Configuring Layer 2 VPN EVPN on a VTEP	14
Configuring an EVPN Instance on the VLAN on a VTEP	17
Configuring the Loopback Interface on a VTEP	18
Configuring the Access-Facing Interface in the VLAN on a VTEP	19
Configuring the NVE Interface on a VTEP	20
Configuring BGP on a VTEP with EVPN Address Family	21
Configuration Examples for EVPN VXLAN Layer 2 Overlay Network	22
Verifying EVPN VXLAN Layer 2 Overlay Network	29

---

**CHAPTER 3**

<b>Configuring EVPN VXLAN Layer 3 Overlay Network</b>	<b>33</b>
Information About EVPN VXLAN Layer 3 Overlay Network	33
How to Configure EVPN VXLAN Layer 3 Overlay Network	34
Configuring an IP VRF on a VTEP	34
Configuring the Loopback Interface on a VTEP	36
Configuring the Core-facing VLAN on a VTEP	37
Configuring Access-facing VLAN on a VTEP	38
Configuring Switch Virtual Interface for the Core-facing VLAN	38
Configuring the Switch Virtual Interface for the Access-facing VLANs	39
Configuring the NVE Interface on a VTEP	40
Configuring BGP with IPv4 or IPv6 or Both Address Families on VTEP	41
Configuration Examples for EVPN VXLAN Layer 3 Overlay Network	44
Verifying EVPN VXLAN Layer 3 Overlay Network	51

---

**CHAPTER 4**

<b>Configuring EVPN VXLAN Integrated Routing and Bridging</b>	<b>53</b>
Restrictions for EVPN VXLAN Integrated Routing and Bridging	53
Information About EVPN VXLAN Integrated Routing and Bridging	53
EVPN VXLAN Distributed Anycast Gateway	54
Manual MAC Address Configuration	56
MAC Aliasing	56
EVPN VXLAN Centralized Default Gateway	56

Default Gateway MAC Address Assignment	59
How to Configure EVPN VXLAN Integrated Routing and Bridging	59
Configuring EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway	59
Configuring Layer 2 VPN EVPN on a VTEP	59
Configuring IP VRF on VTEP	60
Configuring the Loopback Interface on a VTEP	60
Configuring Core-facing and Access-facing VLANs on a VTEP	60
Configuring Switch Virtual Interface for the Core-facing VLAN on a VTEP	61
Configuring Switch Virtual Interface for the Access-facing VLANs on a VTEP	62
Configuring the NVE Interface on a VTEP	63
Configuring BGP with EVPN and VRF Address Families on a VTEP	64
Configuring EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway	67
Configuration Examples for EVPN VXLAN Integrated Routing and Bridging	68
Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway	68
Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway	82
Verifying EVPN VXLAN Integrated Routing and Bridging	92

**CHAPTER 5**

<b>Configuring Spine Switches in a BGP EVPN VXLAN Fabric</b>	<b>93</b>
Restrictions for Spine Switches in a BGP EVPN VXLAN Fabric	93
Information About Spine Switches in a BGP EVPN VXLAN Fabric	93
Deployment Scenarios for Spine Switches and Leaf Switches in a BGP EVPN VXLAN Fabric	94
Configuration Examples for Spine Switches in a BGP EVPN VXLAN Network	94
Configuration Example for Spine Switches Using iBGP when the Spine Switches and Leaf Switches are in the Same Autonomous System	95
Configuration Example for Spine Switches Using eBGP when the Spine Switches are in One Autonomous System and the Leaf Switches are in a Different Autonomous System	111
Configuration Example for Spine Switches Using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System	130

**CHAPTER 6**

<b>Configuring DHCP Relay in a BGP EVPN VXLAN Fabric</b>	<b>151</b>
Restrictions for DHCP Relay in a BGP EVPN VXLAN Fabric	151

Information About DHCP Relay in a BGP EVPN VXLAN Fabric	151
DHCP Relay on VTEPs	152
How to Configure DHCP Relay in a BGP EVPN VXLAN Fabric	153
Configuring DHCP Relay on a VTEP	153
Configuring DHCP Relay on the Access SVI of a VTEP	154
Configuring the Router Interface on the Border VTEP for DHCP Server Reachability	155
Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric	157

**CHAPTER 7****Configuring Tenant Routed Multicast 163**

Restrictions for Tenant Routed Multicast	163
Information about Tenant Routed Multicast	164
PIM-SM with Distributed Anycast-RP Mode	165
PIM-SSM Mode	167
How to Configure Tenant Routed Multicast	168
Configuring the Default Multicast Distribution Tree in the VRF	168
Configuring Multicast Routing on the Overlay VRF	169
Configuring Multicast on Switch Virtual Interfaces for the Core-facing and Access-facing VLANs	169
Configuring BGP with MVPN Address Family on VTEP	170
Configuring RP for Underlay Network	171
Configuring RP and SSM for Overlay Network	172
Configuration Examples for Tenant Routed Multicast	173
Verifying Tenant Routed Multicast	186

**CHAPTER 8****Configuring EVPN VXLAN External Connectivity 189**

Restrictions for EVPN VXLAN External Connectivity	189
Information About EVPN VXLAN External Connectivity	189
Implementation of Border Nodes for EVPN VXLAN External Connectivity	189
External Connectivity with Layer 3 Networks	191
External Connectivity with Layer 2 Networks	192
How to Configure EVPN VXLAN External Connectivity	193
Enabling Layer 3 External Connectivity with VRF-Lite	193
Configuring the VRF on the Border VTEP Interface that Faces the External Router	194
Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN	195

Configuring BGP on a Border VTEP for External Connectivity with MPLS Layer 3 VPN	196
Enabling Layer 2 External Connectivity with IEEE 802.1Q Networks	200
Enabling Layer 2 External Connectivity with a VPLS Network Through an Access VFI	201
Defining an Access VFI on a Border VTEP	202
Adding an Access VFI and an EVPN Instance as Members of the VLAN of a Border VTEP	203
Configuring VPLS on a Border VTEP	204
Configuration Examples for EVPN VXLAN External Connectivity	204
Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP	204
Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP	215

**CHAPTER 9****Troubleshooting BGP EVPN VXLAN 227**

Troubleshooting Scenarios for BGP EVPN VXLAN	227
Troubleshooting Broadcast, Unknown Unicast, Multicast Traffic Forwarding	228
Troubleshooting Unicast Forwarding Between VTEPs in the Same VLAN Through a Layer 2 VNI	232
Verifying the Provisioning of an EVPN VXLAN Layer 2 Overlay Network	233
Verifying Intra-Subnet Traffic Movement in an EVPN VXLAN Layer 2 Overlay Network	237
Troubleshooting Unicast Forwarding Between VTEPs in Different VLANs Through a Layer 3 VNI	244
Verifying the Provisioning of an EVPN VXLAN Layer 3 Overlay Network	245
Verifying Inter-Subnet Traffic Movement and Symmetric IRB in an EVPN VXLAN Layer 3 Overlay Network	250
Troubleshooting Unicast Forwarding Between a VXLAN Network and an IP Network	257
Verifying the Provisioning of an EVPN VXLAN Layer 3 Overlay Network	257
Verifying Traffic from a VXLAN Fabric to an IP Network Through a Border Leaf Switch Using Route Type 5	257

**CHAPTER 10****Feature History for BGP EVPN VXLAN 261**

Feature History for BGP EVPN VXLAN	261
------------------------------------	-----







# CHAPTER 1

## BGP EVPN VXLAN Overview

- [BGP EVPN VXLAN, on page 1](#)
- [The Evolution of BGP EVPN VXLAN, on page 1](#)
- [Benefits of Deploying an Overlay-Underlay Architecture using BGP EVPN VXLAN, on page 2](#)
- [Fundamental Concepts of BGP EVPN VXLAN, on page 2](#)

## BGP EVPN VXLAN

BGP EVPN VXLAN is a campus network solution for Cisco Catalyst 9000 Series Switches running Cisco IOS XE software. This solution is a result of proposed IETF standards and Internet drafts submitted by the BGP Enabled ServiceS (bess<sup>1</sup>) workgroup. It is designed to provide a unified overlay network solution and also address the challenges and drawbacks of existing technologies.

This chapter provides a background for the evolution of the solution and covers conceptual information and basic terminology that is required to understand BGP EVPN VXLAN. Later chapters of this configuration guide cover information about configuration, implementation, functionalities, and troubleshooting BGP EVPN VXLAN.

## The Evolution of BGP EVPN VXLAN

Traditionally, VLANs have been the standard method for providing network segmentation in campus networks. VLANs use loop prevention techniques such as Spanning Tree Protocol (STP), which impose restrictions on network design and resiliency. Further, there is a limitation with the number of VLANs that can be used to address layer 2 segments (4094 VLANs). Therefore, VLANs are a limiting factor for IT departments and cloud providers who build large and complex campus networks.

VXLAN is designed to overcome the inherent limitations of VLANs and STP. It is a proposed IETF standard [[RFC 7348](#)] to provide the same Ethernet Layer 2 network services as VLANs do, but with greater flexibility. Functionally, VXLAN is a MAC-in-UDP encapsulation protocol that runs as a virtual overlay on an existing Layer 3 network.

However, VXLAN by itself does not provide optimal switching and routing capabilities in a network because it uses a “flood and learn” mechanism that limits scalability. “Flood and learn” mechanism is where the host’s information is flooded across the network for it to be reachable. To provide optimal switching and routing capabilities, a VXLAN overlay requires:

- An underlying transport network that performs data plane forwarding to allow unicast communication between end points connected to the fabric.
- A control plane that is capable of distributing Layer 2 and Layer 3 host reachability information across the network.

To meet these additional requirements, Internet drafts submitted by the bess workgroup ([draft-ietf-bess-evpn-overlay-12](#)) proposed MP-BGP to carry Layer 2 MAC and Layer 3 IP information simultaneously. MP-BGP incorporates Network Layer Reachability Information (NLRI) to achieve this. With MAC and IP information available together for forwarding decisions, routing and switching within a network is optimized. This also minimizes the use of the conventional "flood and learn" mechanism used by VXLAN and allows for scalability in the fabric. EVPN is the extension that allows BGP to transport Layer 2 MAC and Layer 3 IP information. This deployment is called a BGP EVPN VXLAN fabric (also referred to as VXLAN fabric).

## Benefits of Deploying an Overlay-Underlay Architecture using BGP EVPN VXLAN

Deploying an overlay-underlay architecture using BGP EVPN VXLAN provides the following advantages:

- Scalability — VXLAN provides Layer 2 connectivity that allows for infrastructure that can scale to 16 million tenant networks. It overcomes the 4094-segment limitation of VLANs. This is necessary to address today's multi-tenant cloud requirements.
- Flexibility — VXLAN allows workloads to be placed anywhere, along with the traffic separation required, in a multitenant environment. The traffic separation is done by network segmentation using VXLAN segment IDs or VXLAN network identifiers (VNIs). Workloads for a tenant can be distributed across different physical devices but they are identified by their respective Layer 2 VNI or Layer 3 VNI.
- Mobility — Virtual machines can be moved from one location to another without updating spine switch tables. This is because entities within the same tenant VXLAN network retain the same VXLAN segment ID, regardless of their location.

## Fundamental Concepts of BGP EVPN VXLAN

This section provides information about the various fundamental concepts and terminologies that are involved in the working of BGP EVPN VXLAN.

### VXLAN Overlay

An overlay network is a virtual network that is built over an existing Layer 2 or Layer 3 network by forming a static or dynamic tunnel that runs on top of the physical network infrastructure. The existing Layer 2 or Layer 3 network is what forms the underlay and is covered further below in this chapter.

When a data packet is sent through an overlay, the original packet or frame is packaged or encapsulated at a source edge device with an outer header and dispatched toward an appropriate destination edge device. The intermediate network devices forward the packet based on the outer header but are not aware of the data in

the original packet. At the destination edge device, the packet is decapsulated by stripping off the overlay header and then forwarded based on the actual data within.

In the context of BGP EVPN VXLAN, VXLAN is used as the overlay technology to encapsulate the data packets and tunnel the traffic over a Layer 3 network. VXLAN creates a Layer 2 overlay network by using a MAC-in-UDP encapsulation. A VXLAN header is added to the original Layer 2 frame and it is then placed within a UDP-IP packet. A VXLAN overlay network is also called as a VXLAN segment. Only host devices and virtual machines within the same VXLAN segment can communicate with each other.

#### **VXLAN Network Identifier**

Each VXLAN segment is identified through a 24-bit segment ID, termed the VXLAN network identifier. This ensures that up to 16 million VXLAN segments can be present within the same administrative domain.

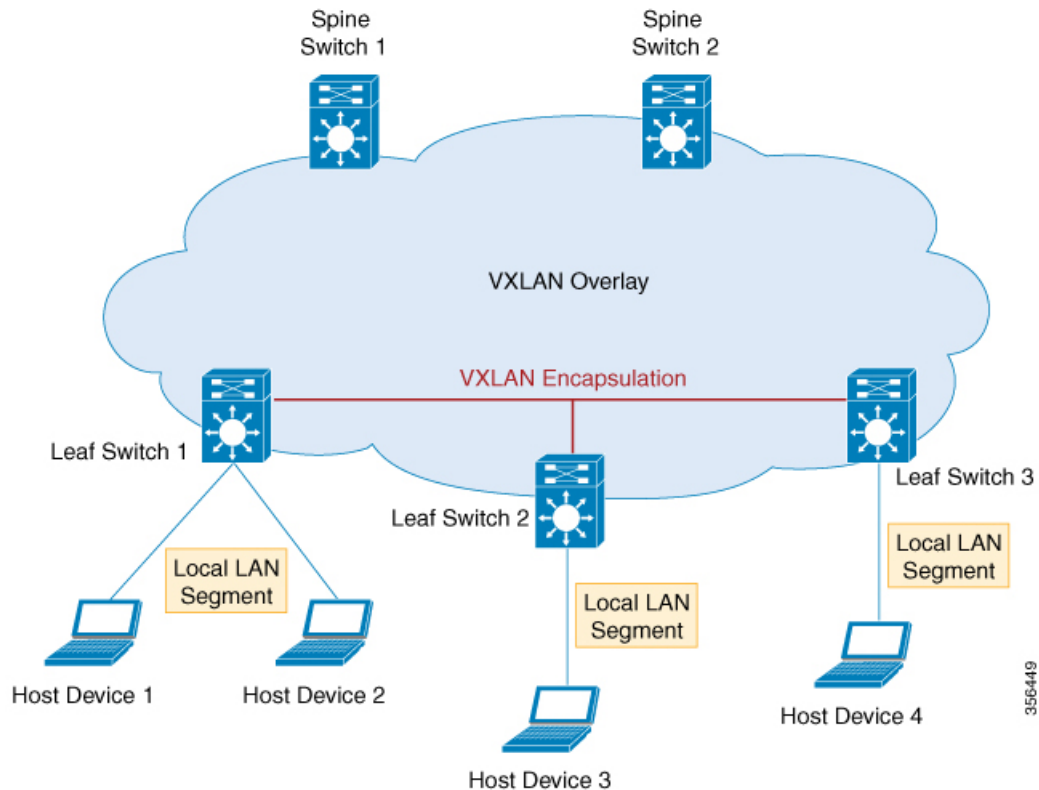
## **Virtual Tunnel End Points**

Every VXLAN segment has tunnel edge devices known as Virtual Tunnel End points (VTEPs). These devices sit at the edge of the VXLAN network and are responsible for creating instances of VXLAN tunnels, and for performing VXLAN encapsulation and decapsulation.

A VTEP has a switch interface on the local LAN segment to support local endpoint communication through bridging, and an IP interface to interact with the transport IP network.

The IP interface has a unique IP address that identifies the VTEP on the transport IP network. The VTEP uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface. A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC address-to-VTEP mappings through its IP interface.

The following figure illustrates the working of an overlay VXLAN network connecting various VTEPs:



## Overlay Multicast

Overlay multicast is the method by which a overlay network forwards multicast traffic between various VTEPs present in the network. Tenant Routed Multicast (TRM) provides a mechanism to efficiently forward multicast traffic in a VXLAN overlay network. TRM is a BGP-EVPN based solution that enables multicast routing between sources and receivers connected on VTEPs in VXLAN fabric.

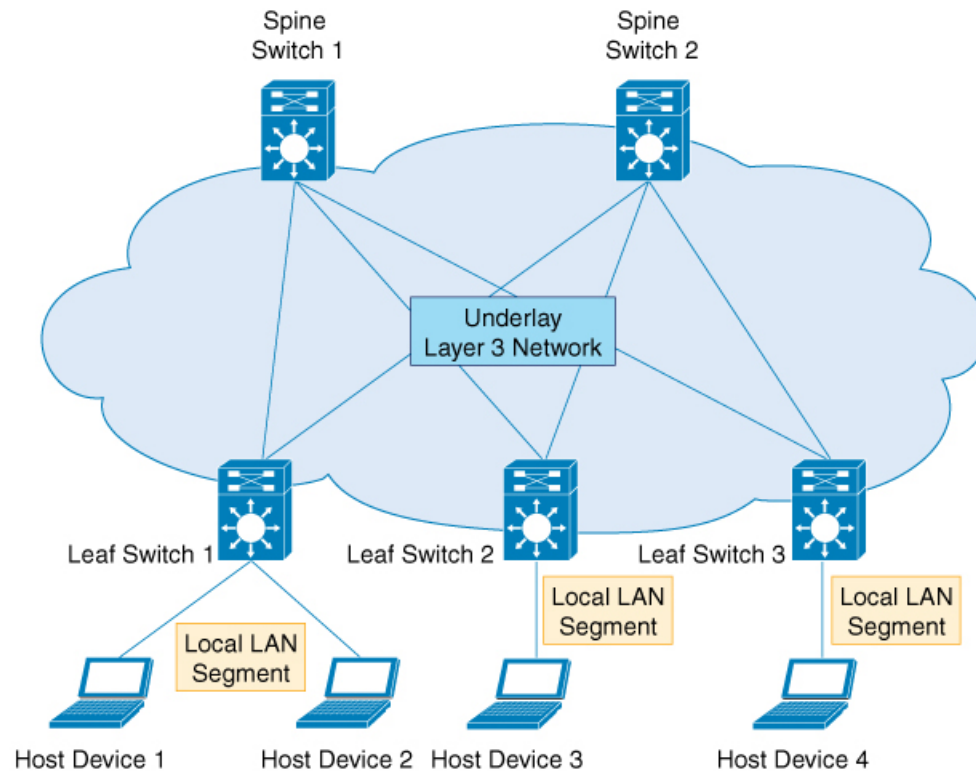
Without TRM, the multicast traffic is sent as part of the underlay network in the form of BUM traffic either using underlay multicast or ingress replication methods. This does not allow sources and receivers that are present across different subnets to communicate with each other. Using TRM, multicast communication is moved out of the BUM underlay traffic. This enables multicast communication in the overlay network irrespective of the subnet in which the source or the receiver resides.

## Underlay

An underlay network is the physical network over which the virtual overlay network is established. Once the overlay network is defined along with the data-plane encapsulation, a method of transport is required to move the data across the physical network underneath. This method of transport is typically an underlay transport network, or simply the underlay.

In BGP EVPN VXLAN, the underlay Layer 3 network transports the VXLAN-encapsulated packets between the source and destination VTEPs and provides reachability between them. The VXLAN overlay and the underlying IP network between the VTEPs are independent of each other.

The following figure illustrates an underlay network:



356449

## EVPN Control Plane

The overlay requires a mechanism to know which end host device is behind which overlay edge device. VXLAN natively operates on a flood and learn mechanism where broadcast, unknown unicast and multicast (BUM) traffic in a given VXLAN network is sent over the IP core to every VTEP that has membership in that network. IP multicast is used to send traffic over the network. The receiving VTEPs decapsulate the packet and, based on the inner frame, perform Layer 2 MAC learning. The inner source MAC address is learned against the outer source IP address corresponding to the source VTEP. In this way, reverse traffic is unicasted toward the previously learnt end host.

The drawback of the flood and learn mechanism is that it does not allow scalability in a VXLAN network. In order to address this issue, a control plane is used to manage the MAC address learning and VTEP discovery. In BGP EVPN VXLAN deployments, Ethernet Virtual Private Network (EVPN) is used as the control plane. EVPN control plane provides the capability to exchange both MAC address and IP address information. EVPN uses Multi Protocol Border Gateway Protocol (MP-BGP) as the routing protocol to distribute reachability information pertaining to the VXLAN overlay network, including endpoint MAC addresses, endpoint IP addresses, and subnet reachability information. BGP EVPN distribution protocol facilitates the mapping information to be built by the tunnel edge devices in the location-identity mapping database.

## Route Target

A route target is an extended attribute in EVPN route updates that controls route distribution in a multi-tenant network. EVPN VTEPs have an import route target setting and an export route target setting for every VRF and Layer 2 Virtual Network Instance (VNI). When a VTEP advertises EVPN routes, it affixes its export route target in the route update. These routes are received by the other VTEPs in the network. The receiving

VTEPs compare the route target value carried with the route against their own local import route target setting. If the two values match, the route is accepted and programmed in the routing table. Otherwise, the route is not imported.

## EVPN Route Types

The EVPN control plane advertises the following types of information:

- Route type 1 – This is an Ethernet Auto-Discovery (EAD) route type used to advertise Ethernet segment identifier, Ethernet Tag ID, and EVPN instance information. EAD route advertisements may be sent for each EVPN instance or for each Ethernet segment.
- Route type 2 – This advertises endpoint reachability information, including MAC and IP addresses of the endpoints or VTEPs.
- Route type 3 – This performs multicast router advertisement, announcing the capability and intention to use ingress replication for specific VNIs.
- Route type 4 – This is an Ethernet Segment route used to advertise the Ethernet segment identifier, IP address length, and the originating router's IP address.
- Route type 5 – This is an IP prefix route used to advertise internal IP subnet and externally learned routes to a VXLAN network.

## EVPN Instance

An EVPN Instance (EVI) represents a Virtual Private Network (VPN) on a VTEP. It is the equivalent of IP VRF in Layer 3 VPN and is also known as a MAC VRF.

## Ethernet Segment

An Ethernet segment is associated with an access-facing interface of a VTEP and represents the connection with a host device. Each Ethernet segment is assigned a unique value known as Ethernet segment identifier (ESI). When a host device is connected to more than one VTEPs, then the ESI for these connections remains the same.

## EVPN Multihoming

EVPN multihoming allows you to connect a Layer 2 device or an end host device to more than one leaf switch in the VXLAN network. This provides redundancy and allows network optimization over single-homed topologies where the customer network is connected to a single leaf switch. Redundancy in the connection with the leaf switches ensures that there is no traffic disruption when there is a network failure. Multihomed topologies are more resilient, secure and efficient than single-homed topologies. EVPN multihoming operates in single-active and all-active redundancy modes.

## Stretched VLAN and Subnet

By running over the existing networking infrastructure, EVPN VXLAN provides a means to stretch a Layer 2 network. EVPN VXLAN overlay allows Layer 2 segments and broadcast domains to be extended across sites or campus buildings over a Layer 3 core network. Layer 2 extension with EVPN VXLAN simplifies end user IP address management and provides seamless mobility in large campus networks.

## Spine Leaf Architecture

Spine-leaf architecture is a two-layer network topology where one layer is composed of leaf switches and the other layer has one or more spine switches. This design connects all the leaf switches by providing multiple paths through the various spine switches.

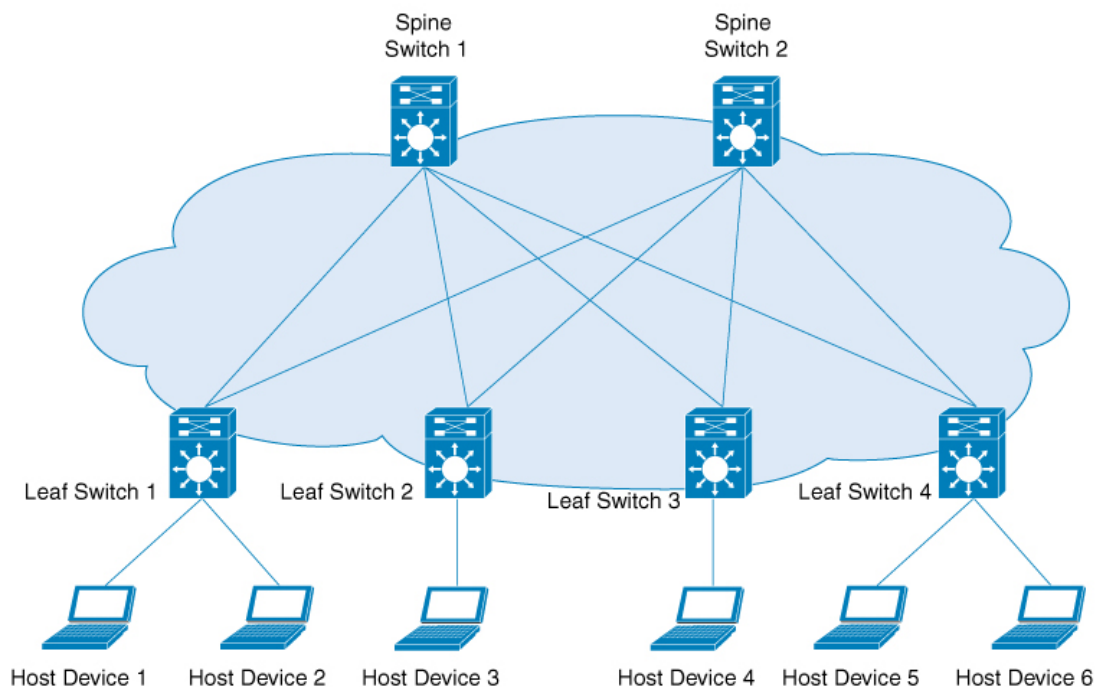
### Spine Switch

Spine switches are the connecting nodes between all the leaf switches. They forward the traffic between the leaf switches and are unaware of the endpoint addresses. By providing multiple paths to connect the leaf switches, spine switches provide redundancy to the network.

### Leaf Switch

Leaf switches are the nodes that are connected to the host or access devices. As a leaf switch sits on the edge of the network, it is also called as an edge or Network Virtualization Edge (NVE). When a host device on one leaf switch tries to communicate with a host device on another leaf switch, the traffic between the leaf switches is sent through a spine switch. Leaf switches function as VTEPs in a VXLAN network and perform the encapsulation and decapsulation.

The following image shows a typical spine-leaf topology where four leaf switches are connected through two spine switches:

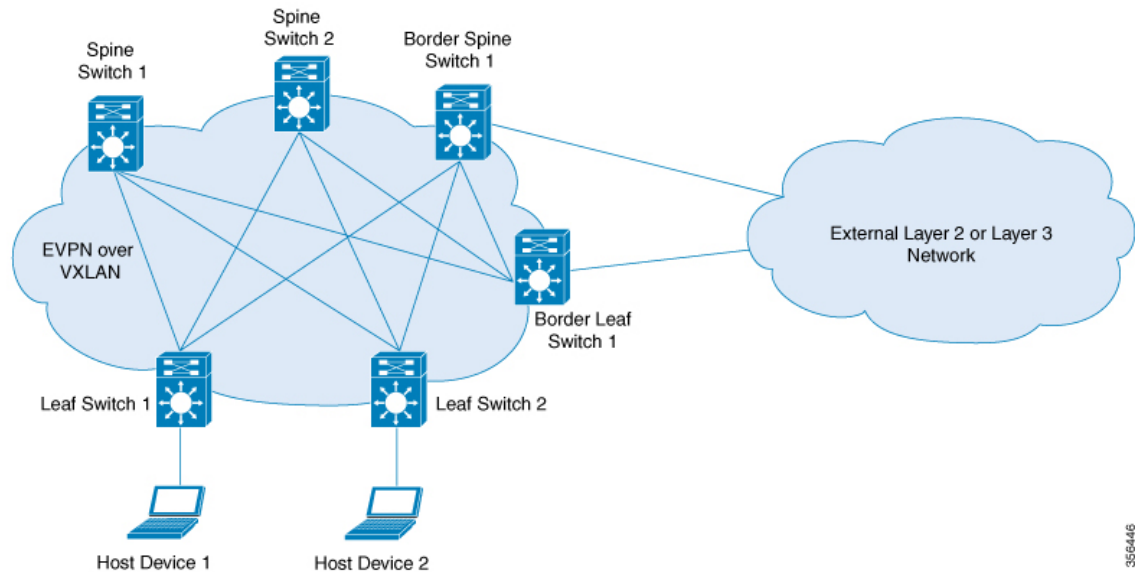


356448

### Border Spine Switch and Border Leaf Switch

External connectivity of the VXLAN fabric with other Layer 2 and Layer 3 networks is facilitated through nodes known as border nodes. If the border functionality is established through a spine switch, it is known as a border spine switch. If it is established through a leaf switch, it is known as a border leaf switch.

The following image shows a spine-leaf topology with one border leaf switch and one border spine switch connecting the fabric with an external network:



356446

## Integrated Routing and Bridging

EVPN VXLAN supports Integrated Routing and Bridging (IRB) functionality which allows the VTEPs in a VXLAN network to forward both Layer 2 (bridged) and Layer 3 (routed) traffic. When a VTEP forwards Layer 2 traffic, it is said to be performing bridging. Similarly, when a VTEP forwards Layer 3 traffic, it is said to be performing routing. The traffic between different subnets is forwarded through the VXLAN gateways. IRB is implemented in two ways:

- Asymmetric IRB
- Symmetric IRB

For more information about IRB, see [Information About EVPN VXLAN Integrated Routing and Bridging, on page 53](#) section.

## VXLAN Gateways

A VXLAN Gateway is an entity in the network that forwards traffic between VXLAN segments, or from a VXLAN environment to a non-VXLAN environment. Leaf switches in a VXLAN network can function as both Layer 2 and Layer 3 VXLAN gateways.

Layer 2 VXLAN gateways forward traffic within the same VLAN. Layer 2 VXLAN gateways allow VXLAN to VLAN bridging by mapping a VNI segment to a VLAN.

Layer 3 VXLAN gateways forward traffic to a different VLAN. Layer 3 VXLAN gateways allow both VXLAN to VXLAN routing as well as VXLAN to VLAN routing. VXLAN to VXLAN routing provides Layer 3 connectivity between two VNIs whereas VXLAN to VLAN routing provides connectivity between a VNI and a VLAN.



## Layer 2 Virtual Network Instance

The creation of a VXLAN overlay network allows host devices connected to various leaf nodes, that are separated by multiple Layer 3 networks, to interact as if they were connected to a single Layer 2 network, which is the VXLAN segment. This logical Layer 2 segment is called as Layer 2 VNI. The traffic that flows through a Layer 2 VNI between two VLANs within the same subnet is known as bridged traffic.

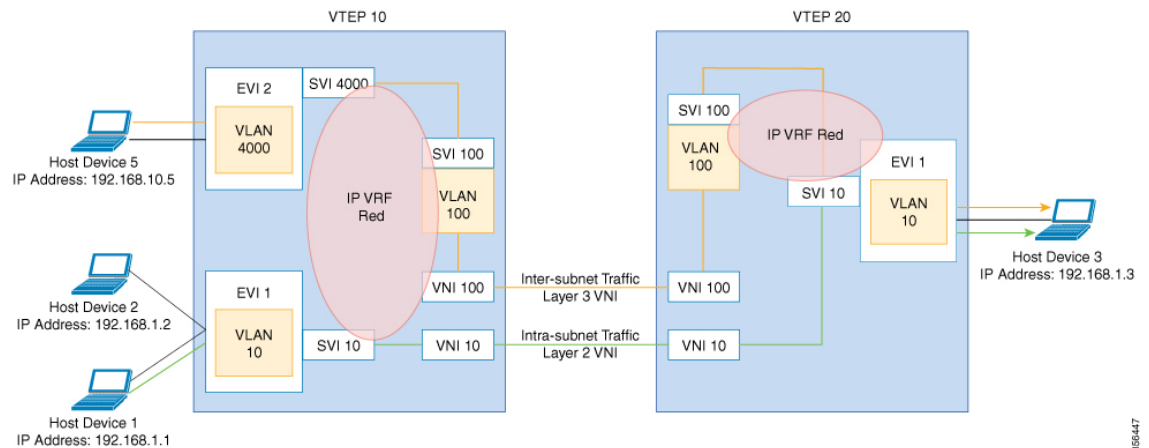
A VLAN that is locally defined on a VTEP can be mapped to a Layer 2 VNI. In order to allow host devices to connect to a Layer 2 VNI, the connected VLAN must be mapped to the Layer 2 VNI, and then the Layer 2 VNI is associated with the Network Virtualization Edge (NVE) logical interface on the VTEP.

## Layer 3 Virtual Network Instance

When endpoints connected to a Layer 2 VNI need to communicate with endpoints belonging to different IP subnets, they send the traffic to their default gateway. Communication between endpoints belonging to different Layer 2 VNIs is possible only through a Layer 3 routing function. In an EVPN VXLAN deployment, the various Layer 2 segments that are defined by combining the local VLANs and the global Layer 2 VNIs can be associated to a VRF in order to communicate.

A Layer 3 VNI facilitates Layer 3 segmentation for every VRF on a VTEP. This is done by mapping each VRF instance to a unique Layer 3 VNI in the network and associating the various Layer 2 VNIs for a VTEP to the same VRF. This allows inter- VXLAN communication throughout the Layer 3 VNI within a particular VRF instance. The use of VRFs to enable a logical Layer 3 isolation is known as multi-tenancy. The traffic that flows through a Layer 3 VNI between two VLANs in different subnets is known as routed traffic.

The following image shows the movement of traffic between host devices in same and different subnets through Layer 2 and Layer 3 VNIs:



## Mobility

The identity of an endpoint in the BGP EVPN control plane is derived from its MAC address and IP address, and BGP EVPN provides a mechanism to support endpoint mobility within a VXLAN overlay.

RFC 7432 defines the scope of endpoint mobility within the VXLAN fabric.

### MAC Mobility and Duplicate MAC Detection

A MAC move occurs when an endpoint (or host) moves from one port to another. The new port may be within the same VTEP, or in a different VTEP, in the same VLAN. The BGP EVPN control plane resolves such moves by advertising MAC routes (EVPN route type 2). When an endpoint's MAC address is learned on a new port, the new VTEP it is in advertises (on the BGP EVPN control plane) that it is the local VTEP for the host. All other VTEPs receive the new MAC route.

A host may move several times, causing the corresponding VTEPs to advertise as many MAC routes. There may also be a delay between the time a new MAC route is advertised and when the old route is withdrawn from the route tables of other VTEPs, resulting in two locations briefly having the same MAC route. Here, a MAC mobility sequence number helps decide the most current of the MAC routes.

When the host MAC address is learned for the first time, the MAC mobility sequence number is set to 0. The value 0 indicates that the MAC address has not had a mobility event, and the host is still at the original location. If a MAC mobility event is detected, a new Route type 2 (MAC or IP advertisement) is added to the BGP EVPN control plane by the new VTEP below which the endpoint moved (its new location). Every time the host moves, the VTEP that detects its new location increments the sequence number by 1 and then advertises the MAC route for that host on the BGP EVPN control plane. On receiving the MAC route at the old location (VTEP), the old VTEP withdraws the old route.

A case may arise in which the same MAC address is simultaneously learned on two different ports. The EVPN control plane detects this condition and alerts the user that there is a duplicate MAC. The duplicate MAC condition may be cleared either by manual intervention, or automatically when the MAC address ages out on one of the ports.

### IP Mobility and Duplicate IP Detection

BGP EVPN supports IP mobility in a similar manner to the way it supports MAC mobility. The principal difference is that an IP move is detected when the IP address is learned on a different MAC address, regardless of whether it was learned on the same port or a different port. A duplicate IP address is detected when the same IP address is simultaneously learned on two different MAC addresses, and the user is alerted when this occurs.



## CHAPTER 2

# Configuring EVPN VXLAN Layer 2 Overlay Network

---

- [Information About EVPN VXLAN Layer 2 Overlay Network, on page 11](#)
- [How to Configure EVPN VXLAN Layer 2 Overlay Network, on page 14](#)
- [Configuration Examples for EVPN VXLAN Layer 2 Overlay Network, on page 22](#)
- [Verifying EVPN VXLAN Layer 2 Overlay Network, on page 29](#)

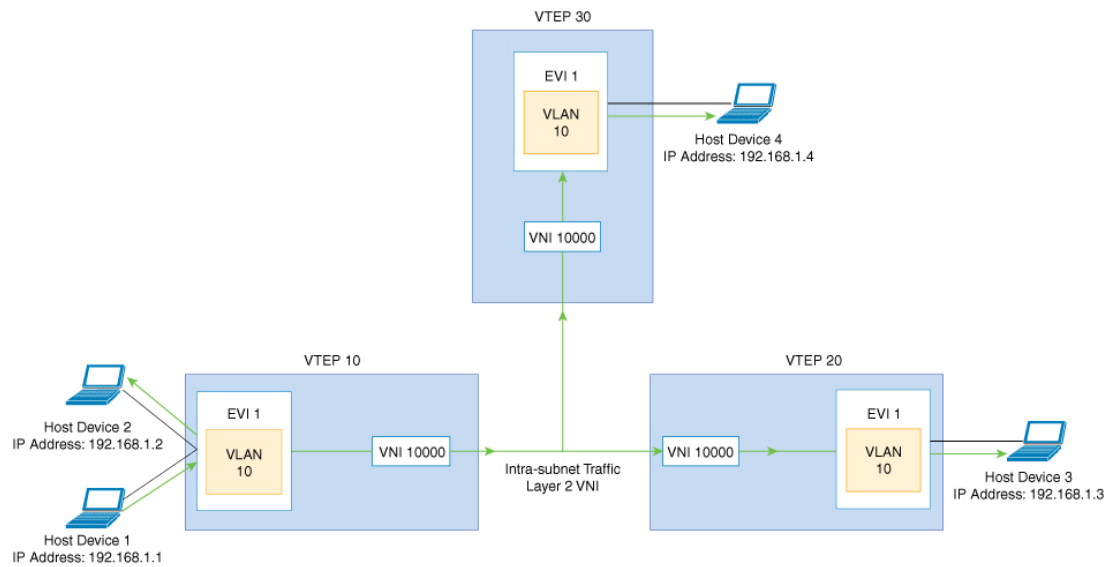
## Information About EVPN VXLAN Layer 2 Overlay Network

An EVPN VXLAN Layer 2 overlay network allows host devices in the same subnet to send bridged or Layer 2 traffic to each other. The network forwards the bridged traffic using a Layer 2 virtual network instance (VNI).

### Broadcast, Unknown Unicast, and Multicast Traffic

Multidestination Layer 2 traffic in a VXLAN network is typically referred to as broadcast, unknown unicast, and multicast (BUM) traffic. In a BGP EVPN VXLAN fabric, the underlay network forwards the BUM traffic to all the endpoints connected to a common Layer 2 broadcast domain in the VXLAN overlay.

The following image shows the flow of BUM traffic through a Layer 2 VNI. The network forwards BUM traffic from host device 1 to all the VTEPs which in turn send the traffic to all the host devices in the same subnet.



The MP-BGP EVPN control plane uses two different methods to forward BUM traffic in a VXLAN network:

- Underlay Multicast
- Ingress Replication

## Underlay Multicast

In underlay multicast, the underlay network replicates the traffic through a multicast group. Forwarding BUM traffic using underlay multicast requires the configuration of IP multicast in the underlay network. A single copy of the BUM traffic moves from the ingress or source VTEP towards the underlay transport network. The network forwards this copy along the multicast tree so that it reaches all egress or destination VTEPs participating in the given multicast group. Various branch points in the network replicate the copy as it travels along the multicast tree. The branch points replicate the copy only if the receivers are part of the multicast group associated with the VNI.

BUM traffic forwarding through underlay multicast is achieved by mapping a Layer 2 VNI to the multicast group. This mapping must be configured on all the VTEPs associated with the Layer 2 VNI. When a VTEP joins the multicast group, it receives all the traffic that is forwarded on that group. If the VTEP receives traffic in a VNI that is not associated with it, it simply drops the traffic. This approach maintains a single link within the network, thus providing an efficient way to forward BUM traffic.

## Ingress Replication

Ingress replication, or headend replication, is a unicast approach to handle multdestination Layer 2 overlay BUM traffic. Ingress replication involves an ingress device replicating every incoming BUM packet and sending them as a separate unicast to the remote egress devices. Ingress replication happens through EVPN route type 3, also called as inclusive multicast ethernet tag (IMET) route. BGP EVPN ingress replication uses IMET route for auto-discovery of remote peers in order to set up the BUM tunnels over VXLAN. Using ingress replication to handle BUM traffic can result in scaling issues as an ingress device needs to replicate the BUM traffic as many times as there are VTEPs associated with the Layer 2 VNI.

### Ingress Replication Operation

IMET routes carry the remote or egress VNIs advertised from the remote peers, which can be different from the local VNI. The network creates a VXLAN tunnel adjacency when an ingress device receives IMET ingress replication routes from remote NVE peers. The tunnel adjacency is a midchain adjacency which contains IP or UDP encapsulation for the VXLAN Tunnel. If there is more than one VNI along the tunnel, then multiple VNIs share the tunnel. Ingress replication on EVPN can have multiple unicast tunnel adjacencies and different egress VNIs for each remote peer.

The network builds a flooded replication list with the routes advertised by each VTEP. The dynamic replication list stores all the remote destination peers discovered on a BGP IMET route in the same Layer 2 VNI. The replication list gets updated every time you configure the Layer 2 VNI at a remote peer. The network removes the tunnel adjacency and VXLAN encapsulation from the replication list every time a remote NVE peer withdraws the IMET ingress replication route. The network deletes the tunnel adjacency when there is no NVE peer using it.

Any BUM traffic that reaches the ingress device gets replicated after the replication list is built. The ingress device forwards the replicated traffic throughout the network to all the remote peers in the same VNI.

## Flooding Suppression

EVPN allows the distribution of the binding between IPv4 or IPv6 addresses and MAC addresses among the VTEPs of the network. It distributes the MAC-IP binding among all the VTEPs that participate in the EVPN instance associated with the MAC-IP routes. The MAC address associated with the IPv4 or IPv6 addresses is locally known even though it is learned from a remote VTEP. Locally connected endpoints send an Address Resolution Protocol (ARP) or an IPv6 neighbor discovery request when they look for a remote endpoint. The MAC-IP binding distribution allows a VTEP to perform a lookup in the local cache when it receives an ARP or an IPv6 neighbor discovery request. If the MAC-IP address information for the remote end point is available, the VTEP can use this information to avoid flooding the ARP request. If the MAC or IP address information for the remote end point is not available, the request floods throughout the fabric.

Flooding suppression avoids the flooding of ARP and IPv6 neighbor discovery packets over the EVPN VXLAN network. It suppresses the flooding to both the local and remote host or access devices. The network suppresses the flooding by implementing an ARP or neighbor discovery relay. This is achieved by using the known MAC address for the specified IPv4 or IPv6 address to convert broadcast and multicast requests to unicast requests. Flooding suppression is enabled by default on an EVPN-enabled VLAN. An EVPN VXLAN network suppresses the flooding for the following types of traffic:

### ARP Flooding Suppression

VTEPs send ARP requests as broadcast packets. ARP requests represent a large percentage of Layer 2 broadcast traffic. Flooding suppression converts them to unicast packets and reduces the network flood.

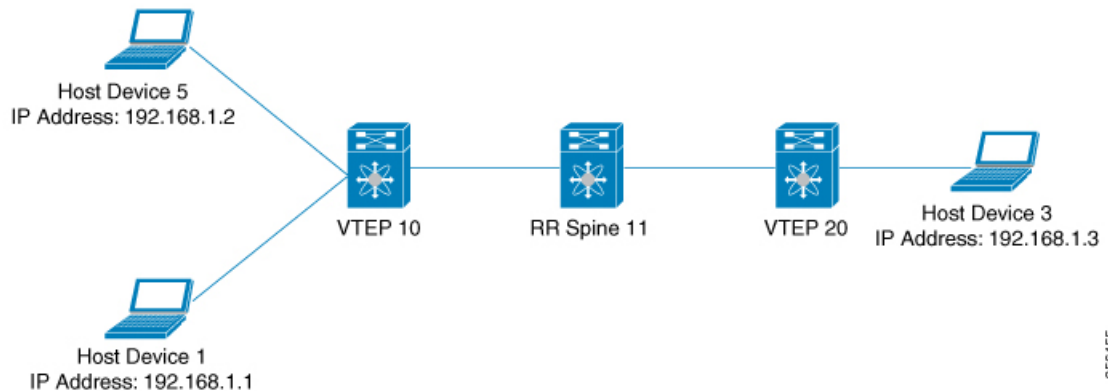
### IPv6 Neighbor Discovery Flooding Suppression

The IPv6 neighbor discovery process enables the discovery of a neighbor and helps the peers to determine each other's link-layer addresses. It also verifies the reachability of a neighbor and tracks the neighboring routers. IPv6 neighbor discovery uses Internet Control Message Protocol (ICMP) messages and solicited-node multicast addresses to achieve these functions.

Flooding suppression suppresses all multicast neighbor solicitation packets among Internet Control Message Protocol version 6 (ICMPv6) packets.

# How to Configure EVPN VXLAN Layer 2 Overlay Network

The following figure shows a sample topology of an EVPN VXLAN Network. Host device 1 and host device 3 are part of the same subnet. The network forwards BUM traffic from host device 1 to host device 3 using a Layer 2 VNI through either underlay multicast or ingress replication methods.



**Note** In a two-VTEP topology, a spine switch is not mandatory. For information about configuration of spine switches in an EVPN VXLAN network, see *Configuring Spine Switches in a BGP EVPN VXLAN Fabric* module.

Perform the following set of procedures to configure an EVPN VXLAN Layer 2 overlay network and forward the BUM traffic:

- Configure Layer 2 VPN EVPN on the VTEPs.
- Configure an EVPN instance in the VLAN on the VTEPs.
- Configure the access-facing interface in the VLAN on the VTEPs.
- Configure the loopback interface on the VTEPs.
- Configure the network virtualization endpoint (NVE) interface on the VTEPs.
- Configure BGP with EVPN address family on the VTEPs.
- Configure underlay multicast, if the specified replication type is static. For more information, see *IP Multicast Routing Configuration Guide*.

## Configuring Layer 2 VPN EVPN on a VTEP

To configure the Layer 2 VPN EVPN parameters on a VTEP, perform the following steps:

## Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>l2vpn evpn</b> <b>Example:</b> Device (config)# <b>l2vpn evpn</b>	Enters EVPN configuration mode.
<b>Step 4</b>	<b>replication-type {ingress   static}</b> <b>Example:</b> Device (config-evpn)# <b>replication-type static</b>	Configures the Layer 2 VPN EVPN replication type. <b>Note</b> Configure the Layer 2 VPN EVPN replication type as static, if multicast is enabled in the underlay network for EVPN BUM traffic.  When the Layer 2 VPN EVPN replication type is configured as static, the IMET route is not advertised and forwarding of BUM traffic relies on underlay multicast being configured on each VTEP.
<b>Step 5</b>	<b>router-id loopback-interface-id</b> <b>Example:</b> Device (config-evpn)# <b>router-id loopback0</b>	Specifies the interface that will supply the IP addresses to be used in auto-generating route distinguishers.
<b>Step 6</b>	<b>default-gateway advertise</b> <b>Example:</b> Device (config-evpn)# <b>default-gateway advertise</b>	(Optional) Enables default gateway advertisement on the switch. To configure distributed anycast gateway in a VXLAN network using MAC aliasing, enable default gateway advertisement on all the leaf switches in the network.  This command is applicable in integrated routing and bridging (IRB) scenarios where Layer 2 and Layer 3 VNI coexist in a VRF. Refer to <i>Configuring EVPN VXLAN Integrated Routing and Bridging</i> module for more details.  This command is mandatory only if the same MAC address is not manually configured on all the access SVIs.

	Command or Action	Purpose
		<b>Note</b> Use the <b>default-gateway advertise {enable   disable}</b> command in EVPN instance configuration mode to override the global default gateway advertisement settings and enable or disable it for a specific EVPN instance.
<b>Step 7</b>	<b>logging peer state</b> <b>Example:</b> Device(config-evpn) # <b>logging peer state</b>	(Optional) Displays syslog message when the first route is received or the last route is withdrawn from a given remote VTEP.
<b>Step 8</b>	<b>mac duplication limit limit-number time time-limit</b> <b>Example:</b> Device(config-evpn) # <b>mac duplication limit 20 time 5</b>	(Optional) Changes parameters for detecting duplicate MAC addresses.
<b>Step 9</b>	<b>ip duplication limit limit-number time time-limit</b> <b>Example:</b> Device(config-evpn) # <b>ip duplication limit 20 time 5</b>	(Optional) Changes parameters for detecting duplicate IP addresses.
<b>Step 10</b>	<b>route-target auto vni</b> <b>Example:</b> Device(config-evpn) # <b>route-target auto vni</b>	(Optional) Specifies to use VNI instead of EVPN instance number to auto-generate route target.
<b>Step 11</b>	<b>exit</b> <b>Example:</b> Device(config-evpn) # <b>exit</b>	Exits EVPN configuration mode and enters global configuration mode.
<b>Step 12</b>	<b>l2vpn evpn instance evpn-instance-number vlan-based</b> <b>Example:</b> Device(config) # <b>l2vpn evpn instance 1 vlan-based</b>	Configures a VLAN based EVPN instance in Layer 2 VPN configuration mode.  An EVPN instance needs to be explicitly configured only when something needs to be configured per EVPN instance such as a route target.
<b>Step 13</b>	<b>encapsulation vxlan</b> <b>Example:</b> Device(config-evpn-evi) # <b>encapsulation vxlan</b>	(Optional) Defines the encapsulation format as VXLAN.  The encapsulation format is VXLAN by default.
<b>Step 14</b>	<b>replication-type {ingress   static}</b> <b>Example:</b>	(Optional) Sets the replication type for the EVPN instance.



	Command or Action	Purpose
	Device (config-evpn-evi) # <b>replication-type ingress</b>	In case a global replication type has already been configured, this overrides the global setting.
<b>Step 15</b>	<b>default-gateway advertise {enable   disable}</b>  <b>Example:</b> Device (config-evpn-evi) # <b>default-gateway advertise disable</b>	(Optional) Enables or disables the default gateway advertisement for the EVPN instance.  In case default gateway advertisement has already been globally configured, this overrides the global setting.  This command is mandatory only if the same MAC address is not manually configured on all the access SVIs.  To configure distributed anycast gateway in a VXLAN network using MAC aliasing, enable default gateway advertisement on all the leaf switches in the network.
<b>Step 16</b>	<b>ip local-learning {enable   disable}</b>  <b>Example:</b> Device (config-evpn-evi) # <b>ip local-learning disable</b>	(Optional) Enables or disables local IP address learning for the specified EVPN instance.  In case IP address learning has already been globally configured, this overrides the global setting.
<b>Step 17</b>	<b>no auto-route-target</b>  <b>Example:</b> Device (config-evpn-evi) # <b>no auto-route-target</b>	(Optional) Disables auto generation of route targets.
<b>Step 18</b>	<b>rd rd-value</b>  <b>Example:</b> Device (config-evpn-evi) # <b>rd 65000:100</b>	(Optional) Configures a route distinguisher manually.
<b>Step 19</b>	<b>route-target {import   export   both} rt-value</b>  <b>Example:</b> Device (config-evpn-evi) # <b>route-target both 65000:100</b>	(Optional) Configures route targets manually.  <b>Note</b> Configure route targets manually if the auto-generated route target values (ASN:EVI or ASN:VNI) are different between the VTEPs.
<b>Step 20</b>	<b>end</b>  <b>Example:</b> Device (config-evpn-evi) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring an EVPN Instance on the VLAN on a VTEP

To configure an EVPN instance on the VLAN on a VTEP, perform the following steps:

**Procedure**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>vlan configuration <i>vlan-id</i></b> <b>Example:</b> Device(config)# <b>vlan configuration 11</b>	Enters VLAN feature configuration mode for the specified VLAN interface.
<b>Step 4</b>	<b>member evpn-instance <i>evpn-instance-id</i> vni <i>l2-vni-number</i></b> <b>Example:</b> Device(config-vlan)# <b>member evpn-instance 1 vni 10000</b>	Adds EVPN instance as a member of the VLAN configuration. The VNI here is used as a Layer 2 VNI.
<b>Step 5</b>	<b>end</b> <b>Example:</b> Device(config-vlan)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring the Loopback Interface on a VTEP

To configure the loopback interface on a VTEP, perform the following steps:

**Procedure**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface <i>loopback-interface-id</i></b> <b>Example:</b> Device(config)# <b>interface Loopback0</b>	Enters interface configuration mode for the specified Loopback interface.

	Command or Action	Purpose
<b>Step 4</b>	<b>ip address</b> <i>ipv4-address</i> <b>Example:</b> Device(config-if) # <b>ip address 10.12.11.11</b>	Configures the IP address for the Loopback interface.
<b>Step 5</b>	<b>ip pim sparse mode</b> <b>Example:</b> Device(config-if) # <b>ip pim sparse mode</b>	Enables Protocol Independent Multicast (PIM) sparse mode on the Loopback interface.
<b>Step 6</b>	<b>end</b> <b>Example:</b> Device(config-vlan) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring the Access-Facing Interface in the VLAN on a VTEP

To configure the access-facing interface in the VLAN on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>interface-name</i> <b>Example:</b> Device(config) # <b>interface GigabitEthernet1/0/1</b>	Enters interface configuration mode for the specified interface.
<b>Step 4</b>	<b>switchport access vlan</b> <i>vlan-id</i> <b>Example:</b> Device(config-if) # <b>switchport access vlan 11</b>	Configures the interface as a static-access port of the specified VLAN. Interface can also be configured as a trunk interface, if required.
<b>Step 5</b>	<b>end</b> <b>Example:</b> Device(config-if) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring the NVE Interface on a VTEP

To add a VNI member to the NVE interface of a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface nve-interface-id</b> <b>Example:</b> Device(config)# <b>interface nve1</b>	Defines the interface to be configured as a trunk, and enters interface configuration mode.
<b>Step 4</b>	<b>no ip address</b> <b>Example:</b> Device(config-if)# <b>no ip address</b>	Disables IP processing on the interface by removing its IP address.
<b>Step 5</b>	<b>source-interface loopback-interface-id</b> <b>Example:</b> Device(config-if)# <b>source-interface loopback0</b>	Sets the IP address of the specified loopback interface as the source IP address.
<b>Step 6</b>	<b>host-reachability protocol bgp</b> <b>Example:</b> Device(config-if)# <b>host-reachability protocol bgp</b>	Configures BGP as the host-reachability protocol on the interface.
<b>Step 7</b>	<b>member vni layer2-vni-id {ingress-replication   mcast-group multicast-group-address}</b> <b>Example:</b> Device(config-if)# <b>member vni 10000 mcast-group 227.0.0.1</b>	Associates the Layer 2 VNI member with the NVE.  The specified replication type must match the replication type that is configured globally or for the specific EVPN instance. Use <b>mcast-group</b> keyword for static replication and <b>ingress-replication</b> keyword for ingress replication.
<b>Step 8</b>	<b>end</b> <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring BGP on a VTEP with EVPN Address Family

To configure BGP on a VTEP with EVPN address family and with spine switch as the neighbor, perform the following steps:

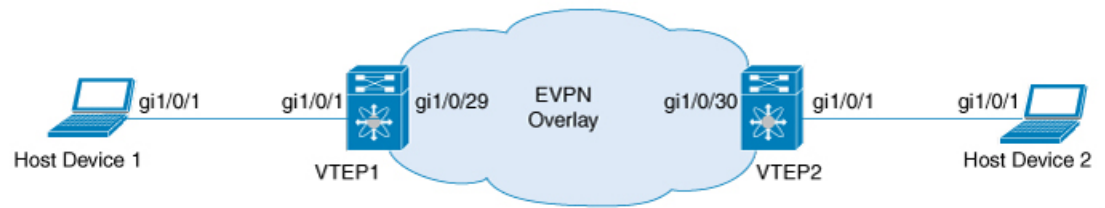
### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>router bgp <i>autonomous-system-number</i></b> <b>Example:</b> Device(config)# <b>router bgp 1</b>	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
<b>Step 4</b>	<b>bgp log-neighbor-changes</b> <b>Example:</b> Device(config-router)# <b>bgp log-neighbor-changes</b>	(Optional) Enables the generation of logging messages when the status of a BGP neighbor changes.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
<b>Step 5</b>	<b>bgp update-delay <i>time-period</i></b> <b>Example:</b> Device(config-router)# <b>bgp update-delay 1</b>	(Optional) Sets the maximum initial delay period before sending the first update.  The range is 1 to 3600 seconds.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
<b>Step 6</b>	<b>bgp graceful-restart</b> <b>Example:</b> Device(config-router)# <b>bgp graceful-restart</b>	(Optional) Enables the BGP graceful restart capability for all BGP neighbors.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
<b>Step 7</b>	<b>no bgp default ipv4-unicast</b> <b>Example:</b> Device(config-router)# <b>no bgp default ipv4-unicast</b>	(Optional) Disables default IPv4 unicast address family for BGP peering session establishment.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
<b>Step 8</b>	<b>neighbor <i>ip-address</i> remote-as <i>number</i></b> <b>Example:</b> Device(config-router)# <b>neighbor 10.11.11.11 remote-as 1</b>	Defines multiprotocol-BGP neighbors. Under each neighbor, define the Layer 2 Virtual Private Network (L2VPN) EVPN configuration.

	Command or Action	Purpose
		Use the IP address of the spine switch as the neighbor IP address.
<b>Step 9</b>	<b>neighbor</b> { <i>ip-address</i>   <i>group-name</i> } <b>update-source</b> <i>interface</i> <b>Example:</b> Device(config-router)# <b>neighbor</b> 10.11.11.11 <b>update-source</b> Loopback0	Configures update source. Update source can be configured per neighbor or per peer-group. Use the IP address of the spine switch as the neighbor IP address.
<b>Step 10</b>	<b>address-family</b> l2vpn evpn <b>Example:</b> Device(config-router)# <b>address-family</b> l2vpn evpn	Specifies the L2VPN address family and enters address family configuration mode.
<b>Step 11</b>	<b>neighbor</b> <i>ip-address</i> <b>activate</b> <b>Example:</b> Device(config-router-af)# <b>neighbor</b> 10.11.11.11 <b>activate</b>	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
<b>Step 12</b>	<b>neighbor</b> <i>ip-address</i> <b>send-community</b> [ <b>both</b>   <b>extended</b>   <b>standard</b> ] <b>Example:</b> Device(config-router-af)# <b>neighbor</b> 10.11.11.11 <b>send-community</b> both	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
<b>Step 13</b>	<b>exit-address-family</b> <b>Example:</b> Device(config-router-af)# <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 14</b>	<b>end</b> <b>Example:</b> Device(config-router)# <b>end</b>	Returns to privileged EXEC mode.

## Configuration Examples for EVPN VXLAN Layer 2 Overlay Network

This section provides an example for configuring an EVPN VXLAN Layer 2 overlay network. This example shows a sample configuration for a VXLAN network with 2 VTEPs, VTEP 1 and VTEP 2, connected to perform bridging.



**Note** In a two-VTEP topology, a spine switch is not mandatory. For information about configuration of spine switches in an EVPN VXLAN network, see *Configuring Spine Switches in a BGP EVPN VXLAN Fabric* module.

*Table 1: Configuration Example for a VXLAN Network with Two VTEPs Connected to Perform Bridging*

VTEP 1	VTEP 2
--------	--------



VTEP 1	VTEP 2
<pre>VTEP1# show running-config Building configuration... ! hostname VTEP1 ! ip routing ip multicast-routing ! l2vpn evpn   replication-type static   router-id Loopback0 ! l2vpn evpn instance 1 vlan-based   encapsulation vxlan   route-target export 103:1   route-target import 104:1 ! system mtu 9150 ! vlan configuration 201   member evpn-instance 1 vni 6000 ! ! interface Loopback0   ip address 10.1.1.10 255.255.255.255   ip pim sparse-mode ! ! interface GigabitEthernet1/0/1   description host1-interface   switchport access vlan 201   switchport mode access ! ! interface GigabitEthernet1/0/29   description core-underlay-interface   no switchport   ip address 172.16.1.29 255.255.255.0   ip pim sparse-mode ! ! interface nve10   no ip address   source-interface Loopback0   host-reachability protocol bgp   member vni 6000 mcast-group 232.1.1.1 ! router ospf 1   router-id 10.1.1.10   network 10.1.1.0 0.0.0.255 area 0   network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10   bgp router-id interface Loopback0   bgp log-neighbor-changes   bgp update-delay 1   no bgp default ipv4-unicast   neighbor 10.2.2.20 remote-as 10   neighbor 10.2.2.20 update-source Loopback0 ! address-family ipv4 exit-address-family</pre>	<pre>VTEP2# show running-config Building configuration... ! hostname VTEP2 ! ip routing ip multicast-routing ! l2vpn evpn   replication-type static   router-id Loopback0 ! l2vpn evpn instance 1 vlan-based   encapsulation vxlan   route-target export 104:1   route-target import 103:1 ! system mtu 9150 ! vlan configuration 201   member evpn-instance 1 vni 6000 ! ! interface Loopback0   ip address 10.2.2.20 255.255.255.255   ip pim sparse-mode ! ! interface GigabitEthernet1/0/1   description host2-interface   switchport access vlan 201   switchport mode access ! ! interface GigabitEthernet1/0/30   description core-underlay-interface   no switchport   ip address 172.16.1.30 255.255.255.0   ip pim sparse-mode ! ! interface nve10   no ip address   source-interface Loopback0   host-reachability protocol bgp   member vni 6000 mcast-group 232.1.1.1 ! router ospf 1   router-id 10.2.2.20   network 10.2.2.0 0.0.0.255 area 0   network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10   bgp router-id interface Loopback0   bgp log-neighbor-changes   bgp update-delay 1   no bgp default ipv4-unicast   neighbor 10.1.1.10 remote-as 10   neighbor 10.1.1.10 update-source Loopback0 ! address-family ipv4 exit-address-family</pre>

VTEP 1	VTEP 2
<pre> ! address-family l2vpn evpn   neighbor 10.2.2.20 activate   neighbor 10.2.2.20 send-community both exit-address-family ! ip pim rp-address 10.1.1.10 ! end </pre>	<pre> ! address-family l2vpn evpn   neighbor 10.1.1.10 activate   neighbor 10.1.1.10 send-community both exit-address-family ! ip pim rp-address 10.1.1.10 ! end </pre>

The following examples provide outputs for **show** commands on VTEP 1 and VTEP 2 in the topology configured above.

- [show l2vpn evpn peers vxlan, on page 26](#)
- [show nve peers, on page 26](#)
- [show l2vpn evpn mac, on page 27](#)
- [show bgp l2vpn evpn all, on page 27](#)
- [show platform software fed switch active matm macTable vlan, on page 28](#)

### show l2vpn evpn peers vxlan

#### VTEP 1

This example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 1:

```

VTEP1# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP              Num routes  eVNI      UP time
-----
nve10     6000             10.2.2.20            3           6000     00:12:44

```

#### VTEP 2

This example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 2:

```

VTEP2# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP              Num routes  eVNI      UP time
-----
nve10     6000             10.1.1.10            3           6000     00:01:41

```

### show nve peers

#### VTEP 1

This example shows the output for the **show nve peers** command on VTEP 1:

```

VTEP1# show nve peers
Interface VNI      Type Peer-IP              RMAC/Num_RTs  eVNI      state flags UP time
-----
nve10     6000     L2CP 10.2.2.20        3           6000     UP   N/A  00:12:48

```

**VTEP 2**

This example shows the output for the **show nve peers** command on VTEP 2:

```
VTEP2# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve10    6000    L2CP 10.1.1.10      3           6000      UP    N/A  00:01:46
```

**show l2vpn evpn mac****VTEP 1**

This example shows the output for the **show l2vpn evpn mac** command on VTEP 1:

```
VTEP1# show l2vpn evpn mac
MAC Address      EVI  VLAN  ESI                               Ether Tag  Next Hop(s)
-----
0018.736c.5681  1    201  0000.0000.0000.0000.0000  0          10.2.2.20
0018.736c.56c3  1    201  0000.0000.0000.0000.0000  0          10.2.2.20
0059.dc50.ae01  1    201  0000.0000.0000.0000.0000  0          Gi1/0/1:201
0059.dc50.ae4c  1    201  0000.0000.0000.0000.0000  0          Gi1/0/1:201
```

**VTEP 2**

This example shows the output for the **show l2vpn evpn mac** command on VTEP 2:

```
VTEP2# show l2vpn evpn mac
MAC Address      EVI  VLAN  ESI                               Ether Tag  Next Hop(s)
-----
0018.736c.5681  1    201  0000.0000.0000.0000.0000  0          Gi1/0/1:201
0018.736c.56c3  1    201  0000.0000.0000.0000.0000  0          Gi1/0/1:201
0059.dc50.ae01  1    201  0000.0000.0000.0000.0000  0          10.1.1.10
0059.dc50.ae4c  1    201  0000.0000.0000.0000.0000  0          10.1.1.10
```

**show bgp l2vpn evpn all****VTEP 1**

This example shows the output for the **show bgp l2vpn evpn all** command on VTEP 1:

```
VTEP1# show bgp l2vpn evpn all
BGP table version is 101, local router ID is 10.1.1.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.10:1
*>i  [2] [10.1.1.10:1] [0] [48] [0018736C5681] [0] [*]/20
      10.2.2.20          0      100      0 ?
*>i  [2] [10.1.1.10:1] [0] [48] [0018736C56C3] [0] [*]/20
      10.2.2.20          0      100      0 ?
*>i  [2] [10.1.1.10:1] [0] [48] [0018736C56C3] [32] [192.168.1.89]/24
```

```

      10.2.2.20          0 100 0 ?
*> [2][10.1.1.10:1][0][48][0059DC50AE01][0][*]/20
      ::                32768 ?
*> [2][10.1.1.10:1][0][48][0059DC50AE4C][0][*]/20
      ::                32768 ?
*> [2][10.1.1.10:1][0][48][0059DC50AE4C][32][192.168.1.81]/24
      ::                32768 ?
Route Distinguisher: 10.2.2.20:1
*>i [2][10.2.2.20:1][0][48][0018736C5681][0][*]/20
      10.2.2.20          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0018736C56C3][0][*]/20
      10.2.2.20          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0018736C56C3][32][192.168.1.89]/24
      10.2.2.20          0 100 0 ?

```

## VTEP 2

This example shows the output for the **show bgp l2vpn evpn all** command on VTEP 2:

```

VTEP2# show bgp l2vpn evpn all
BGP table version is 99, local router ID is 10.2.2.20
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.10:1
*>i [2][10.1.1.10:1][0][48][0059DC50AE01][0][*]/20
      10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][0059DC50AE4C][0][*]/20
      10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][0059DC50AE4C][32][192.168.1.81]/24
      10.1.1.10          0 100 0 ?
Route Distinguisher: 10.2.2.20:1
*> [2][10.2.2.20:1][0][48][0018736C5681][0][*]/20
      ::                32768 ?
*> [2][10.2.2.20:1][0][48][0018736C56C3][0][*]/20
      ::                32768 ?
*> [2][10.2.2.20:1][0][48][0018736C56C3][32][192.168.1.89]/24
   Network          Next Hop          Metric LocPrf Weight Path
      ::                32768 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE01][0][*]/20
      10.1.1.10          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE4C][0][*]/20
      10.1.1.10          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE4C][32][192.168.1.81]/24
      10.1.1.10          0 100 0 ?

```

## show platform software fed switch active matm macTable vlan

### VTEP 1

This example shows the output for the **show platform software fed switch active matm mactable vlan** command on VTEP 1:

```

VTEP1# show platform software fed switch active matm macTable vlan 201
VLAN  MAC              Type Seq#  EC_Bi  Flags machandle  siHandle
      riHandle          diHandle          *a_time *e_time  ports

```

```

201 0018.736c.5681 0x1000001 0 0 64 0x7f5d852abaf8 0x7f5d850c1858
    0x7f5d8527def8 0x0 0 0 RLOC 10.2.2.20 adj_id 81

201 0018.736c.56c3 0x1000001 0 0 64 0x7f5d855be2b8 0x7f5d850c1858
    0x7f5d8527def8 0x0 0 0 RLOC 10.2.2.20 adj_id 81

201 0059.dc50.ae01 0x1 22 0 0 0x7f5d855c6388 0x7f5d85035248
    0x0 0x7f5d8517eae8 300 11 GigabitEthernet1/0/1

201 0059.dc50.ae4c 0x1 26 0 0 0x7f5d84fba3c8 0x7f5d85035248
    0x0 0x7f5d8517eae8 300 58 GigabitEthernet1/0/1

```

Total Mac number of addresses:: 4

## VTEP 2

This example shows the output for the **show platform software fed switch active matm macTable vlan** command on VTEP 2:

```

VTEP2# show platform software fed switch active matm macTable vlan 201
VLAN  MAC                Type Seq#  EC_Bi  Flags  machandle  ports  siHandle
      riHandle            diHandle  *a_time *e_time
-----
201  0018.736c.5681        0x1   38     0      0  0x7f40e196cac8  0x7f40e196cf28
    0x0 0x7f40e0f6da38      300    12  GigabitEthernet1/0/1

201  0018.736c.56c3        0x1   39     0      0  0x7f40e19b6878  0x7f40e196cf28
    0x0 0x7f40e0f6da38      300    17  GigabitEthernet1/0/1

201  0059.dc50.ae01        0x1000001 0     0      64  0x7f40e19b88f8  0x7f40e1937b88
    0x7f40e193bd58 0x0 0 17 RLOC 10.1.1.10 adj_id 28

201  0059.dc50.ae4c        0x1000001 0     0      64  0x7f40e194d638  0x7f40e1937b88
    0x7f40e193bd58 0x0 0 17 RLOC 10.1.1.10 adj_id 28

```

Total Mac number of addresses:: 4

# Verifying EVPN VXLAN Layer 2 Overlay Network

The following table lists the **show** commands that are used to verify a Layer 2 VXLAN overlay network:

**Table 2: Commands to Verify EVPN VXLAN Layer 2 Overlay Network**

Command	Purpose
<b>show l2vpn evpn evi [detail]</b>	Displays detailed information for a particular EVPN instance or all EVPN instances.
<b>show l2vpn evpn mac [detail]</b>	Displays the MAC address database for Layer 2 EVPN.
<b>show l2vpn evpn mac ip [detail]</b>	Displays the IP address database for Layer 2 EVPN.

Command	Purpose
<b>show l2vpn evpn summary</b>	Displays a summary of Layer 2 EVPN information.
<b>show l2vpn evpn capabilities</b>	Displays platform capability information for Layer 2 EVPN.
<b>show l2vpn evpn peers</b>	Displays Layer 2 EVPN peer route counts and up time.
<b>show l2vpn evpn route-target</b>	Displays Layer 2 EVPN import route targets.
<b>show l2vpn evpn memory</b>	Displays Layer 2 EVPN memory usage.
<b>show l2route evpn summary</b>	Displays a summary of EVPN routes.
<b>show l2route evpn mac [detail]</b>	Displays MAC address information learnt by the switch in the EVPN control plane.
<b>show l2route evpn mac ip [detail]</b>	Displays MAC and IP address information learnt by the switch in the EVPN control plane.
<b>show l2route evpn imet detail</b>	Displays the IMET route details for Layer 2 EVPN address family.  This command shows details only about traffic forwarded using ingress replication.
<b>show bgp l2vpn evpn</b>	Displays BGP information for Layer 2 VPN EVPN address family.
<b>show bgp l2vpn evpn route-type 2</b>	Displays BGP information for route type 2 of L2VPN EVPN address family.
<b>show bgp l2vpn evpn evi context</b>	Displays context information for Layer 2 EVPN instances.
<b>show bgp l2vpn evpn evi <i>evpn-instance-id</i> route-type 3</b>	Displays route type 3 information for the specified Layer 2 EVPN instance.  This command shows details only about traffic forwarded using ingress replication.
<b>show l2fib bridge-domain <i>bridge-domain-number</i> detail</b>	Displays detailed information for a Layer 2 forwarding information base bridge domain.
<b>show l2fib bridge-domain <i>bridge-domain-number</i> address unicast</b>	Displays unicast MAC address information for a Layer 2 forwarding information base bridge domain.
<b>show nve vni</b>	Displays information about VXLAN network identifier members associated with an NVE interface.
<b>show nve vni <i>vni-id</i> detail</b>	Displays detailed NVE interface state information for a VXLAN network identifier member.

Command	Purpose
<b>show nve peers</b>	Displays NVE interface state information for peer leaf switches.
<b>show mac address-table vlan <i>vlan-id</i></b>	Displays MAC addresses for a VLAN.
<b>show platform software fed switch active matm macTable vlan <i>vlan-id</i></b>	Displays MAC addresses for a VLAN from MAC address table manager database for Forwarding Engine Driver (FED).
<b>show device-tracking database</b>	Displays device tracking database.
<b>show device-tracking database mac</b>	Displays device tracking MAC address database.
<b>show ip mroute</b>	Displays multicast routing table information.







## CHAPTER 3

# Configuring EVPN VXLAN Layer 3 Overlay Network

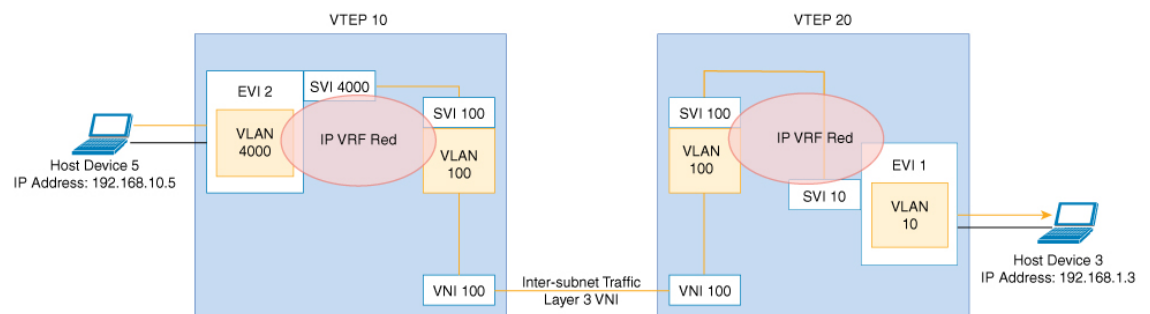
- [Information About EVPN VXLAN Layer 3 Overlay Network](#), on page 33
- [How to Configure EVPN VXLAN Layer 3 Overlay Network](#), on page 34
- [Configuration Examples for EVPN VXLAN Layer 3 Overlay Network](#), on page 44
- [Verifying EVPN VXLAN Layer 3 Overlay Network](#), on page 51

## Information About EVPN VXLAN Layer 3 Overlay Network

An EVPN VXLAN Layer 3 overlay network allows host devices in different Layer 2 networks to send Layer 3 or routed traffic to each other. The network forwards the routed traffic using a Layer 3 virtual network instance (VNI) and an IP VRF.

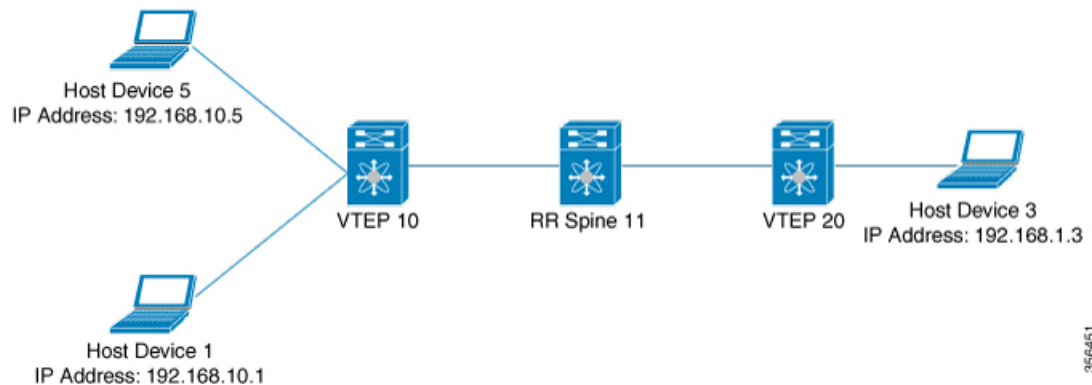
This module provides information only about how to configure a Layer 3 overlay network. You can also configure both Layer 2 and Layer 3 overlay networks together to enable integrated routing and bridging (IRB). For more information about IRB, see *Configuring EVPN VXLAN Integrated Routing and Bridging* module.

The following figure shows the movement of traffic in an EVPN VXLAN Layer 3 overlay network using a Layer 3 VNI:



# How to Configure EVPN VXLAN Layer 3 Overlay Network

The following figure shows a sample topology of an EVPN VXLAN Network. Host device 3 and host device 5 are part of different subnets. The network forwards traffic from host device 1 to host device 3 using a Layer 3 VNI and an IP VRF.



## Note

In a two-VTEP topology, a spine switch is not mandatory. For information about configuration of spine switches in an EVPN VXLAN network, see *Configuring Spine Switches in a BGP EVPN VXLAN Fabric* module.

Perform the following set of procedures to configure an EVPN VXLAN Layer 3 overlay network:

- Configure the IP VRF on the VTEPs.
- Configure the core-facing VLAN on the VTEPs.
- Configure the access-facing VLAN on the VTEPs.
- Configure the switch virtual interface (SVI) for the core-facing VLAN.
- Configure the SVI for the access-facing VLAN.
- Configure the loopback interface on the VTEPs.
- Configure the network virtualization endpoint (NVE) interface on the VTEPs.
- Configure BGP with either IPv4 or IPv6 or both address families on the VTEPs.

## Configuring an IP VRF on a VTEP

To configure an IP VRF on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
Step 1	<code>enable</code>	Enables privileged EXEC mode.

	Command or Action	Purpose
	<b>Example:</b> Device> <b>enable</b>	Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>vrf definition</b> <i>vrf-name</i> <b>Example:</b> Device (config)# <b>vrf definition Green</b>	Enters the VRF configuration mode for the specified VRF instance.
<b>Step 4</b>	<b>rd</b> <i>vpn-route-distinguisher</i> <b>Example:</b> Device (config-vrf)# <b>rd 100:1</b>	Specifies the route distinguisher for the VRF instance.
<b>Step 5</b>	<b>address-family ipv4</b> [ <b>multicast</b>   <b>unicast</b> ] <b>Example:</b> Device (config-vrf)# <b>address-family ipv4</b>	Enters the IPv4 address family configuration mode.
<b>Step 6</b>	<b>route-target</b> { <b>export</b>   <b>import</b>   <b>both</b> } <i>route-target-ext-community</i> <b>Example:</b> Device (config-vrf-af)# <b>route-target export 100:1</b> <b>Example:</b> Device (config-vrf-af)# <b>route-target import 100:1</b>	Creates a list of import, export, or both import and export route target communities for the specified VRF.  Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).
<b>Step 7</b>	<b>route-target</b> { <b>export</b>   <b>import</b>   <b>both</b> } <i>route-target-ext-community</i> <b>stitching</b> <b>Example:</b> Device (config-vrf-af)# <b>route-target export 100:1 stitching</b> <b>Example:</b> Device (config-vrf-af)# <b>route-target import 100:1 stitching</b>	Configures importing, exporting, or both importing and exporting of EVPN route target communities for the VRF.
<b>Step 8</b>	<b>exit-address-family</b> <b>Example:</b> Device (config-vrf-af)# <b>exit-address-family</b>	Exits VRF address family configuration mode and enters VRF configuration mode.
<b>Step 9</b>	<b>address-family ipv6</b> [ <b>multicast</b>   <b>unicast</b> ] <b>Example:</b> Device (config-vrf)# <b>address-family ipv6</b>	Enters the IPv6 address family configuration mode.

	Command or Action	Purpose
<b>Step 10</b>	<b>route-target {export   import   both}</b> <i>route-target-ext-community</i>  <b>Example:</b> Device (config-vrf-af) # <b>route-target export 100:1</b>  <b>Example:</b> Device (config-vrf-af) # <b>route-target import 100:1</b>	Creates a list of import, export, or both import and export route target communities for the specified VRF.  Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).
<b>Step 11</b>	<b>route-target {export   import   both}</b> <i>route-target-ext-community stitching</i>  <b>Example:</b> Device (config-vrf-af) # <b>route-target export 100:1 stitching</b>  <b>Example:</b> Device (config-vrf-af) # <b>route-target import 100:1 stitching</b>	Configures importing, exporting, or both importing and exporting of VXLAN route target communities for the VRF.
<b>Step 12</b>	<b>exit-address-family</b>  <b>Example:</b> Device (config-vrf-af) # <b>exit-address-family</b>	Exits VRF address family configuration mode and enters VRF configuration mode.
<b>Step 13</b>	<b>end</b>  <b>Example:</b> Device (config-vrf) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring the Loopback Interface on a VTEP

To configure the loopback interface on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface loopback-interface-id</b>  <b>Example:</b>	Enters interface configuration mode for the specified Loopback interface.

	Command or Action	Purpose
	Device(config)# <b>interface Loopback0</b>	
<b>Step 4</b>	<b>ip address <i>ipv4-address</i></b> <b>Example:</b> Device(config-if)# <b>ip address 10.12.11.11 255.255.255.255</b>	Configures the IP address for the Loopback interface.
<b>Step 5</b>	<b>ip pim sparse mode</b> <b>Example:</b> Device(config-if)# <b>ip pim sparse mode</b>	(Optional) Enables Protocol Independent Multicast (PIM) sparse mode on the Loopback interface.  <b>Note</b> Enable PIM sparse mode only if EVPN VXLAN Layer 2 overlay network is also configured on the VTEP with underlay multicast as the mechanism for forwarding BUM traffic.
<b>Step 6</b>	<b>end</b> <b>Example:</b> Device(config-vlan)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring the Core-facing VLAN on a VTEP

To configure the core-facing VLAN on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>vlan configuration <i>vlan-id</i></b> <b>Example:</b> Device(config)# <b>vlan configuration 11</b>	Enters VLAN feature configuration mode for the specified VLAN interface.
<b>Step 4</b>	<b>member vni <i>l3-vni-number</i></b> <b>Example:</b> Device(config-vlan)# <b>member vni 5000</b>	Adds EVPN instance as a member of the VLAN configuration.  The VNI here is used as a Layer 3 VNI.

	Command or Action	Purpose
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(config-vlan)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring Access-facing VLAN on a VTEP

To configure the access-facing VLAN on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface interface-name</b>  <b>Example:</b> Device(config)# <b>interface</b> <b>GigabitEthernet1/0/1</b>	Enters interface configuration mode for the specified interface.
<b>Step 4</b>	<b>switchport access vlan vlan-id</b>  <b>Example:</b> Device(config-if)# <b>switchport access vlan</b> <b>40</b>	Configures the interface as a static-access port of the specified VLAN. Interface can also be configured as a trunk interface, if required.
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring Switch Virtual Interface for the Core-facing VLAN

To configure an SVI for the core-facing VLAN on the VTEP:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b>	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
	<code>Device&gt; enable</code>	
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <code>Device# configure terminal</code>	Enters global configuration mode.
<b>Step 3</b>	<b>interface vlan <i>vlan-id</i></b> <b>Example:</b> <code>Device(config)# interface vlan 11</code>	Enters interface configuration mode for the specified VLAN.
<b>Step 4</b>	<b>vrf forwarding <i>vrf-name</i></b> <b>Example:</b> <code>Device(config-if)# vrf forwarding Green</code>	Configures the SVI for the VLAN.
<b>Step 5</b>	<b>ip unnumbered <i>Loopback-interface</i></b> <b>Example:</b> <code>Device(config-if)# ip unnumbered Loopback0</code>	Enables IP processing on the Loopback interface without assigning an explicit IP address to the interface.
<b>Step 6</b>	<b>no autostate</b> <b>Example:</b> <code>Device(config-if)# no autostate</code>	Disables autostate on the interface.  In EVPN deployments, once a VLAN is used for a core-facing SVI, it should not be allowed in any trunk. For a core-facing SVI to function properly, the <b>no autostate</b> command must be configured under the SVI.
<b>Step 7</b>	<b>end</b> <b>Example:</b> <code>Device(config-if)# end</code>	Returns to privileged EXEC mode.

## Configuring the Switch Virtual Interface for the Access-facing VLANs

To configure the SVI for the access-facing VLAN on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> <code>Device&gt; enable</code>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <code>Device# configure terminal</code>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 3</b>	<b>interface</b> <i>vlan</i> <i>vlan-id</i> <b>Example:</b> Device(config)# <b>interface</b> <b>vlan</b> <b>40</b>	Enters interface configuration mode for the specified VLAN.
<b>Step 4</b>	<b>vrf forwarding</b> <i>vrf-name</i> <b>Example:</b> Device(config-if)# <b>vrf forwarding</b> <b>Green</b>	Configures the SVI for the VLAN.
<b>Step 5</b>	<b>ip address</b> <i>ip-address</i> <b>Example:</b> Device(config-if)# <b>ip address</b> <b>192.168.10.100 255.255.255.0</b>	Configures the IP address of the SVI.
<b>Step 6</b>	<b>mac-address</b> <i>mac-address-value</i> <b>Example:</b> Device(config-if)# <b>mac-address</b> <b>aabb.cc01.f100</b>	(Optional) Manually sets the MAC address for the VLAN interface.
<b>Step 7</b>	<b>exit</b> <b>Example:</b> Device(config-if)# <b>exit</b>	Returns to global configuration mode.
<b>Step 8</b>	<b>end</b> <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring the NVE Interface on a VTEP

To add a Layer 3 VNI member to the NVE interface on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>nve-interface-id</i> <b>Example:</b> Device(config)# <b>interface</b> <b>nve1</b>	Defines the interface to be configured as a trunk, and enters interface configuration mode.



	Command or Action	Purpose
<b>Step 4</b>	<b>no ip address</b> <b>Example:</b> Device(config-if)# <b>no ip address</b>	Disables IP processing on the interface by removing its IP address.
<b>Step 5</b>	<b>source-interface loopback-interface-id</b> <b>Example:</b> Device(config-if)# <b>source-interface loopback0</b>	Sets the IP address of the specified loopback interface as the source IP address.
<b>Step 6</b>	<b>host-reachability protocol bgp</b> <b>Example:</b> Device(config-if)# <b>host-reachability protocol bgp</b>	Configures BGP as the host-reachability protocol on the interface.
<b>Step 7</b>	<b>member vni vni-id vrf vrf-name</b> <b>Example:</b> Device(config-if)# <b>member vni 5000 vrf Green</b>	Associates the Layer 3 VNI id with the NVE interface. <b>Note</b> The Layer 3 VNI id must match with the VNI id configured in the core VLAN on the VTEP.
<b>Step 8</b>	<b>end</b> <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring BGP with IPv4 or IPv6 or Both Address Families on VTEP

To configure BGP on a VTEP with IPv4 or IPv6 or both address families and a spine switch as the neighbor, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>router bgp autonomous-system-number</b> <b>Example:</b> Device(config)# <b>router bgp 1</b>	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.

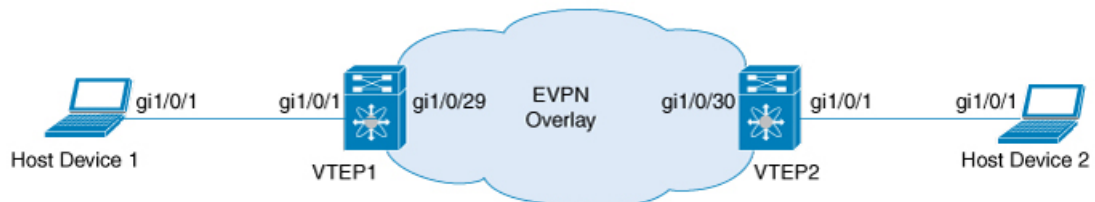
	Command or Action	Purpose
Step 4	<b>bgp log-neighbor-changes</b> <b>Example:</b> <pre>Device(config-router)# bgp log-neighbor-changes</pre>	(Optional) Enables the generation of logging messages when the status of a BGP neighbor changes.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
Step 5	<b>bgp update-delay time-period</b> <b>Example:</b> <pre>Device(config-router)# bgp update-delay 1</pre>	(Optional) Sets the maximum initial delay period before sending the first update.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
Step 6	<b>bgp graceful-restart</b> <b>Example:</b> <pre>Device(config-router)# bgp graceful-restart</pre>	(Optional) Enables the BGP graceful restart capability for all BGP neighbors.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
Step 7	<b>no bgp default ipv4-unicast</b> <b>Example:</b> <pre>Device(config-router)# no bgp default ipv4-unicast</pre>	(Optional) Disables default IPv4 unicast address family for BGP peering session establishment.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
Step 8	<b>neighbor ip-address remote-as number</b> <b>Example:</b> <pre>Device(config-router)# neighbor 10.11.11.11 remote-as 1</pre>	Defines multiprotocol-BGP neighbors. Under each neighbor, define the configuration.  Use the IP address of the spine switch as the neighbor IP address.
Step 9	<b>neighbor {ip-address   group-name} update-source interface</b> <b>Example:</b> <pre>Device(config-router)# neighbor 10.11.11.11 update-source Loopback0</pre>	Configures update source. Update source can be configured per neighbor or per peer-group.  Use the IP address of the spine switch as the neighbor IP address.
Step 10	<b>address-family l2vpn evpn</b> <b>Example:</b> <pre>Device(config-router)# address-family l2vpn evpn</pre>	Specifies the L2VPN address family and enters address family configuration mode.
Step 11	<b>neighbor ip-address activate</b> <b>Example:</b> <pre>Device(config-router-af)# neighbor 10.11.11.11 activate</pre>	Enables the exchange information from a BGP neighbor.  Use the IP address of the spine switch as the neighbor IP address.
Step 12	<b>neighbor ip-address send-community [both   extended   standard]</b> <b>Example:</b>	Specifies the communities attribute sent to a BGP neighbor.  Use the IP address of the spine switch as the neighbor IP address.

	Command or Action	Purpose
	Device(config-router-af)# <b>neighbor</b> 10.11.11.11 <b>send-community both</b>	
<b>Step 13</b>	<b>exit-address-family</b>  <b>Example:</b> Device(config-router-af)# <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 14</b>	<b>address-family ipv4 vrf vrf-name</b>  <b>Example:</b> Device(config-router)# <b>address-family</b> <b>ipv4 vrf Green</b>	Specifies the IPv4 address family and enters address family configuration mode.
<b>Step 15</b>	<b>advertise l2vpn evpn</b>  <b>Example:</b> Device(config-router-af)# <b>advertise</b> <b>l2vpn evpn</b>	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.
<b>Step 16</b>	<b>redistribute connected</b>  <b>Example:</b> Device(config-router-af)# <b>redistribute</b> <b>connected</b>	(Optional) Redistributes connected routes to BGP.
<b>Step 17</b>	<b>redistribute static</b>  <b>Example:</b> Device(config-router-af)# <b>redistribute</b> <b>static</b>	(Optional) Redistributes static routes to BGP.
<b>Step 18</b>	<b>exit-address-family</b>  <b>Example:</b> Device(config-router-af)# <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 19</b>	<b>address-family ipv6 vrf vrf-name</b>  <b>Example:</b> Device(config-router)# <b>address-family</b> <b>ipv6 vrf green</b>	Specifies the IPv6 address family and enters address family configuration mode.
<b>Step 20</b>	<b>advertise l2vpn evpn</b>  <b>Example:</b> Device(config-router-af)# <b>advertise</b> <b>l2vpn evpn</b>	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.
<b>Step 21</b>	<b>redistribute connected</b>  <b>Example:</b> Device(config-router-af)# <b>redistribute</b> <b>connected</b>	(Optional) Redistributes connected routes to BGP.

	Command or Action	Purpose
<b>Step 22</b>	<b>redistribute static</b>  <b>Example:</b> Device (config-router-af) # <b>redistribute static</b>	(Optional) Redistributes static routes to BGP.
<b>Step 23</b>	<b>exit-address-family</b>  <b>Example:</b> Device (config-router-af) # <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 24</b>	<b>end</b>  <b>Example:</b> Device (config-router) # <b>end</b>	Returns to privileged EXEC mode.

## Configuration Examples for EVPN VXLAN Layer 3 Overlay Network

This section provides an example for configuring an EVPN VXLAN Layer 3 overlay network. This example shows a sample configuration for a VXLAN network with 2 VTEPs, VTEP 1 and VTEP 2, connected to perform routing.



### Note

In a two-VTEP topology, a spine switch is not mandatory. For information about configuration of spine switches in an EVPN VXLAN network, see *Configuring Spine Switches in a BGP EVPN VXLAN Fabric* module.

*Table 3: Configuration Example for a VXLAN Network with Two VTEPs Connected to Perform Routing*

VTEP 1	VTEP 2
--------	--------

VTEP 1	VTEP 2
<pre> VTEP1# show running-config ! hostname VTEP1 ! ! vrf definition green  rd 103:2  !  address-family ipv4   route-target export 103:2   route-target import 104:2   route-target export 103:2 stitching   route-target import 104:2 stitching  exit-address-family !  address-family ipv6   route-target export 103:2   route-target import 104:2   route-target export 103:2 stitching   route-target import 104:2 stitching  exit-address-family ! ip multicast-routing ipv6 unicast-routing ! ! system mtu 9150 ! vlan configuration 200  member vni 5000 ! ! interface Loopback0  ip address 10.1.1.10 255.255.255.255  ip pim sparse-mode ! interface Loopback13  description demo only (for rt5 distribution)  vrf forwarding green  ip address 10.1.13.13 255.255.255.0 ! interface GigabitEthernet1/0/1  description access interface  switchport access vlan 201  switchport mode access ! ! interface GigabitEthernet1/0/29  description core-underlay-interface  no switchport  ip address 172.16.1.29 255.255.255.0  ip pim sparse-mode ! ! interface Vlan200  description core svi for l3vni  vrf forwarding green  ip unnumbered Loopback0  ipv6 enable  no autostate ! interface Vlan201 </pre>	<pre> VTEP2# show running-config ! hostname VTEP2 ! ! vrf definition green  rd 104:2  !  address-family ipv4   route-target export 104:2   route-target import 103:2   route-target export 104:2 stitching   route-target import 103:2 stitching  exit-address-family !  address-family ipv6   route-target export 104:2   route-target import 103:2   route-target export 104:2 stitching   route-target import 103:2 stitching  exit-address-family ! ip multicast-routing ipv6 unicast-routing ! ! system mtu 9150 ! vlan configuration 200  member vni 5000 ! ! interface Loopback0  ip address 10.2.2.20 255.255.255.255  ip pim sparse-mode ! interface Loopback14  description demo only (for rt5 distribution)  vrf forwarding green  ip address 10.1.14.14 255.255.255.0 ! interface GigabitEthernet1/0/1  description access interface  switchport access vlan 202  switchport mode access ! ! interface GigabitEthernet1/0/30  description core-underlay-interface  no switchport  ip address 172.16.1.30 255.255.255.0  ip pim sparse-mode ! ! interface Vlan200  description core svi for l3vni  vrf forwarding green  ip unnumbered Loopback0  ipv6 enable  no autostate ! interface Vlan202 </pre>

VTEP 1	VTEP 2
<pre> description access-svi vrf forwarding green ip address 192.168.1.201 255.255.255.0 ipv6 address 2001:DB8:201::201/64 ipv6 enable ! interface nve10 no ip address source-interface Loopback0 host-reachability protocol bgp member vni 5000 vrf green ! router ospf 1 router-id 10.1.1.10 network 10.1.1.0 0.0.0.255 area 0 network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10 bgp router-id interface Loopback0 bgp log-neighbor-changes bgp update-delay 1 no bgp default ipv4-unicast neighbor 10.2.2.20 remote-as 10 neighbor 10.2.2.20 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 10.2.2.20 activate neighbor 10.2.2.20 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 10.1.1.10 ! ! end </pre>	<pre> description access-svi vrf forwarding green ip address 192.168.2.202 255.255.255.0 ipv6 address 2001:DB8:202::202/64 ipv6 enable ! interface nve10 no ip address source-interface Loopback0 host-reachability protocol bgp member vni 5000 vrf green ! router ospf 1 router-id 10.2.2.20 network 10.2.2.0 0.0.0.255 area 0 network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10 bgp router-id interface Loopback0 bgp log-neighbor-changes bgp update-delay 1 no bgp default ipv4-unicast neighbor 10.1.1.10 remote-as 10 neighbor 10.1.1.10 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 10.1.1.10 activate neighbor 10.1.1.10 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 10.1.1.10 ! ! end </pre>

The following examples provide outputs for **show** commands on VTEP 1 and VTEP 2 in the topology configured above.

- [show nve peers, on page 48](#)
- [show bgp l2vpn evpn all, on page 48](#)
- [show ip route vrf, on page 49](#)
- [show platform software fed switch active matm mactable vlan, on page 50](#)

**show nve peers****VTEP 1**

The following example shows the output for the **show nve peers** command on VTEP 1:

```
VTEP1# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI    state flags UP time
nve10    5000    L3CP 10.2.2.20      380e.4d9b.6a4a 5000     UP  A/M/4 00:38:37
nve10    5000    L3CP 10.2.2.20      380e.4d9b.6a4a 5000     UP  A/-/6 00:03:16
```

**VTEP 2**

The following example shows the output for the **show nve peers** command on VTEP 2:

```
VTEP2# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI    state flags UP time
nve10    5000    L3CP 10.1.1.10      a0f8.4910.bce2 5000     UP  A/-/4 00:38:53
nve10    5000    L3CP 10.1.1.10      a0f8.4910.bce2 5000     UP  A/M/6 00:38:53
```

**show bgp l2vpn evpn all****VTEP 1**

The following example shows the output for the **show bgp l2vpn evpn all** command on VTEP 1:

```
VTEP1# show bgp l2vpn evpn all
BGP table version is 26, local router ID is 10.1.1.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 103:2 (default for vrf green)
*> [5][103:2][0][24][10.1.13.0]/17
      0.0.0.0          0          32768 ?
*> [5][103:2][0][24][192.168.1.0]/17
      0.0.0.0          0          32768 ?
*> [5][103:2][0][64][2001:DB8:201:]/29
      ::              0          32768 ?
Route Distinguisher: 104:2
*>i [5][104:2][0][24][10.1.14.0]/17
      10.2.2.20        0    100    0 ?
*>i [5][104:2][0][24][192.168.2.0]/17
      10.2.2.20        0    100    0 ?
*>i [5][104:2][0][64][2001:DB8:202:]/29
      10.2.2.20        0    100    0 ?
```

**VTEP 2**

The following example shows the output for the **show bgp l2vpn evpn all** command on VTEP 2:



```

VTEP2# show bgp l2vpn evpn all
BGP table version is 12, local router ID is 10.2.2.20
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 103:2
 *>i  [5][103:2][0][24][10.1.13.0]/17
              10.1.1.10              0      100      0 ?
 *>i  [5][103:2][0][24][192.168.1.0]/17
              10.1.1.10              0      100      0 ?
 *>i  [5][103:2][0][64][2001:DB8:201::]/29
              10.1.1.10              0      100      0 ?
Route Distinguisher: 104:2 (default for vrf green)
 *>  [5][104:2][0][24][10.1.14.0]/17
              0.0.0.0              0              32768 ?
 *>  [5][104:2][0][24][192.168.2.0]/17
              0.0.0.0              0              32768 ?
 *>  [5][104:2][0][64][2001:DB8:202::]/29
      Network          Next Hop          Metric LocPrf Weight Path
              ::              0              32768 ?

```

## show ip route vrf

### VTEP 1

The following example shows the output for the **show ip route vrf** command on VTEP 1:

```

VTEP1# show ip route vrf green
Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.1.13.0/24 is directly connected, Loopback13
L       10.1.13.13/32 is directly connected, Loopback13
B       10.1.14.0/24 [200/0] via 10.2.2.20, 00:42:01, Vlan200
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, Vlan201
L       192.168.1.201/32 is directly connected, Vlan201
B       192.168.2.0/24 [200/0] via 10.2.2.20, 00:06:00, Vlan200

```

### VTEP 2

The following example shows the output for the **show ip route vrf** command on VTEP 2:

```

VTEP2# show ip route vrf green
Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
        n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        H - NHRP, G - NHRP registered, g - NHRP registration summary
        o - ODR, P - periodic downloaded static route, l - LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B       10.1.13.0/24 [200/0] via 10.1.1.10, 00:42:38, Vlan200
C       10.1.14.0/24 is directly connected, Loopback14
L       10.1.14.14/32 is directly connected, Loopback14
B       192.168.1.0/24 [200/0] via 10.1.1.10, 00:42:38, Vlan200
        192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, Vlan202
L       192.168.2.202/32 is directly connected, Vlan202

```

### show platform software fed switch active matm mactable vlan

#### VTEP 1

The following example shows the output for the **show platform software fed switch active matm mactable vlan 200** command on VTEP 1:



**Note** The MAC address of the peer's core SVI interface must be present in the core VLAN.

```

VTEP1# show platform software fed switch active matm macTable vlan 200
VLAN  MAC                               Type Seq#  EC_Bi  Flags machandle      siHandle
      riHandle                          diHandle  *a_time *e_time  ports
-----
200   a0f8.4910.bce2                       0x8002   0 19880   64 0x7f5d8503fd48      0x7f5d852b6d28
      0x0                                0x5234   0       0       0   0 Vlan200
200   380e.4d9b.6a4a                       0x1000001 0 0       64 0x7f5d85117598      0x7f5d85110f78
      0x7f5d851b9648                       0x0      0       0       0   0 RLOC 10.2.2.20 adj_id 22

```

Total Mac number of addresses:: 2

#### VTEP 2

The following example shows the output for the **show platform software fed switch active matm mactable vlan 200** command on VTEP 2:



**Note** The MAC address of the peer's core SVI interface must be present in the core VLAN.

```
VTEP2# show platform software fed switch active matm macTable vlan 200
VLAN  MAC                               Type  Seq#  EC_Bi  Flags  machandle          siHandle
      riHandle                          diHandle                *a_time  *e_time  ports

-----
200   380e.4d9b.6a4a                       0x8002  0  42949  64  0x7f40e15fd308    0x7f40e15f49d8
      0x0                                0x0
200   a0f8.4910.bce2                       0x1000001  0  0      64  0x7f40e193c478    0x7f40e1938168
      0x7f40e1937bf8                       0x0
                                           0      0      RLOC 10.1.1.10 adj_id 86

Total Mac number of addresses:: 2
```

## Verifying EVPN VXLAN Layer 3 Overlay Network

The following table lists the **show** commands that are used to verify a Layer 3 VXLAN overlay network:

**Table 4: Commands to Verify EVPN VXLAN Layer 3 Overlay Network**

Command	Purpose
<b>show nve vni</b>	Displays information about VXLAN network identifier members associated with an NVE interface.
<b>show nve vni vni-id detail</b>	Displays detailed NVE interface state information for a VXLAN network identifier member.
<b>show nve peers</b>	Displays NVE interface state information for peer leaf switches.
<b>show mac address-table vlan vlan-id</b>	Displays MAC addresses for a VLAN.
<b>show platform software fed switch active matm macTable vlan vlan-id</b>	Displays MAC addresses for a VLAN from MAC address table manager database for Forwarding Engine Driver (FED).
<b>show ip route vrf vrf-name</b>	Displays the IP routing table associated with a specific VRF.
<b>show ip cef vrf vrf-name</b>	Displays entries in the Cisco Express Forwarding (CEF) table associated with a VRF.
<b>show arp vrf vrf-name</b>	Displays entries in the Address Resolution Protocol (ARP) table associated with a VRF.
<b>show bgp l2vpn evpn route-type 5</b>	Displays BGP information for route type 5 of Layer 2 VPN EVPN address family.

Command	Purpose
<b>show bgp l2vpn evpn all</b>	Displays all BGP information for L2VPN EVPN address family.



## CHAPTER 4

# Configuring EVPN VXLAN Integrated Routing and Bridging

---

- [Restrictions for EVPN VXLAN Integrated Routing and Bridging, on page 53](#)
- [Information About EVPN VXLAN Integrated Routing and Bridging, on page 53](#)
- [How to Configure EVPN VXLAN Integrated Routing and Bridging, on page 59](#)
- [Configuration Examples for EVPN VXLAN Integrated Routing and Bridging, on page 68](#)
- [Verifying EVPN VXLAN Integrated Routing and Bridging, on page 92](#)

## Restrictions for EVPN VXLAN Integrated Routing and Bridging

This section provides restrictions for both EVPN VXLAN distributed anycast gateway and centralized default gateway functionalities that are used to enable integrated routing and bridging (IRB).

### EVPN VXLAN Distributed Anycast Gateway

The same subnet mask and IP address must be configured on all the switch virtual interfaces (SVIs) that act as a distributed anycast gateway (DAG).

### EVPN VXLAN Centralized Default Gateway

- Only one centralized gateway (CGW) leaf switch or VTEP is supported in an EVPN VXLAN network.
- HSRP and VRRP are not supported for the EVPN VXLAN Layer 2 overlay networks when you use centralized gateway.

## Information About EVPN VXLAN Integrated Routing and Bridging

EVPN VXLAN integrated routing and bridging (IRB) allows the VTEPs or leaf switches in an EVPN VXLAN network to perform both bridging and routing. IRB allows the VTEPs to forward both Layer 2 or bridged and Layer 3 or routed traffic. A VTEP performs bridging when it forwards traffic to the same subnet. Similarly, a VTEP performs routing when it forwards traffic to a different subnet. The VTEPs in the network forward traffic to each other through the VXLAN gateways. BGP EVPN VXLAN implements IRB in two ways:

- Asymmetric IRB

- Symmetric IRB

### Asymmetric IRB

In asymmetric IRB, the ingress VTEP performs both bridging and routing whereas the egress VTEP performs only bridging. A packet first moves through a MAC VRF followed by an IP VRF on the network virtualisation endpoint (NVE) of the ingress VTEP. It then moves only through a MAC VRF on the NVE of the egress VTEP. The NVE of the ingress VTEP manages all the packet processing associated with intersubnet forwarding semantics.

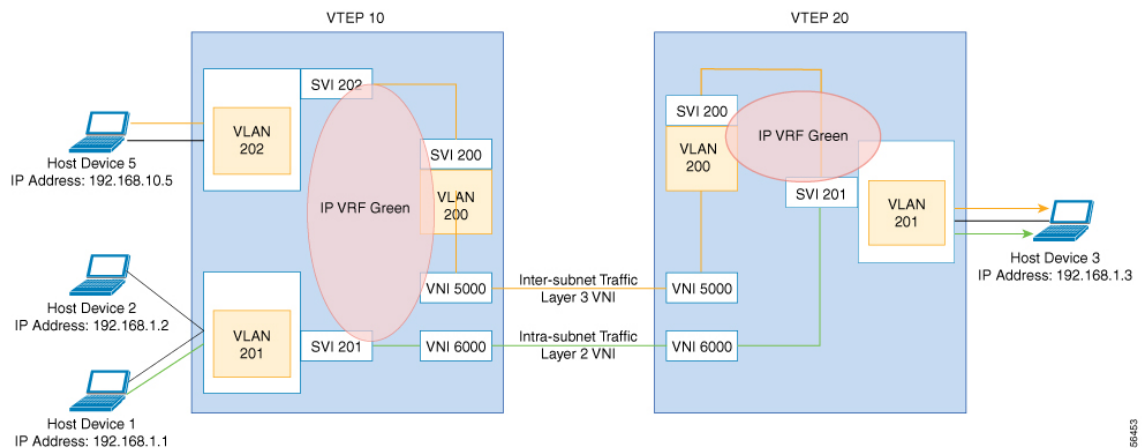
The return traffic during asymmetric IRB goes through a different virtual network instance (VNI) compared to the source traffic. Asymmetric IRB needs the source and destination VNIs to be associated with both the ingress and egress VTEPs.

### Symmetric IRB

In symmetric IRB, both the ingress and egress VTEPs perform both bridging and routing. A packet first moves through a MAC VRF followed by an IP VRF on the NVE of the ingress VTEP. It then moves through an IP VRF followed by a MAC VRF on the NVE of the egress VTEP. The NVEs of ingress and egress VTEPs equally share all the packet processing associated with intersubnet forwarding semantics.

In symmetric IRB, you are required to define only the VNIs of locally attached endpoints on the ingress and egress VTEPs. Symmetric IRB offers better scalability in terms of the number of VNIs that a BGP EVPN VXLAN fabric supports.

The following figure shows the implementation of symmetric IRB and the movement of traffic in an EVPN VXLAN network:

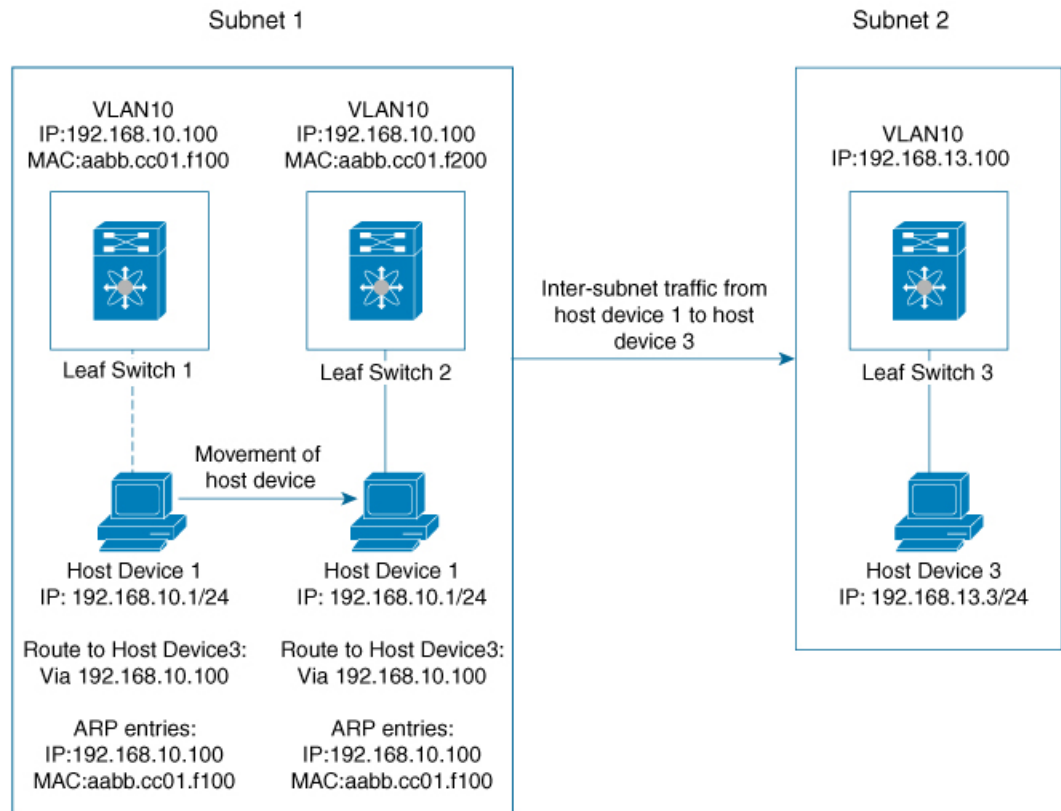


## EVPN VXLAN Distributed Anycast Gateway

Distributed anycast gateway is a default gateway addressing mechanism in a BGP EVPN VXLAN fabric. The feature enables the use of the same gateway IP and MAC address across all the VETPs in an EVPN VXLAN network. This ensures that every VTEP functions as the default gateway for the workloads directly connected to it. The feature facilitates flexible workload placement, host mobility, and optimal traffic forwarding across the BGP EVPN VXLAN fabric.

The scenario shown in the following figure depicts a distributed gateway. Subnet 1 contains two leaf switches, leaf switch 1 and leaf switch 2, acting together as a distributed default gateway for VLAN 10. Host device 1 is connected to leaf switch 1 and needs to send traffic to host device 3, which is in a different subnet. When host device 1 tries to send traffic outside of subnet 1, the traffic goes through the configured gateway on leaf switch 1. Host device 1 registers the Address Resolution Protocol (ARP) entries of the gateway VLAN MAC and IP addresses on leaf switch 1.

Figure 1: Distributed Gateway Topology



When multiple VTEPs act together as one single distributed default gateway for the same VLAN, the VLAN IP address remains the same across all of them. This IP address becomes the gateway IP address for any host device in the VLAN that tries to reach an IP address outside its subnet. But, each VTEP retains its own MAC address.

In the preceding figure, consider the scenario where host device 1 moves from leaf switch 1 to leaf switch 2. The host device remains within the same network and still maintains the same ARP entries for gateway MAC and IP addresses. But the MAC addresses of the VLAN interfaces on leaf switch 2 and leaf switch 1 are different. This results in a MAC address mismatch between the ARP entry and the VLAN on leaf switch 2. As a result, any traffic that host device 1 tries to send outside of Subnet 1 is either lost or continuously flooded as unknown unicast. EVPN VXLAN distributed anycast gateway feature prevents this traffic loss by ensuring that all the VTEPs have the same gateway MAC and IP addresses.

Manual MAC address configuration and MAC aliasing are the two methods used to maintain the same MAC address across all VTEPs and configure distributed anycast gateway.

## Manual MAC Address Configuration

Manual MAC address configuration is the conventional method of enabling distributed anycast gateway in an EVPN VXLAN network. In this method, you manually configure the same MAC address on the Layer 2 VNI VLAN SVI on all the VTEPs in the network. You must configure the same MAC address on all the VTEPs in the same Layer 2 VNI.



### Note

The VLAN SVIs on all the leaf switches must already share the same gateway IP address.

In the [Figure 1: Distributed Gateway Topology, on page 55](#) image, to enable distributed anycast gateway in subnet 1, configure the same MAC address on leaf switch 1 and leaf switch 2. This ensures that the ARP entries of gateway MAC and IP addresses on host device 1 match with the MAC and IP addresses of both leaf switch 1 and leaf switch 2.

## MAC Aliasing

MAC aliasing for distributed anycast gateway removes the need to configure the same MAC address explicitly on the VLAN interfaces of every VTEP. MAC aliasing allows the VTEPs to advertise their VLAN MAC addresses as the gateway MAC addresses to all the other VTEPs in the network. The VTEPs in the network store the advertised MAC address as a gateway MAC address provided their VLAN IP address matches with the gateway IP address.

In the [Figure 1: Distributed Gateway Topology, on page 55](#) image, consider the scenario where MAC aliasing is enabled in subnet 1. Leaf switch 1 and leaf switch 2 advertise their MAC addresses to each other as gateway MAC addresses. This allows leaf switch 2 to recognize the MAC address in the ARP entry of host device 1 as a gateway MAC address. It allows host device 1 to send traffic outside of subnet 1 even though its VLAN MAC address does not match with the ARP entry.

MAC aliasing in an EVPN VXLAN network is configured by enabling the default gateway advertisement on all the VTEPs.

## EVPN VXLAN Centralized Default Gateway

In this scenario, the network has a CGW VTEP that performs the Layer 3 gateway function for all the Layer 2 VNIs. All the other VTEPs in the network perform only bridging. The CGW VTEP acts as the Layer 3 gateway and performs routing for the inter-subnet VXLAN traffic.

The CGW VTEP advertises the SVI MAC-IP route for a particular VXLAN-enabled VLAN to all other Layer 2 VTEPs that have the same Layer 2 VNI configured. This allows the VTEPs to import and install the remote SVI MAC-IP route as a VXLAN Layer 3 gateway address. A host device uses the address of an SVI in the same VLAN on the CGW VTEP as its gateway address. Configure the SVI for the Layer 2 VNI VLAN only on the CGW VTEP. Do not configure the SVI (for the respective Layer 2 VNI VLAN) on any other VTEP in the network that acts as a Layer 2 VTEP.

When a host device connected to a Layer 2 VTEP sends traffic to a different subnet, the traffic is bridged from the Layer 2 VTEP to the CGW VTEP. The CGW VTEP then routes the traffic to the destination subnet. The destination subnet can be another VXLAN-enabled VLAN or an external route.

If the CGW VTEP needs to route the traffic between 2 VXLAN-enabled VLANs, then configure the CGW on the same VTEP for both VLANs. In other words, configure the SVI on the same VTEP for both VLANs.





- 
- Note** We recommend that you configure a centralized default gateway in an EVPN VXLAN network if:
- You require a boundary between Layer 2 and Layer 3 segments at the border of the BGP EVPN VXLAN fabric.
  - The inter-subnet traffic is subjected to a firewall inspection or any policy on a centralized plane.
- 



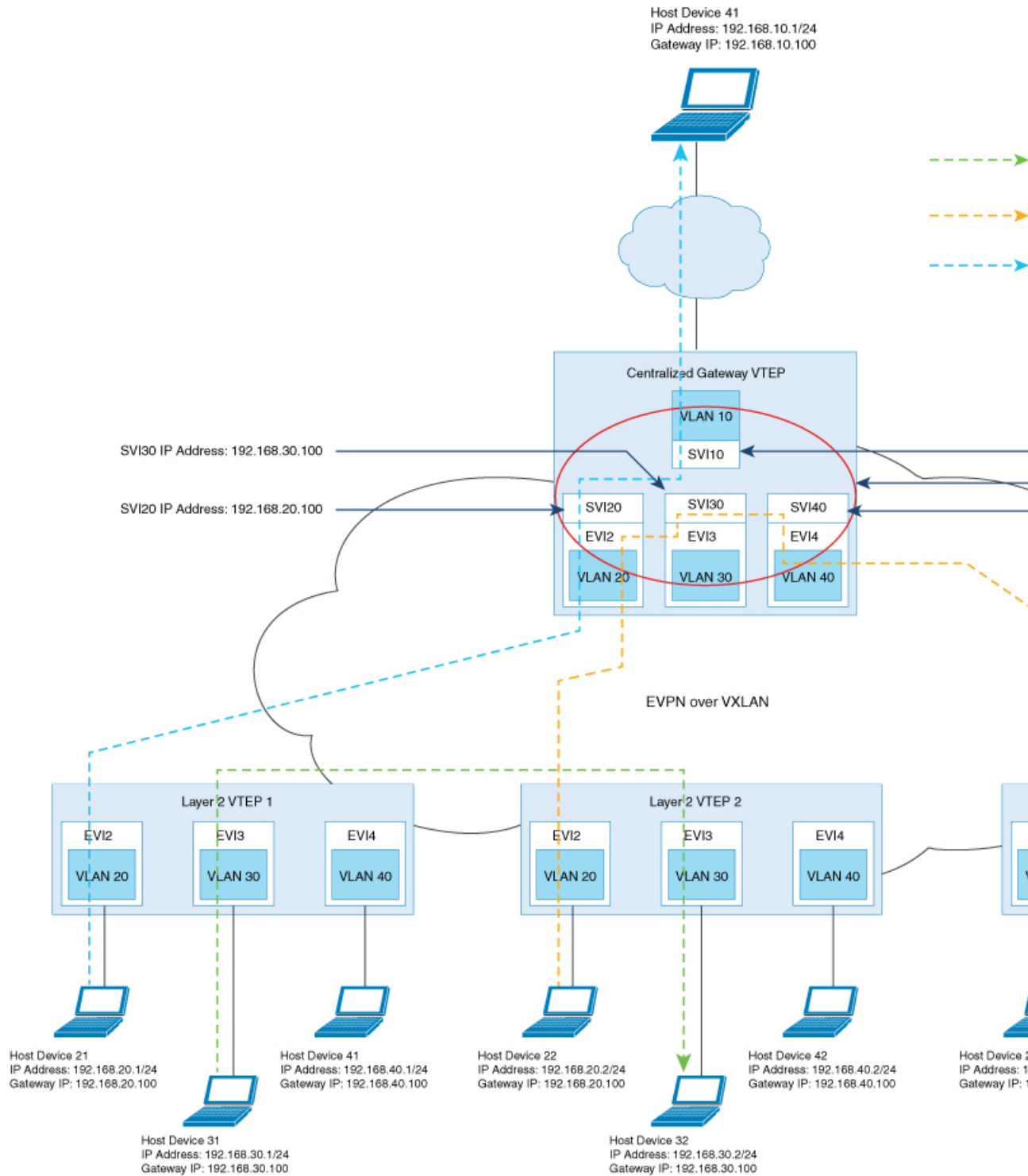
---

**Note** Toggling between DAG and CGW on an SVI in a VLAN disrupts the traffic for that VLAN.

---

The following image shows an EVPN VXLAN network with centralized default gateway configured:

Figure 2: An EVPN VXLAN Network with Centralized Default Gateway



## Default Gateway MAC Address Assignment

When leaf switches import gateway addresses, it can result in a conflict if the SVI of a leaf switch has the same IP and MAC address as the imported addresses. To avoid this conflict, the SVI MAC-IP routes are tagged with the Default Gateway Extended Community attribute. The attribute helps the receiving leaf switches to distinguish the MAC-IP routes of the SVIs from the MAC-IP routes of the host devices. When a leaf switch receives a route tagged with the attribute, it results in one of the following scenarios:

- If the leaf switch does not have a local SVI for the same MAC VRF, it installs the route only as a remote MAC route. The leaf switch implements centralized gateway functionality in this scenario.
- If the leaf switch has a local SVI with a matching IP address but different MAC address, it installs the MAC route as a route that points to the local SVI. The leaf switch implements MAC aliasing for distributed anycast gateway in this scenario.
- If the leaf switch has an SVI with no matching IP address, it invalidates the MAC-IP route and issues an error. See [RFC4732](#) for more details about the error.

## How to Configure EVPN VXLAN Integrated Routing and Bridging

To configure EVPN VXLAN IRB, you need to configure EVPN VXLAN Layer 2 and Layer 3 overlay networks, and enable the gateways in the VXLAN network.

### Configuring EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway

To enable IRB in an EVPN VXLAN network using distributed anycast gateway, perform the following set of procedures:

- Configure Layer 2 VPN EVPN on the VTEPs.  
Enable distributed anycast gateway for the VXLAN network when you configure Layer 2 VPN.
- Configure the core-facing and access-facing VLANs on the VTEPs.
- Configure switch virtual interface (SVI) for the core-facing VLAN on the VTEPs.
- Configure SVI for the access-facing VLAN on the VTEPs.
- Configure the IP VRF on the VTEPs.
- Configure the Loopback interface on the VTEPs.
- Configure the Network Virtualization Endpoint (NVE) interface on the VTEPs.
- Configure BGP with EVPN address family on the VTEPs.

### Configuring Layer 2 VPN EVPN on a VTEP

See [Configuring Layer 2 VPN EVPN on a VTEP, on page 14](#) for detailed steps.

## Configuring IP VRF on VTEP

See [Configuring an IP VRF on a VTEP, on page 34](#) for detailed steps.

## Configuring the Loopback Interface on a VTEP

See [Configuring the Loopback Interface on a VTEP, on page 36](#) for detailed steps.

## Configuring Core-facing and Access-facing VLANs on a VTEP

To configure the core-facing and access-facing VLANs on a VTEP and enable IRB in the EVPN VXLAN network, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>vlan configuration</b> <i>vlan-id</i>  <b>Example:</b> Device(config)# <b>vlan configuration 201</b>	Enters VLAN feature configuration mode for the specified VLAN interface.
<b>Step 4</b>	<b>member evpn-instance</b> <i>evpn-instance-id</i> <b>vni</b> <i>l2-vni-number</i>  <b>Example:</b> Device(config-vlan)# <b>member evpn-instance 1 vni 6000</b>	Adds EVPN instance as a member of the VLAN configuration.  The VNI here is used as a Layer 2 VNI.
<b>Step 5</b>	<b>exit</b>  <b>Example:</b> Device(config-vlan)# <b>exit</b>	Returns to global configuration mode.
<b>Step 6</b>	<b>vlan configuration</b> <i>vlan-id</i>  <b>Example:</b> Device(config)# <b>vlan configuration 202</b>	Enters VLAN feature configuration mode for the specified VLAN interface.
<b>Step 7</b>	<b>member evpn-instance</b> <i>evpn-instance-id</i> <b>vni</b> <i>l2-vni-number</i>  <b>Example:</b> Device(config-vlan)# <b>member evpn-instance 2 vni 7000</b>	Adds EVPN instance as a member of the VLAN configuration.  The VNI here is used as a Layer 2 VNI.

	Command or Action	Purpose
<b>Step 8</b>	<b>exit</b> <b>Example:</b> Device(config-vlan)# <b>exit</b>	Returns to global configuration mode.
<b>Step 9</b>	<b>vlan configuration <i>vlan-id</i></b> <b>Example:</b> Device(config)# <b>vlan configuration 200</b>	Enters VLAN feature configuration mode for the specified VLAN interface.
<b>Step 10</b>	<b>member vni <i>l3-vni-number</i></b> <b>Example:</b> Device(config-vlan)# <b>member vni 5000</b>	Adds EVPN instance as a member of the VLAN configuration. The VNI here is used as a Layer 3 VNI.
<b>Step 11</b>	<b>exit</b> <b>Example:</b> Device(config-vlan)# <b>exit</b>	Returns to global configuration mode.
<b>Step 12</b>	<b>end</b> <b>Example:</b> Device(config-vlan)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring Switch Virtual Interface for the Core-facing VLAN on a VTEP

To configure an SVI for the core-facing VLAN on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface vlan <i>vlan-id</i></b> <b>Example:</b> Device(config)# <b>interface vlan 200</b>	Enters interface configuration mode for the specified VLAN.
<b>Step 4</b>	<b>vrf forwarding <i>vrf-name</i></b> <b>Example:</b> Device(config-if)# <b>vrf forwarding Green</b>	Configures the SVI for the VLAN.

	Command or Action	Purpose
<b>Step 5</b>	<b>ip unnumbered</b> <i>Loopback-interface</i> <b>Example:</b> Device(config-if)# <b>ip unnumbered</b> Loopback0	Enables IP processing on the Loopback interface without assigning an explicit IP address to the interface.
<b>Step 6</b>	<b>no autostate</b> <b>Example:</b> Device(config-if)# <b>no autostate</b>	Disables autostate on the interface.  In EVPN deployments, once a VLAN is used for a core-facing SVI, it should not be allowed in any trunk. For a core-facing SVI to function properly, the <b>no autostate</b> command must be configured under the SVI.
<b>Step 7</b>	<b>end</b> <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring Switch Virtual Interface for the Access-facing VLANs on a VTEP

To configure SVIs for the access-facing VLANs on a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface vlan</b> <i>vlan-id</i> <b>Example:</b> Device(config)# <b>interface vlan 202</b>	Enters interface configuration mode for the specified VLAN.
<b>Step 4</b>	<b>vrf forwarding</b> <i>vrf-name</i> <b>Example:</b> Device(config-if)# <b>vrf forwarding Green</b>	Configures the SVI for the VLAN.
<b>Step 5</b>	<b>ip address</b> <i>gateway-ip-address</i> <b>Example:</b> Device(config-if)# <b>ip address</b> 192.168.10.1 255.255.255.0	Configures the gateway IP address for the access SVI.  Configure the same gateway IP address for this SVI on all the other VTEPs.

	Command or Action	Purpose
<b>Step 6</b>	<b>mac-address</b> <i>mac-address-value</i> <b>Example:</b> Device(config-if) # <b>mac-address</b> <b>aabb.cc01.f100</b>	(Optional) Manually sets the MAC address for the VLAN interface.  To configure distributed anycast gateway in a VXLAN network using manual MAC configuration, configure the same MAC address on the corresponding Layer 2 VNI SVIs on all the VTEPs in a VXLAN network.
<b>Step 7</b>	<b>end</b> <b>Example:</b> Device(config-if) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring the NVE Interface on a VTEP

To add Layer 2 and Layer 3 VNI members to the NVE interface of a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>nve-interface-id</i> <b>Example:</b> Device(config) # <b>interface nve1</b>	Defines the interface to be configured as a trunk, and enters interface configuration mode.
<b>Step 4</b>	<b>no ip address</b> <b>Example:</b> Device(config-if) # <b>no ip address</b>	Disables IP processing on the interface by removing its IP address.
<b>Step 5</b>	<b>source-interface</b> <i>loopback-interface-id</i> <b>Example:</b> Device(config-if) # <b>source-interface</b> <b>loopback0</b>	Sets the IP address of the specified loopback interface as the source IP address.
<b>Step 6</b>	<b>host-reachability protocol bgp</b> <b>Example:</b> Device(config-if) # <b>host-reachability</b> <b>protocol bgp</b>	Configures BGP as the host-reachability protocol on the interface.

	Command or Action	Purpose
<b>Step 7</b>	<b>member vni</b> <i>layer2-vni-id</i> { <b>ingress-replication</b>   <b>mcast-group</b> <i>multicast-group-address</i>  <b>Example:</b> Device(config-if)# <b>member vni 6000 mcast-group 227.0.0.1</b>  Device(config-if)# <b>member vni 7000 mcast-group 227.0.0.1</b>	Associates the Layer 2 VNI member with the NVE.  The specified replication type must match the replication type that is configured globally or for the specific EVPN instance. Use <b>mcast-group</b> keyword for static replication and <b>ingress-replication</b> keyword for ingress replication.
<b>Step 8</b>	<b>member vni</b> <i>layer3-vni-id</i> <b>vrf</b> <i>vrf-name</i>  <b>Example:</b> Device(config-if)# <b>member vni 5000 vrf Green</b>	Associates the Layer 3 VNI member with the NVE.
<b>Step 9</b>	<b>end</b>  <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring BGP with EVPN and VRF Address Families on a VTEP

To configure BGP on a VTEP with EVPN and VRF address families and a spine switch as the neighbor, perform these steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>router bgp</b> <i>autonomous-system-number</i>  <b>Example:</b> Device(config)# <b>router bgp 1</b>	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
<b>Step 4</b>	<b>bgp log-neighbor-changes</b>  <b>Example:</b> Device(config-router)# <b>bgp log-neighbor-changes</b>	(Optional) Enables the generation of logging messages when the status of a BGP neighbor changes.  For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .



	Command or Action	Purpose
Step 5	<b>bgp update-delay</b> <i>time-period</i> <b>Example:</b> Device(config-router) # <b>bgp update-delay</b> <b>1</b>	(Optional) Sets the maximum initial delay period before sending the first update. For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
Step 6	<b>bgp graceful-restart</b> <b>Example:</b> Device(config-router) # <b>bgp graceful-restart</b>	(Optional) Enables the BGP graceful restart capability for all BGP neighbors. For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
Step 7	<b>no bgp default ipv4-unicast</b> <b>Example:</b> Device(config-router) # <b>no bgp default ipv4-unicast</b>	(Optional) Disables default IPv4 unicast address family for BGP peering session establishment. For more information, see <i>Configuring BGP</i> module of the <i>IP Routing Configuration Guide</i> .
Step 8	<b>neighbor ip-address remote-as number</b> <b>Example:</b> Device(config-router) # <b>neighbor 10.11.11.11 remote-as 1</b>	Defines multiprotocol-BGP neighbors. Under each neighbor, define the Layer 2 Virtual Private Network (L2VPN) EVPN configuration. Use the IP address of the spine switch as the neighbor IP address.
Step 9	<b>neighbor {ip-address   group-name} update-source interface</b> <b>Example:</b> Device(config-router) # <b>neighbor 10.11.11.11 update-source Loopback0</b>	Configures update source. Update source can be configured per neighbor or per peer-group. Use the IP address of the spine switch as the neighbor IP address.
Step 10	<b>address-family l2vpn evpn</b> <b>Example:</b> Device(config-router) # <b>address-family l2vpn evpn</b>	Specifies the L2VPN address family and enters address family configuration mode.
Step 11	<b>neighbor ip-address activate</b> <b>Example:</b> Device(config-router-af) # <b>neighbor 10.11.11.11 activate</b>	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 12	<b>neighbor ip-address send-community [both   extended   standard]</b> <b>Example:</b> Device(config-router-af) # <b>neighbor 10.11.11.11 send-community both</b>	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 13	<b>exit-address-family</b> <b>Example:</b>	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
	Device (config-router-af) # <b>exit-address-family</b>	
<b>Step 14</b>	<b>address-family ipv4 vrf vrf-name</b>  <b>Example:</b> Device (config-router) # <b>address-family ipv4 vrf green</b>	Specifies the IPv4 address family and enters address family configuration mode.
<b>Step 15</b>	<b>advertise l2vpn evpn</b>  <b>Example:</b> Device (config-router-af) # <b>advertise l2vpn evpn</b>	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.
<b>Step 16</b>	<b>redistribute connected</b>  <b>Example:</b> Device (config-router-af) # <b>redistribute connected</b>	Redistributes connected routes to BGP.
<b>Step 17</b>	<b>redistribute static</b>  <b>Example:</b> Device (config-router-af) # <b>redistribute static</b>	Redistributes static routes to BGP.
<b>Step 18</b>	<b>exit-address-family</b>  <b>Example:</b> Device (config-router-af) # <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 19</b>	<b>address-family ipv6 vrf vrf-name</b>  <b>Example:</b> Device (config-router) # <b>address-family ipv6 vrf green</b>	Specifies the IPv6 address family and enters address family configuration mode.
<b>Step 20</b>	<b>advertise l2vpn evpn</b>  <b>Example:</b> Device (config-router-af) # <b>advertise l2vpn evpn</b>	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.
<b>Step 21</b>	<b>redistribute connected</b>  <b>Example:</b> Device (config-router-af) # <b>redistribute connected</b>	Redistributes connected routes to BGP.
<b>Step 22</b>	<b>redistribute static</b>  <b>Example:</b> Device (config-router-af) # <b>redistribute static</b>	Redistributes static routes to BGP.

	Command or Action	Purpose
Step 23	<b>exit-address-family</b> <b>Example:</b> Device(config-router-af) # <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
Step 24	<b>end</b> <b>Example:</b> Device(config-router) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway

This section provides information about how to configure EVPN VXLAN IRB using centralized default gateway. Centralized default gateway implements asymmetric IRB in the EVPN VXLAN network. To enable IRB using centralized default gateway in an EVPN VXLAN network, perform these steps:

- On the centralized gateway VTEP, perform these procedures:
  - [Configuring Layer 2 VPN EVPN on a VTEP, on page 59](#)
  - [Configuring the Loopback Interface on a VTEP, on page 60](#)
  - [Configuring Core-facing and Access-facing VLANs on a VTEP, on page 60](#)
  - [Configuring Switch Virtual Interface for the Core-facing VLAN on a VTEP, on page 61](#)
  - [Configuring Switch Virtual Interface for the Access-facing VLANs on a VTEP, on page 62](#)
  - [Configuring the NVE Interface on a VTEP, on page 63](#)
  - [Configuring BGP with EVPN and VRF Address Families on a VTEP, on page 64](#)



**Note** Do not configure a Layer 3 VNI on the centralized gateway VTEP for the specific VRF in which the Layer 2 VNIs are configured.



**Note** In order to set the SVI state to up for the respective EVPN instance, run the **no autostate** command in the VLAN configuration mode for the SVI.



**Note** The centralized gateway leaf switch can be configured either as a single switch or a stack of switches..

- On all the other VTEPs, configure only EVPN VXLAN Layer 2 overlay network. To configure EVPN VXLAN Layer 2 overlay network on the leaf switches, perform all the procedures listed in [How to Configure EVPN VXLAN Layer 2 Overlay Network](#), on page 14.



**Note** As Layer 2 VTEPs perform only bridging, do not configure the SVIs on the non-centralized-gateway VTEPs.

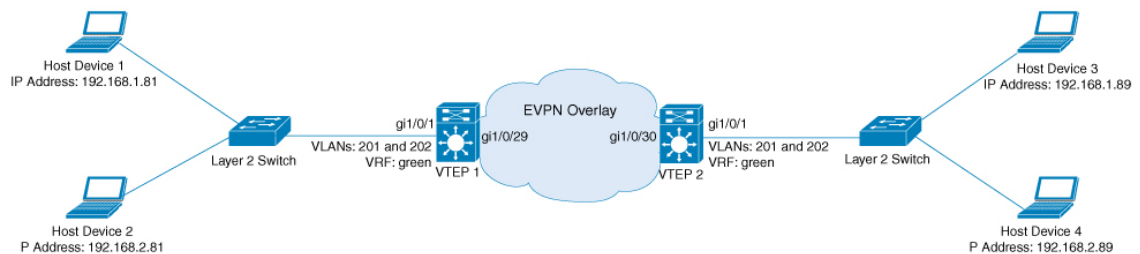
## Configuration Examples for EVPN VXLAN Integrated Routing and Bridging

This section provides examples to show how EVPN VXLAN IRB is enabled using distributed anycast gateway and centralized default gateway.

### Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway

This section provides an example to show how to enable EVPN VXLAN IRB using distributed anycast gateway. The following example shows a sample configuration for an EVPN VXLAN network with 2 VTEPs. VTEP 1 and VTEP 2 are connected to perform integrated routing and bridging.

**Figure 3: EVPN VXLAN Topology for IRB using Distributed Anycast Gateway**



**Note** In a two-VTEP topology, a spine switch is not mandatory. For information about configuration of spine switches in an EVPN VXLAN network, see *Configuring Spine Switches in a BGP EVPN VXLAN Fabric* module.

*Table 5: Configuration Example for an EVPN VXLAN Network with Two VTEPs Connected to Perform Integrated Routing and Bridging Using Distributed Anycast Gateway*

VTEP 1	VTEP 2
--------	--------

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway

VTEP 1	VTEP 2
<pre> VTEP1# show running-config ! hostname VTEP1 ! vrf definition green  rd 103:2  !  address-family ipv4   route-target export 103:2   route-target import 104:2   route-target export 103:2 stitching   route-target import 104:2 stitching  exit-address-family  !  address-family ipv6   route-target export 103:2   route-target import 104:2   route-target export 103:2 stitching   route-target import 104:2 stitching  exit-address-family  ! ip routing ip multicast-routing ipv6 unicast-routing ! ! l2vpn evpn  replication-type static  router-id Loopback0  default-gateway advertise  ! l2vpn evpn instance 1 vlan-based  encapsulation vxlan  ! l2vpn evpn instance 2 vlan-based  encapsulation vxlan  ! ! system mtu 9150 ! vlan configuration 200  member vni 5000 vlan configuration 201  member evpn-instance 1 vni 6000 vlan configuration 202  member evpn-instance 2 vni 7000 ! ! interface Loopback0  ip address 10.1.1.10 255.255.255.255  ip pim sparse-mode ! interface Loopback13  description demo only (for rt5 distribution)  vrf forwarding green  ip address 10.1.13.13 255.255.255.0 ! interface GigabitEthernet1/0/1  description access-facing-interface  switchport trunk allowed vlan 201,202  switchport mode trunk ! </pre>	<pre> VTEP2# show running-config ! hostname VTEP2 ! vrf definition green  rd 104:2  !  address-family ipv4   route-target export 104:2   route-target import 103:2   route-target export 104:2 stitching   route-target import 103:2 stitching  exit-address-family  !  address-family ipv6   route-target export 104:2   route-target import 103:2   route-target export 104:2 stitching   route-target import 103:2 stitching  exit-address-family  ! ip routing ip multicast-routing ipv6 unicast-routing ! ! l2vpn evpn  replication-type static  router-id Loopback0  default-gateway advertise  ! l2vpn evpn instance 1 vlan-based  encapsulation vxlan  ! l2vpn evpn instance 2 vlan-based  encapsulation vxlan  ! ! system mtu 9150 ! vlan configuration 200  member vni 5000 vlan configuration 201  member evpn-instance 1 vni 6000 vlan configuration 202  member evpn-instance 2 vni 7000 ! ! interface Loopback0  ip address 10.2.2.20 255.255.255.255  ip pim sparse-mode ! interface Loopback14  description demo only (for rt5 distribution)  vrf forwarding green  ip address 10.1.14.14 255.255.255.0 ! interface GigabitEthernet1/0/1  description access-facing-interface  switchport trunk allowed vlan 201,202  switchport mode trunk ! </pre>

VTEP 1	VTEP 2
<pre> ! interface GigabitEthernet1/0/29 description core-underlay-interface no switchport ip address 172.16.1.29 255.255.255.0 ip pim sparse-mode ! ! interface Vlan200 description core svi for l3vni vrf forwarding green ip unnumbered Loopback0 ipv6 enable no autostate ! interface Vlan201 description vni 6000 default-gateway vrf forwarding green ip address 192.168.1.201 255.255.255.0 ipv6 address 2001:DB8:201::201/64 ipv6 enable ! interface Vlan202 description vni 7000 default-gateway vrf forwarding green ip address 192.168.2.202 255.255.255.0 ipv6 address 2001:DB8:202::202/64 ipv6 enable ! ! interface nve10 no ip address source-interface Loopback0 host-reachability protocol bgp member vni 6000 mcast-group 232.1.1.1 member vni 5000 vrf green member vni 7000 mcast-group 232.1.1.1 ! router ospf 1 router-id 10.1.1.10 network 10.1.1.0 0.0.0.255 area 0 network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10 bgp router-id interface Loopback0 bgp log-neighbor-changes bgp update-delay 1 no bgp default ipv4-unicast neighbor 10.2.2.20 remote-as 10 neighbor 10.2.2.20 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 10.2.2.20 activate neighbor 10.2.2.20 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected </pre>	<pre> ! interface GigabitEthernet1/0/30 description core-underlay-interface no switchport ip address 172.16.1.30 255.255.255.0 ip pim sparse-mode ! ! interface Vlan200 description core svi for l3vni vrf forwarding green ip unnumbered Loopback0 ipv6 enable no autostate ! interface Vlan201 description vni 6000 default-gateway vrf forwarding green ip address 192.168.1.201 255.255.255.0 ipv6 address 2001:DB8:201::201/64 ipv6 enable ! interface Vlan202 description vni 7000 default-gateway vrf forwarding green ip address 192.168.2.202 255.255.255.0 ipv6 address 2001:DB8:202::202/64 ipv6 enable ! ! interface nve10 no ip address source-interface Loopback0 host-reachability protocol bgp member vni 6000 mcast-group 232.1.1.1 member vni 7000 mcast-group 232.1.1.1 member vni 5000 vrf green ! router ospf 1 router-id 10.2.2.20 network 10.2.2.0 0.0.0.255 area 0 network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10 bgp router-id interface Loopback0 bgp log-neighbor-changes bgp update-delay 1 no bgp default ipv4-unicast neighbor 10.1.1.10 remote-as 10 neighbor 10.1.1.10 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 10.1.1.10 activate neighbor 10.1.1.10 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected </pre>

VTEP 1	VTEP 2
<pre> redistribute static exit-address-family ! address-family ipv6 vrf green   redistribute connected   redistribute static   advertise l2vpn evpn exit-address-family ! ip pim rp-address 10.1.1.10 ! end </pre>	<pre> redistribute static exit-address-family ! address-family ipv6 vrf green   redistribute connected   redistribute static   advertise l2vpn evpn exit-address-family ! ip pim rp-address 10.1.1.10 ! end </pre>

The following examples provide outputs for **show** commands on VTEP 1 and VTEP 2 in the topology configured above:

- [show nve peers, on page 72](#)
- [show l2vpn evpn peers vxlan, on page 73](#)
- [show l2vpn evpn evi evpn-instance detail, on page 73](#)
- [show l2vpn evpn default-gateway, on page 74](#)
- [show bgp l2vpn evpn all, on page 75](#)
- [show ip route vrf green, on page 78](#)
- [show platform software fed switch active matm mactable vlan, on page 79](#)

### show nve peers

#### VTEP 1

The following example shows the output for the **show nve peers** command on VTEP 1:

```

VTEP1# show nve peers
Interface VNI      Type Peer-IP      RMAC/Num_RTs  eVNI      state flags UP time
nve10    5000    L3CP 10.2.2.20    380e.4d9b.6a4a 5000      UP   A/M/4 01:33:41
nve10    5000    L3CP 10.2.2.20    380e.4d9b.6a4a 5000      UP   A/-/6 00:43:38
nve10    6000    L2CP 10.2.2.20      5           6000      UP   N/A   01:33:41
nve10    7000    L2CP 10.2.2.20      6           7000      UP   N/A   01:33:41

```

#### VTEP 2

The following example shows the output for the **show nve peers** command on VTEP 2:

```

VTEP2# show nve peers
Interface VNI      Type Peer-IP      RMAC/Num_RTs  eVNI      state flags UP time
nve10    5000    L3CP 10.1.1.10      a0f8.4910.bce2 5000      UP   A/M/4 01:33:55
nve10    5000    L3CP 10.1.1.10      a0f8.4910.bce2 5000      UP   A/-/6 01:14:23
nve10    6000    L2CP 10.1.1.10      7           6000      UP   N/A   01:33:55
nve10    7000    L2CP 10.1.1.10      6           7000      UP   N/A   01:33:55

```



**show l2vpn evpn peers vxlan****VTEP 1**

The following example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 1:

```
VTEP1# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP              Num routes eVNI      UP time
-----
nve10     6000      10.2.2.20           5          6000     01:34:50
nve10     7000      10.2.2.20           6          7000     01:34:50
```

**VTEP 2**

The following example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 2:

```
VTEP2# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP              Num routes eVNI      UP time
-----
nve10     6000      10.1.1.10           7          6000     01:35:23
nve10     7000      10.1.1.10           6          7000     01:35:23
```

**show l2vpn evpn evi evpn-instance detail****VTEP 1**

The following example shows the output for the **show l2vpn evpn evi evpn-instance detail** command on VTEP 1:

```
VTEP1# show l2vpn evpn evi 1 detail
EVPN instance:      1 (VLAN Based)
RD:                 10.1.1.10:1 (auto)
Import-RTs:         10:1
Export-RTs:         10:1
Per-EVI Label:     none
State:              Established
Replication Type:   Static (global)
Encapsulation:      vxlan
IP Local Learn:     Enable (global)
Vlan:               201
  Ethernet-Tag:     0
  State:            Established
  Core If:          Vlan200
  Access If:        Vlan201
  NVE If:           nve10
  RMAC:             a0f8.4910.bce2
  Core Vlan:        200
  L2 VNI:           6000
  L3 VNI:           5000
  VTEP IP:          10.1.1.10
  MCAST IP:         232.1.1.1
  VRF:              green
  IPv4 IRB:         Enabled
  IPv6 IRB:         Enabled
Pseudoports:
  GigabitEthernet1/0/1 service instance 201
```

**VTEP 2**

The following example shows the output for the **show l2vpn evpn evi evpn-instance detail** command on VTEP 2:

```
VTEP2# show l2vpn evpn evi 1 detail
EVPN instance:      1 (VLAN Based)
RD:                 10.2.2.20:1 (auto)
Import-RTs:         10:1
Export-RTs:         10:1
Per-EVI Label:      none
State:              Established
Replication Type:   Static (global)
Encapsulation:      vxlan
IP Local Learn:     Enable (global)
Vlan:               201
  Ethernet-Tag:     0
  State:            Established
  Core If:          Vlan200
  Access If:        Vlan201
  NVE If:           nve10
  RMAC:             380e.4d9b.6a4a
  Core Vlan:        200
  L2 VNI:           6000
  L3 VNI:           5000
  VTEP IP:          10.2.2.20
  MCAST IP:         232.1.1.1
  VRF:              green
  IPv4 IRB:         Enabled
  IPv6 IRB:         Enabled
Pseudoports:
  GigabitEthernet1/0/1 service instance 201
```

**show l2vpn evpn default-gateway****VTEP 1**

The following example shows the output for the **show l2vpn evpn default-gateway** command on VTEP 1:

```
VTEP1# show l2vpn evpn default-gateway
Valid Default Gateway Address  EVI  VLAN  MAC Address  Source
-----
Y  192.168.1.201                1    201   a0f8.4910.bccc V1201
Y  192.168.1.201                1    201   380e.4d9b.6a48 10.2.2.20
Y  2001:DB8:201::201            1    201   a0f8.4910.bccc V1201
Y  2001:DB8:201::201            1    201   380e.4d9b.6a48 10.2.2.20
Y  192.168.2.202                2    202   a0f8.4910.bcc2 V1202
Y  192.168.2.202                2    202   380e.4d9b.6a42 10.2.2.20
Y  2001:DB8:202::202            2    202   a0f8.4910.bcc2 V1202
Y  2001:DB8:202::202            2    202   380e.4d9b.6a42 10.2.2.20
```

**VTEP 2**

The following example shows the output for the **show l2vpn evpn default-gateway** command on VTEP 2:

```
VTEP2# show l2vpn evpn default-gateway
```

Valid	Default Gateway Address	EVI	VLAN	MAC Address	Source
Y	192.168.1.201	1	201	380e.4d9b.6a48	V1201
Y	192.168.1.201	1	201	a0f8.4910.bccc	10.1.1.10
Y	2001:DB8:201::201	1	201	380e.4d9b.6a48	V1201
Y	2001:DB8:201::201	1	201	a0f8.4910.bccc	10.1.1.10
Y	192.168.2.202	2	202	380e.4d9b.6a42	V1202
Y	192.168.2.202	2	202	a0f8.4910.bccc	10.1.1.10
Y	2001:DB8:202::202	2	202	380e.4d9b.6a42	V1202
Y	2001:DB8:202::202	2	202	a0f8.4910.bccc	10.1.1.10

```
show bgp l2vpn evpn all
```

## VTEP 1

The following example shows the output for the **show bgp l2vpn evpn all** command on VTEP 1:

```
VTEP1# show bgp l2vpn evpn all
```

```
BGP table version is 705, local router ID is 10.1.1.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Route	Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 10.1.1.10:1						
*>i	[2] [10.1.1.10:1] [0] [48] [0018736C56C3] [0] [*]/20	10.2.2.20	0	100	0	?
*>i	[2] [10.1.1.10:1] [0] [48] [0018736C56C3] [32] [192.168.1.89]/24	10.2.2.20	0	100	0	?
*>	[2] [10.1.1.10:1] [0] [48] [0059DC50AE01] [0] [*]/20	::			32768	?
*>	[2] [10.1.1.10:1] [0] [48] [0059DC50AE4C] [0] [*]/20	::			32768	?
*>	[2] [10.1.1.10:1] [0] [48] [0059DC50AE4C] [32] [192.168.1.81]/24	::			32768	?
*>	[2] [10.1.1.10:1] [0] [48] [0059DC50AE4C] [128] [2001:DB8:201::81]/36	::			32768	?
*>	[2] [10.1.1.10:1] [0] [48] [0059DC50AE4C] [128] [FE80::259:DCFF:FE50:AE4C]/36	::			32768	?
*>i	[2] [10.1.1.10:1] [0] [48] [380E4D9B6A48] [32] [192.168.1.201]/24	10.2.2.20	0	100	0	?
*>i	[2] [10.1.1.10:1] [0] [48] [380E4D9B6A48] [128] [2001:DB8:201::201]/36	10.2.2.20	0	100	0	?
*>	[2] [10.1.1.10:1] [0] [48] [A0F84910BCCC] [32] [192.168.1.201]/24	::			32768	?
*>	[2] [10.1.1.10:1] [0] [48] [A0F84910BCCC] [128] [2001:DB8:201::201]/36	::			32768	?
Route Distinguisher: 10.1.1.10:2						
*>i	[2] [10.1.1.10:2] [0] [48] [0018736C5681] [0] [*]/20	10.2.2.20	0	100	0	?
*>i	[2] [10.1.1.10:2] [0] [48] [0018736C56C2] [0] [*]/20	10.2.2.20	0	100	0	?
*>i	[2] [10.1.1.10:2] [0] [48] [0018736C56C2] [32] [192.168.2.89]/24	10.2.2.20	0	100	0	?
*>	[2] [10.1.1.10:2] [0] [48] [0059DC50AE01] [0] [*]/20	::			32768	?
*>	[2] [10.1.1.10:2] [0] [48] [0059DC50AE42] [0] [*]/20					

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway

```

::: 32768 ?
*> [2][10.1.1.10:2][0][48][0059DC50AE42][32][192.168.2.81]/24
::: 32768 ?
*>i [2][10.1.1.10:2][0][48][380E4D9B6A42][32][192.168.2.202]/24
10.2.2.20 0 100 0 ?
*>i [2][10.1.1.10:2][0][48][380E4D9B6A42][128][2001:DB8:202::202]/36
10.2.2.20 0 100 0 ?
*> [2][10.1.1.10:2][0][48][A0F84910BCC2][32][192.168.2.202]/24
::: 32768 ?
*> [2][10.1.1.10:2][0][48][A0F84910BCC2][128][2001:DB8:202::202]/36
::: 32768 ?
Route Distinguisher: 10.2.2.20:1
*>i [2][10.2.2.20:1][0][48][0018736C56C3][0][*]/20
10.2.2.20 0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0018736C56C3][32][192.168.1.89]/24
10.2.2.20 0 100 0 ?
*>i [2][10.2.2.20:1][0][48][380E4D9B6A48][32][192.168.1.201]/24
10.2.2.20 0 100 0 ?
*>i [2][10.2.2.20:1][0][48][380E4D9B6A48][128][2001:DB8:201::201]/36
10.2.2.20 0 100 0 ?
Route Distinguisher: 10.2.2.20:2
*>i [2][10.2.2.20:2][0][48][0018736C5681][0][*]/20
10.2.2.20 0 100 0 ?
*>i [2][10.2.2.20:2][0][48][0018736C56C2][0][*]/20
10.2.2.20 0 100 0 ?
*>i [2][10.2.2.20:2][0][48][0018736C56C2][32][192.168.2.89]/24
10.2.2.20 0 100 0 ?
*>i [2][10.2.2.20:2][0][48][380E4D9B6A42][32][192.168.2.202]/24
10.2.2.20 0 100 0 ?
*>i [2][10.2.2.20:2][0][48][380E4D9B6A42][128][2001:DB8:202::202]/36
10.2.2.20 0 100 0 ?
Route Distinguisher: 103:2 (default for vrf green)
*> [5][103:2][0][24][10.1.13.0]/17
0.0.0.0 0 32768 ?
*> [5][103:2][0][24][192.168.1.0]/17
0.0.0.0 0 32768 ?
*> [5][103:2][0][24][192.168.2.0]/17
0.0.0.0 0 32768 ?
*> [5][103:2][0][64][2001:DB8:201::]/29
::: 0 32768 ?
*> [5][103:2][0][64][2001:DB8:202::]/29
::: 0 32768 ?
Route Distinguisher: 104:2
*>i [5][104:2][0][24][10.1.14.0]/17
10.2.2.20 0 100 0 ?
*>i [5][104:2][0][24][192.168.1.0]/17
10.2.2.20 0 100 0 ?
*>i [5][104:2][0][24][192.168.2.0]/17
10.2.2.20 0 100 0 ?
*>i [5][104:2][0][64][2001:DB8:201::]/29
10.2.2.20 0 100 0 ?
*>i [5][104:2][0][64][2001:DB8:202::]/29
10.2.2.20 0 100 0 ?

```

**VTEP 2**

The following example shows the output for the **show bgp l2vpn evpn all** command on VTEP 2:

```

VTEP2# show bgp l2vpn evpn all
BGP table version is 584, local router ID is 10.2.2.20
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,

```

```

        x best-external, a additional-path, c RIB-compressed,
        t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 10.1.1.10:1
*>i [2][10.1.1.10:1][0][48][0059DC50AE01][0][*]/20
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][0059DC50AE4C][0][*]/20
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][0059DC50AE4C][32][192.168.1.81]/24
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][0059DC50AE4C][128][2001:DB8:201::81]/36
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][0059DC50AE4C][128][FE80::259:DCFF:FE50:AE4C]/36
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][A0F84910BCCC][32][192.168.1.201]/24
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:1][0][48][A0F84910BCCC][128][2001:DB8:201::201]/36
    10.1.1.10          0 100 0 ?
Route Distinguisher: 10.1.1.10:2
*>i [2][10.1.1.10:2][0][48][0059DC50AE01][0][*]/20
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:2][0][48][0059DC50AE42][0][*]/20
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:2][0][48][0059DC50AE42][32][192.168.2.81]/24
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:2][0][48][A0F84910BCC2][32][192.168.2.202]/24
    10.1.1.10          0 100 0 ?
*>i [2][10.1.1.10:2][0][48][A0F84910BCC2][128][2001:DB8:202::202]/36
    10.1.1.10          0 100 0 ?
Route Distinguisher: 10.2.2.20:1
*> [2][10.2.2.20:1][0][48][0018736C56C3][0][*]/20
    ::                  32768 ?
*> [2][10.2.2.20:1][0][48][0018736C56C3][32][192.168.1.89]/24
    ::                  32768 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE01][0][*]/20
    10.1.1.10          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE4C][0][*]/20
    10.1.1.10          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE4C][32][192.168.1.81]/24
    10.1.1.10          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE4C][128][2001:DB8:201::81]/36
    10.1.1.10          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][0059DC50AE4C][128][FE80::259:DCFF:FE50:AE4C]/36
    10.1.1.10          0 100 0 ?
*> [2][10.2.2.20:1][0][48][380E4D9B6A48][32][192.168.1.201]/24
    ::                  32768 ?
*> [2][10.2.2.20:1][0][48][380E4D9B6A48][128][2001:DB8:201::201]/36
    ::                  32768 ?
*>i [2][10.2.2.20:1][0][48][A0F84910BCCC][32][192.168.1.201]/24
    10.1.1.10          0 100 0 ?
*>i [2][10.2.2.20:1][0][48][A0F84910BCCC][128][2001:DB8:201::201]/36
    10.1.1.10          0 100 0 ?
Route Distinguisher: 10.2.2.20:2
*> [2][10.2.2.20:2][0][48][0018736C5681][0][*]/20
    ::                  32768 ?
*> [2][10.2.2.20:2][0][48][0018736C56C2][0][*]/20
    ::                  32768 ?
*> [2][10.2.2.20:2][0][48][0018736C56C2][32][192.168.2.89]/24
    ::                  32768 ?
*>i [2][10.2.2.20:2][0][48][0059DC50AE01][0][*]/20
    10.1.1.10          0 100 0 ?

```

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway

```

*>i [2][10.2.2.20:2][0][48][0059DC50AE42][0][*]/20
      10.1.1.10          0    100    0 ?
*>i [2][10.2.2.20:2][0][48][0059DC50AE42][32][192.168.2.81]/24
      10.1.1.10          0    100    0 ?
*> [2][10.2.2.20:2][0][48][380E4D9B6A42][32][192.168.2.202]/24
      ::                  32768 ?
*> [2][10.2.2.20:2][0][48][380E4D9B6A42][128][2001:DB8:202::202]/36
      ::                  32768 ?
*>i [2][10.2.2.20:2][0][48][A0F84910BCC2][32][192.168.2.202]/24
      10.1.1.10          0    100    0 ?
*>i [2][10.2.2.20:2][0][48][A0F84910BCC2][128][2001:DB8:202::202]/36
      10.1.1.10          0    100    0 ?
Route Distinguisher: 103:2
*>i [5][103:2][0][24][10.1.13.0]/17
      10.1.1.10          0    100    0 ?
*>i [5][103:2][0][24][192.168.1.0]/17
      10.1.1.10          0    100    0 ?
*>i [5][103:2][0][24][192.168.2.0]/17
      10.1.1.10          0    100    0 ?
*>i [5][103:2][0][64][2001:DB8:201::]/29
      10.1.1.10          0    100    0 ?
*>i [5][103:2][0][64][2001:DB8:202::]/29
      10.1.1.10          0    100    0 ?
Route Distinguisher: 104:2 (default for vrf green)
*> [5][104:2][0][24][10.1.14.0]/17
      0.0.0.0            0          32768 ?
*> [5][104:2][0][24][192.168.1.0]/17
      0.0.0.0            0          32768 ?
*> [5][104:2][0][24][192.168.2.0]/17
      0.0.0.0            0          32768 ?
*> [5][104:2][0][64][2001:DB8:201::]/29
      ::                  0          32768 ?
*> [5][104:2][0][64][2001:DB8:202::]/29
      ::                  0          32768 ?

```

**show ip route vrf green****VTEP 1**

The following example shows the output for the **show ip route vrf vrf-name** command on VTEP 1:

```

VTEP1# show ip route vrf green
Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C    10.1.13.0/24 is directly connected, Loopback13
L    10.1.13.13/32 is directly connected, Loopback13
B    10.1.14.0/24 [200/0] via 10.2.2.20, 01:30:02, Vlan200
     192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks

```

```
C      192.168.1.0/24 is directly connected, Vlan201
B      192.168.1.89/32 [200/0] via 10.2.2.20, 00:04:05, Vlan200
L      192.168.1.201/32 is directly connected, Vlan201
      192.168.2.0/24 is variably subnetted, 3 subnets, 2 masks
C      192.168.2.0/24 is directly connected, Vlan202
B      192.168.2.89/32 [200/0] via 10.2.2.20, 00:04:10, Vlan200
L      192.168.2.202/32 is directly connected, Vlan202
```

## VTEP 2

The following example shows the output for the **show ip route vrf vrf-name** command on VTEP 2:

```
VTEP2# show ip route vrf green
Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B      10.1.13.0/24 [200/0] via 10.1.1.10, 01:31:17, Vlan200
C      10.1.14.0/24 is directly connected, Loopback14
L      10.1.14.14/32 is directly connected, Loopback14
      192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks
C      192.168.1.0/24 is directly connected, Vlan201
B      192.168.1.81/32 [200/0] via 10.1.1.10, 01:39:53, Vlan200
L      192.168.1.201/32 is directly connected, Vlan201
      192.168.2.0/24 is variably subnetted, 3 subnets, 2 masks
C      192.168.2.0/24 is directly connected, Vlan202
B      192.168.2.81/32 [200/0] via 10.1.1.10, 01:39:30, Vlan200
L      192.168.2.202/32 is directly connected, Vlan202
```

## show platform software fed switch active matm mactable vlan

## VTEP 1

The following examples show the output for the **show platform software fed switch active matm mactable vlan vlan-id** command on VTEP 1:



**Note** The MAC address of the peer's core SVI interface must be present in the core VLAN.

```
VTEP1# show platform software fed switch active matm macTable vlan 200
VLAN  MAC                    Type  Seq#  EC_Bi  Flags  machandle          siHandle
      riHandle                diHandle                *a_time  *e_time  ports
-----
200   a0f8.4910.bce2             0x8002      0  19880      64  0x7f5d8503fd48    0x7f5d852b6d28
```

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway

```

0x0          0x5234          0          0  Vlan200
200  380e.4d9b.6a4a  0x1000001  0    0    64  0x7f5d855bfaa8  0x7f5d852aca68
    0x7f5d851c7078  0x0          0          0  RLOC 10.2.2.20 adj_id 126

```

Total Mac number of addresses:: 2

VTEP1# show platform software fed switch active matm macTable vlan 201

VLAN	MAC riHandle	Type diHandle	Seq#	EC_Bi	Flags	machandle *a_time *e_time ports	siHandle
201	00aa.00bb.00cc 0x0	0x8002 0x0	0	42949	64	0x7f5d85007b88 0 0 Vlan201	0x7f5d852b6d28
201	0059.dc50.ae01 0x0	0x1 0x7f5d8517eae8	9	0	0	0x7f5d852abaf8 300 9 GigabitEthernet1/0/1	0x7f5d85035248
201	a0f8.4910.bccc 0x0	0x8002 0x5234	0	19880	64	0x7f5d852ad618 0 9 Vlan201	0x7f5d852b6d28
201	0059.dc50.ae4c 0x0	0x1 0x7f5d8517eae8	16	0	0	0x7f5d855b3ff8 300 95 GigabitEthernet1/0/1	0x7f5d855a2858
201	380e.4d9b.6a48 0x0	0x8002 0x5234	0	0	64	0x7f5d84fbf948 0 95 Vlan201	0x7f5d852b6d28
201	0018.736c.56c3 0x7f5d855c6098	0x1000001 0x0	0	0	64	0x7f5d855c8268 0 95 RLOC 10.2.2.20 adj_id 36	0x7f5d852368b8

Total Mac number of addresses:: 6

VTEP1# show platform software fed switch active matm macTable vlan 202

VLAN	MAC riHandle	Type diHandle	Seq#	EC_Bi	Flags	machandle *a_time *e_time ports	siHandle
202	a0f8.4910.bcc2 0x0	0x8002 0x0	0	19880	64	0x7f5d8503d288 0 0 Vlan202	0x7f5d852b6d28
202	0059.dc50.ae01 0x0	0x1 0x7f5d8517eae8	10	0	0	0x7f5d852ac8b8 300 15 GigabitEthernet1/0/1	0x7f5d852ac668
202	0018.736c.5681 0x7f5d8518dea8	0x1000001 0x0	0	0	64	0x7f5d855ba7a8 0 15 RLOC 10.2.2.20 adj_id 125	0x7f5d855b0c58
202	0059.dc50.ae42 0x0	0x1 0x7f5d8517eae8	17	0	0	0x7f5d8518e848 300 225 GigabitEthernet1/0/1	0x7f5d855a5258
202	380e.4d9b.6a42 0x0	0x8002 0x5234	0	0	64	0x7f5d855a59a8 0 225 Vlan202	0x7f5d852b6d28
202	0018.736c.56c2 0x7f5d8518dea8	0x1000001 0x0	0	0	64	0x7f5d8523d2b8 0 225 RLOC 10.2.2.20 adj_id 125	0x7f5d855b0c58

Total Mac number of addresses:: 6



**VTEP 2**

The following examples show the output for the **show platform software fed switch active matm mactable vlan *vlan-id*** command on VTEP 2:



**Note** The MAC address of the peer's core SVI interface must be present in the core VLAN.

```
VTEP2# show platform software fed switch active matm macTable vlan 200
VLAN  MAC                               Type  Seq#  EC_Bi  Flags  machandle  siHandle
      riHandle                            diHandle  *a_time  *e_time  ports
-----
200   380e.4d9b.6a4a                       0x8002  0     128    64    0x7fa88557f3a8  0x7fa885574e38
      0x0                                  0x5174  0         0     0     Vlan200
200   a0f8.4910.bce2                       0x1000001  0     0      64    0x7fa8859a3d38  0x7fa885947ba8
      0x7fa88598bfb8                       0x0      0         0     0     RLOC 10.1.1.10 adj_id 155

Total Mac number of addresses:: 2
```

```
VTEP2# show platform software fed switch active matm macTable vlan 201
VLAN  MAC                               Type  Seq#  EC_Bi  Flags  machandle  siHandle
      riHandle                            diHandle  *a_time  *e_time  ports
-----
201   380e.4d9b.6a48                       0x8002  0  42949  64    0x7fa885970018  0x7fa885574e38
      0x0                                  0x5174  0         0     0     Vlan201
201   0059.dc50.ae01                       0x1000001  0     0      64    0x7fa8849e1be8  0x7fa88598da48
      0x7fa88598e1f8                       0x0      0         0     0     RLOC 10.1.1.10 adj_id 153
201   0059.dc50.ae4c                       0x1000001  0     0      64    0x7fa885993e68  0x7fa88598da48
      0x7fa88598e1f8                       0x0      0         0     0     RLOC 10.1.1.10 adj_id 153
201   a0f8.4910.bccc                       0x8002  0     0      64    0x7fa8859acc48  0x7fa885574e38
      0x0                                  0x5174  0         0     0     Vlan201
201   0018.736c.56c3                       0x1      68     0      0    0x7fa8859d3908  0x7fa88599e108
      0x0                                  0x7fa884f079d8  300      247  GigabitEthernet1/0/1

Total Mac number of addresses:: 5
```

```
VTEP2# show platform software fed switch active matm macTable vlan 202
VLAN  MAC                               Type  Seq#  EC_Bi  Flags  machandle  siHandle
      riHandle                            diHandle  *a_time  *e_time  ports
-----
202   380e.4d9b.6a42                       0x8002  0  19018  64    0x7fa885994cd8  0x7fa885574e38
      0x0                                  0x5174  0         0     0     Vlan202
202   0018.736c.5681                       0x1      9      0      0    0x7fa88599c4e8  0x7fa88599c218
      0x0                                  0x7fa884f079d8  300         7  GigabitEthernet1/0/1
202   0059.dc50.ae01                       0x1000001  0     0      64    0x7fa8859a3098  0x7fa8859a2dc8
      0x7fa88599ee48                       0x0      0         0     7     RLOC 10.1.1.10 adj_id 154
```

Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway

```

202    0059.dc50.ae42    0x1000001    0    0    64    0x7fa8849e6b78    0x7fa8859a2dc8
      0x7fa88599ee48    0x0          0          0          0          7    RLOC 10.1.1.10 adj_id 154

202    a0f8.4910.bcc2    0x8002      0    0    64    0x7fa88594ddb8    0x7fa885574e38
      0x0          0x5174      0          0          0          7    Vlan202

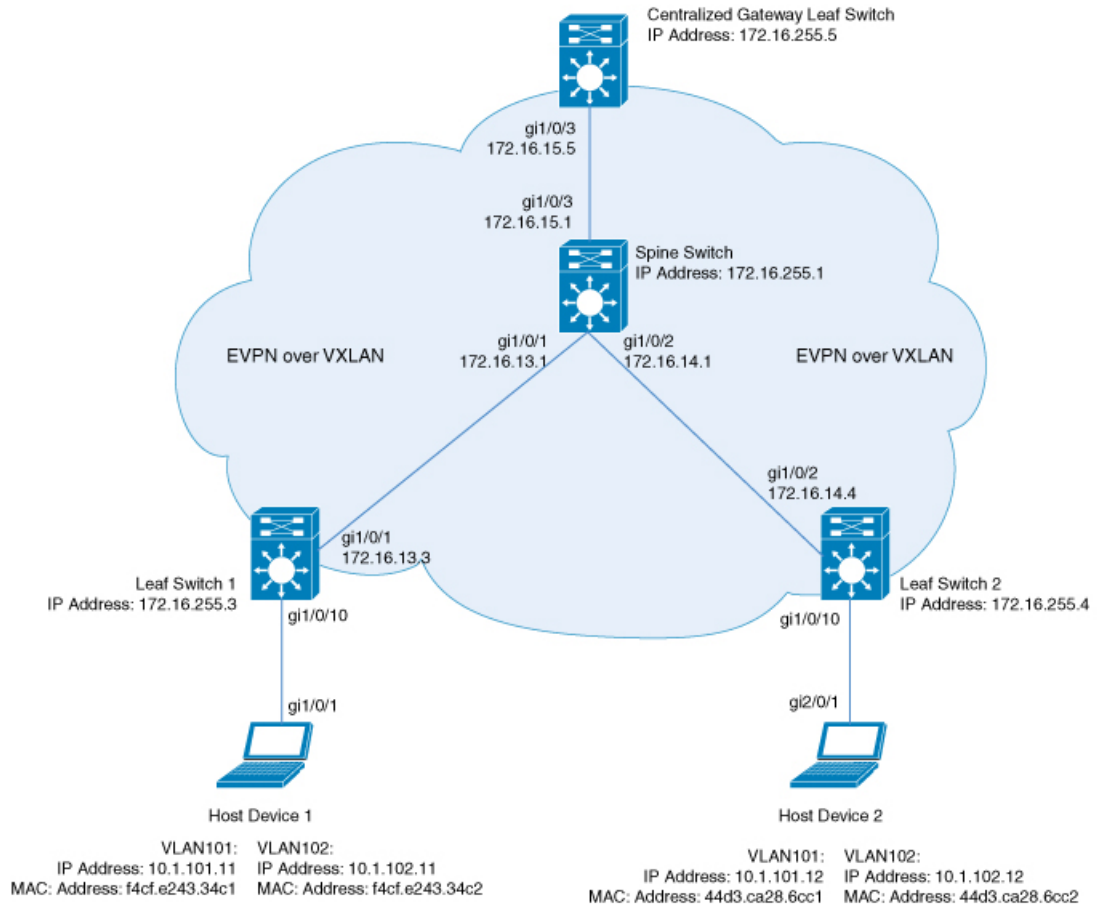
202    0018.736c.56c2    0x1        67    0    0    0x7fa8859d3488    0x7fa8859834f8
      0x0          0x7fa884f079d8    300      267    GigabitEthernet1/0/1

Total Mac number of addresses:: 6
    
```

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway

This section provides an example to show how EVPN VXLAN IRB is configured using centralized default gateway. The example shows how to configure and verify EVPN VXLAN IRB for the topology shown in the following image:

Figure 4: EVPN VXLAN Topology with Centralized Default Gateway



The topology shows an EVPN VXLAN network with a spine switch and three leaf switches (Leaf Switch 1, Leaf Switch 2, and Centralized Gateway Leaf Switch). The Centralized Gateway Leaf Switch provides the default gateway for all the VTEPs in the network. EVPN routes of route type 2 are advertised using the Default Gateway Extended Community attribute.



---

**Note** Each host device in the image is shown with two different VLANs only to illustrate the example.

---



---

**Note** The Centralized Gateway Leaf Switch can be configured either as a single switch or a stack of switches.

---

The following tables provide sample configurations for the devices in the topology above:

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway

Table 6: Configuring the Leaf Switches and Centralized Gateway Leaf Switch to Enable IRB using Centralized Default Gateway

Leaf Switch 1	Centralized Gateway Switch	Leaf Switch 2
<pre>Leaf_Switch1# show running-config hostname Leaf_Switch1 ! ip routing ! ip multicast-routing ! ipv6 unicast-routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type static ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 ! interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport mode trunk ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 10102 ingress-replication ! router ospf 1 router-id 172.16.255.3</pre>	<pre>CGW# show running-config hostname CGW ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! ip routing ! ip multicast-routing ! ipv6 unicast-routing ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type static ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 ! interface Loopback0 ip address 172.16.255.5 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.5 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 !</pre>	<pre>Leaf_Switch2# show running-config hostname Leaf_Switch2 ! ip routing ! ip multicast-routing ! ipv6 unicast-routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 ! interface Loopback0 ip address 172.16.255.4 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport mode trunk ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 10102 ingress-replication ! router ospf 1 router-id 172.16.255.4</pre>

Leaf Switch 1	Centralized Gateway Switch	Leaf Switch 2
<pre> ! router bgp 65001 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both exit-address-family ! ip pim rp-address 172.16.255.255 ! end ! Leaf_Switch1# </pre>	<pre> interface GigabitEthernet1/0/1 no switchport ip address 172.16.15.5 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport mode trunk ! interface Vlan101 description centralized gateway vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! interface Vlan102 description centralized gateway vrf forwarding green ip address 10.1.102.1 255.255.255.0 ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 10102 ingress-replication ! router ospf 1 router-id 172.16.255.5 ! router bgp 65001 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 ! address-family ipv4 redistribute connected redistribute static exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! ip pim rp-address 172.16.255.255 ! end </pre>	<pre> ! router bgp 65001 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both exit-address-family ! ip pim rp-address 172.16.255.255 ! end ! Leaf_Switch2# </pre>

Table 7: Configuring the Spine Switch to Enable IRB using Centralized Default Gateway

Spine Switch
<pre> Spine_Switch# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.14.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/3 no switchport ip address 172.16.15.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.1 ! router bgp 65001 bgp router-id 172.16.255.1 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.3 remote-as 65001 neighbor 172.16.255.3 update-source Loopback0 neighbor 172.16.255.4 remote-as 65001 neighbor 172.16.255.4 update-source Loopback0 neighbor 172.16.255.5 remote-as 65001 neighbor 172.16.255.5 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>

**Spine Switch**

```

!
address-family l2vpn evpn
neighbor 172.16.255.3 activate
neighbor 172.16.255.3 send-community both
neighbor 172.16.255.3 route-reflector-client
neighbor 172.16.255.4 activate
neighbor 172.16.255.4 send-community both
neighbor 172.16.255.4 route-reflector-client
neighbor 172.16.255.5 activate
neighbor 172.16.255.5 send-community both
neighbor 172.16.255.5 route-reflector-client
exit-address-family
!
ip pim rp-address 172.16.255.255
!
end
!
Spine_Switch#

```

The following examples provide sample outputs for **show** commands on Leaf Switch 1, Leaf Switch 2, and Centralized Gateway Leaf Switch to verify IRB for the topology configured above:

**Leaf Switch 1**

The following example shows the output for the **show l2vpn evpn default-gateway** command on Leaf Switch 1:

```

Leaf_Switch1# show l2vpn evpn default-gateway
Valid Default Gateway Address   EVI   VLAN   MAC Address   Source
-----
Y   10.1.101.1                   101   101   7c21.0dbd.2741 172.16.254.5
Y   10.1.102.1                   102   102   7c21.0dbd.274d 172.16.254.5

Leaf-01#sh l2vpn evpn default-gateway detail
Default Gateway Address: 10.1.101.1
EVPN Instance:          101
Vlan:                   101
MAC Address:            7c21.0dbd.2741
Ethernet Tag ID:       0
Source:                 V:10101 172.16.254.5

Default Gateway Address: 10.1.102.1
EVPN Instance:          102
Vlan:                   102
MAC Address:            7c21.0dbd.274d
Ethernet Tag ID:       0
Source:                 V:10102 172.16.254.5

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on Leaf Switch 1 for route type 2 and the IP address of the VLAN interface on Leaf Switch 1:

```

Leaf_Switch1# show bgp l2vpn evpn route-type 2 0 7c210dbd2741 10.1.101.1
BGP routing table entry for [2][172.16.254.3:101][0][48][7C210DBD2741][32][10.1.101.1]/24,
version 31009
Paths: (1 available, best #1, table evi_101)
    Not advertised to any peer

```

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway

```

Refresh Epoch 2
Local, imported path from [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24
(global)
  172.16.254.5 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
    Origin incomplete, metric 0, localpref 100, valid, internal, best
    EVPN ESI: 00000000000000000000, Label1 10101
    Extended Community: RT:65001:101 ENCAP:8 EVPN DEF GW:0:0
    Originator: 172.16.255.5, Cluster list: 172.16.255.1
    rx pathid: 0, tx pathid: 0x0
    net: 0x7F575D9E86B0, path: 0x7F575FBC5B10, pathext: 0x7F575DA095E8, exp_net:
0x7F575D9E8810
    flags: net: 0x0, path: 0x40000000000003, pathext: 0x81
    Updated on Jun 19 2020 12:43:11 UTC
BGP routing table entry for [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24,
version 31007
Paths: (1 available, best #1, table EVPN-BGP-Table)
Not advertised to any peer
Refresh Epoch 2
Local
  172.16.254.5 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
    Origin incomplete, metric 0, localpref 100, valid, internal, best
    EVPN ESI: 00000000000000000000, Label1 10101
    Extended Community: RT:65001:101 ENCAP:8 EVPN DEF GW:0:0
    Originator: 172.16.255.5, Cluster list: 172.16.255.1
    rx pathid: 0, tx pathid: 0x0
    net: 0x7F575D9E8810, path: 0x7F575FBC4958, pathext: 0x7F575DA09828
    flags: net: 0x0, path: 0x3, pathext: 0x81
    Updated on Jun 19 2020 12:43:11 UTC

```

The following example shows the output for the **show l2route evpn mac ip** command on Leaf Switch 1:

```

Leaf_Switch1# show l2route evpn mac ip
EVI      ETag  Prod   Mac Address          Host IP              Next Hop(s)
-----
101      0     BGP   44d3.ca28.6cc1      10.1.101.12         V:10101 172.16.254.4
101      0     BGP   7c21.0dbd.2741      10.1.101.1          V:10101 172.16.254.5
101      0     L2VPN f4cf.e243.34c1      10.1.101.11         Gi1/0/10:101
102      0     BGP   44d3.ca28.6cc2      10.1.102.12         V:10102 172.16.254.4
102      0     BGP   7c21.0dbd.274d      10.1.102.1          V:10102 172.16.254.5
102      0     L2VPN f4cf.e243.34c2      10.1.102.11         Gi1/0/10:102

```

The following example shows the output for the **show l2fib bridge-domain** command on Leaf Switch 1:

```

Leaf_Switch1# show l2fib bridge-domain 101 address unicast 7c21.0dbd.2741
MAC Address           : 7c21.0dbd.2741
Reference Count       : 1
Epoch                : 0
Producer              : BGP
Flags                 : None
Adjacency             : VXLAN_UC PL:2863(1) T:VXLAN_UC [MAC]10101:172.16.254.5
PD Adjacency         : VXLAN_UC PL:2863(1) T:VXLAN_UC [MAC]10101:172.16.254.5

```



## Leaf Switch 2

The following example shows the output for the **show l2vpn evpn default-gateway** command on Leaf Switch 2:

```
Leaf_Switch2# show l2vpn evpn default-gateway
Valid Default Gateway Address   EVI   VLAN  MAC Address   Source
-----
Y   10.1.101.1                   101   101   7c21.0dbd.2741 172.16.254.5
Y   10.1.102.1                   102   102   7c21.0dbd.274d 172.16.254.5

Leaf-02#sh l2vpn evpn default-gateway detail
Default Gateway Address: 10.1.101.1
EVPN Instance:          101
Vlan:                   101
MAC Address:            7c21.0dbd.2741
Ethernet Tag ID:       0
Source:                 V:10101 172.16.254.5

Default Gateway Address: 10.1.102.1
EVPN Instance:          102
Vlan:                   102
MAC Address:            7c21.0dbd.274d
Ethernet Tag ID:       0
Source:                 V:10102 172.16.254.5
```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on Leaf Switch 2 for route type 2 and the IP address of the VLAN interface on Leaf Switch 1:

```
Leaf_Switch2# show l2vpn evpn route-type 2 0 7c210dbd2741 10.1.101.1
BGP routing table entry for [2][172.16.254.4:101][0][48][7C210DBD2741][32][10.1.101.1]/24,
version 17202
Paths: (1 available, best #1, table evi_101)
  Not advertised to any peer
  Refresh Epoch 1
  Local, imported path from [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24
(global)
    172.16.254.5 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101
      Extended Community: RT:65001:101 ENCAP:8 EVPN DEF GW:0:0
      Originator: 172.16.255.5, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F84B88F13F0, path: 0x7F84BB57B4C8, pathext: 0x7F84B8F48C48, exp_net:
0x7F84B88F0210
      flags: net: 0x0, path: 0x40000000000003, pathext: 0x81
      Updated on Jun 19 2020 12:47:39 UTC
BGP routing table entry for [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24,
version 17200
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    172.16.254.5 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101
      Extended Community: RT:65001:101 ENCAP:8 EVPN DEF GW:0:0
      Originator: 172.16.255.5, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F84B88F0210, path: 0x7F84BB57AC58, pathext: 0x7F84B8F48E28
```

## Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway

```
flags: net: 0x0, path: 0x3, pathext: 0x81
Updated on Jun 19 2020 12:47:39 UTC
```

The following example shows the output for the **show l2route evpn mac ip** command on Leaf Switch 2:

```
Leaf_Switch2# show l2route evpn mac ip
-----
```

EVI	ETag	Prod	Mac Address	Host IP	Next Hop(s)
101	0	L2VPN	44d3.ca28.6cc1	10.1.101.12	Gi1/0/10:101
101	0	BGP	7c21.0dbd.2741	10.1.101.1	V:10101 172.16.254.5
101	0	BGP	f4cf.e243.34c1	10.1.101.11	V:10101 172.16.254.3
102	0	L2VPN	44d3.ca28.6cc2	10.1.102.12	Gi1/0/10:102
102	0	BGP	7c21.0dbd.274d	10.1.102.1	V:10102 172.16.254.5
102	0	BGP	f4cf.e243.34c2	10.1.102.11	V:10102 172.16.254.3

The following example shows the output for the **show l2fib bridge-domain** command on Leaf Switch 2:

```
Leaf_Switch2# show l2fib bridge-domain 101 address unicast 7c21.0dbd.2741

MAC Address           : 7c21.0dbd.2741
Reference Count       : 1
Epoch                : 0
Producer              : BGP
Flags                 : None
Adjacency             : VXLAN_UC PL:831(1) T:VXLAN_UC [MAC]10101:172.16.254.5
PD Adjacency         : VXLAN_UC PL:831(1) T:VXLAN_UC [MAC]10101:172.16.254.5
```

### Centralized Gateway Leaf Switch

The following example shows the output for the **show l2vpn evpn default-gateway** command on Centralized Gateway Leaf Switch:

```
CGW# show l2vpn evpn default-gateway
-----
```

Valid	Default Gateway Address	EVI	VLAN	MAC Address	Source
Y	10.1.101.1	101	101	7c21.0dbd.2741	V1101
Y	10.1.102.1	102	102	7c21.0dbd.274d	V1102

```
-----
CGW#sh l2vpn evpn default-gateway detail
Default Gateway Address: 10.1.101.1
EVPN Instance:          101
Vlan:                   101
MAC Address:            7c21.0dbd.2741
Ethernet Tag ID:       0
Source:                 V:10101 Vlan101

Default Gateway Address: 10.1.102.1
EVPN Instance:          102
Vlan:                   102
MAC Address:            7c21.0dbd.274d
Ethernet Tag ID:       0
Source:                 V:10102 Vlan102
```

The following example shows the output for the **show bgp l2vpn evpn evi evpn-instance route-type** command on Centralized Gateway Leaf Switch for route type 2 and the IP address of the VLAN interface on Leaf Switch 1:

```
CGW# show bgp l2vpn evpn evi 101 route-type 2 0 7c210dbd2741 10.1.101.1
BGP routing table entry for [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24,
version 39
Paths: (1 available, best #1, table evi_101)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (172.16.255.5)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      EVPN ESI: 00000000000000000000, Label1 10101
      Extended Community: RT:65001:101 ENCAP:8 EVPN DEF GW:0:0
      Local irb vxlan vtep:
        vrf: not found, l3-vni:0
        local router mac:0000.0000.0000
        core-irb interface: (not found)
        vtep-ip:172.16.254.5
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F3805208AF0, path: 0x7F380521B380, pathext: 0x7F3806746D98
      flags: net: 0x0, path: 0x4000028000003, pathext: 0x81
      Updated on Jun 19 2020 12:46:25 UTC
```

The following example shows the output for the **show l2route evpn mac ip** command on Centralized Gateway Leaf Switch:

```
CGW# show l2route evpn mac ip
```

EVI	ETag	Prod	Mac Address	Host IP	Next Hop(s)
101	0	BGP	44d3.ca28.6cc1	10.1.101.12	V:10101 172.16.254.4
101	0	L2VPN	7c21.0dbd.2741	10.1.101.1	V1101:0
101	0	BGP	f4cf.e243.34c1	10.1.101.11	V:10101 172.16.254.3
102	0	BGP	44d3.ca28.6cc2	10.1.102.12	V:10102 172.16.254.4
102	0	L2VPN	7c21.0dbd.274d	10.1.102.1	V1102:0
102	0	BGP	f4cf.e243.34c2	10.1.102.11	V:10102 172.16.254.3

The following example shows the output for the **show l2route evpn default-gateway detail** command on Centralized Gateway Leaf Switch:

```
CGW# show l2route evpn default-gateway detail
Ethernet Tag:          0          EVPN Instance:          101

Producer Name:        L2VPN
MAC Address:          7c21.0dbd.2741
Host IP:              10.1.101.1
Sequence Number:      0
ESI:                  0000.0000.0000.0000.0000
MAC Route Flags:      BInt()Dgl
Next Hop(s):          V1101:0

EVPN Instance:        102
Ethernet Tag:          0
Producer Name:        L2VPN
MAC Address:          7c21.0dbd.274d
Host IP:              10.1.102.1
```

```

Sequence Number:      0
ESI:                  0000.0000.0000.0000.0000
MAC Route Flags:      BInt()Dgl
Next Hop(s):          V1102:0

```

## Verifying EVPN VXLAN Integrated Routing and Bridging

The following sections provide information about how to verify EVPN VXLAN integrated routing and bridging:

### Verifying EVPN VXLAN Layer 2 Overlay Network

See [Verifying EVPN VXLAN Layer 2 Overlay Network, on page 29](#) for the list of commands.

### Verifying EVPN VXLAN Layer 3 Overlay Network

See [Verifying EVPN VXLAN Layer 3 Overlay Network, on page 51](#) for the list of commands.

### Verifying Distributed Anycast Gateway

The following table lists the **show** commands that are used to verify distributed anycast gateway:

**Table 8: Commands to Verify Distributed Anycast Gateway**

Command	Purpose
<b>show l2vpn evpn default-gateway</b>	Displays the default gateway database.
<b>show l2vpn l2route default-gateway</b>	Displays the list of sent or received default gateway routes.
<b>show mac address-table</b>	Displays the list of MAC addresses received in default gateway routes that are installed as static MAC addresses for an SVI interface.



## CHAPTER 5

# Configuring Spine Switches in a BGP EVPN VXLAN Fabric

---

- [Restrictions for Spine Switches in a BGP EVPN VXLAN Fabric, on page 93](#)
- [Information About Spine Switches in a BGP EVPN VXLAN Fabric, on page 93](#)
- [Configuration Examples for Spine Switches in a BGP EVPN VXLAN Network, on page 94](#)

## Restrictions for Spine Switches in a BGP EVPN VXLAN Fabric

- In an external BGP (eBGP) peering configuration between spine and leaf switches, by default, the spine switch changes the next-hop attribute of the received BGP routes to its own address and advertises these routes to the leaf switches.

The **neighbor ip-address next-hop-unchanged** command does not affect the prefixes that are exchanged in EVPN and Multicast Virtual Private Network (MVPN) address families.

Instead, run the **set ip next-hop unchanged** command in route-map configuration mode on the spine switches and configure next-hop unchanged for a route map. To use this route-map configuration to communicate with BGP neighbors, run the **neighbor ip-address route-map route-map-name out** command in L2VPN EVPN address family configuration mode on the spine switches. This route-map configuration is required on the spine switch to enable eBGP peering between spine and leaf switches.

## Information About Spine Switches in a BGP EVPN VXLAN Fabric

Spine switches in a BGP EVPN VXLAN fabric act as the connecting nodes between all the leaf switches or VTEPs. They form the backbone of the EVPN VXLAN network and forward traffic between the leaf switches. Each leaf switch is connected to each spine switch in the network. Spine switches enable redundancy within the network and provide multiple paths for VTEPs to forward traffic to each other.

Spine switches in an EVPN VXLAN network are part of the underlay network and transport the VXLAN-encapsulated packets. When deployed as border nodes, spine switches connect the network with an external network and allow movement of traffic. In a BGP EVPN VXLAN fabric, spine switches can also be deployed as route reflectors.

## Deployment Scenarios for Spine Switches and Leaf Switches in a BGP EVPN VXLAN Fabric

Spine switches and leaf switches in a BGP EVPN VXLAN fabric can be deployed in the following ways:

- Spine Switches and Leaf Switches in the Same Autonomous System
- Spine Switches in One Autonomous System and the Leaf Switches in a Different Autonomous System
- Spine Switches in One Autonomous System and Each Leaf Switch in a Different Autonomous System

### Spine Switches and Leaf Switches in the Same Autonomous System

In this scenario, all the devices in the EVPN VXLAN network are in the same autonomous system. The spine switches function as BGP route reflectors and anycast rendezvous points (RPs). Internal Border Gateway Protocol (iBGP) is used to establish peering between the spine switches, and between the spine and leaf switches.

See [Configuration Example for Spine Switches Using iBGP when the Spine Switches and Leaf Switches are in the Same Autonomous System, on page 95](#) for a sample topology and configuration.

### Spine Switches in One Autonomous System and the Leaf Switches in a Different Autonomous System

In this scenario, all the leaf switches are in a single autonomous system that is different from the autonomous system of the spine switches. The spine switches function as BGP route servers. iBGP is used to establish peering between the spine switches. eBGP is used to establish peering between the spine and leaf switches.

See [Configuration Example for Spine Switches Using eBGP when the Spine Switches are in One Autonomous System and the Leaf Switches are in a Different Autonomous System, on page 111](#) for a sample topology and configuration.

### Spine Switches in One Autonomous System and Each Leaf Switch in a Different Autonomous System

In this scenario, each leaf switch is in its own individual autonomous system that is different from the autonomous system of the spine switches. The spine switches function as BGP route servers. iBGP is used to establish peering between the spine switches. eBGP is used to establish peering between the spine and leaf switches.

See [Configuration Example for Spine Switches Using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System, on page 130](#) for a sample topology and configuration.

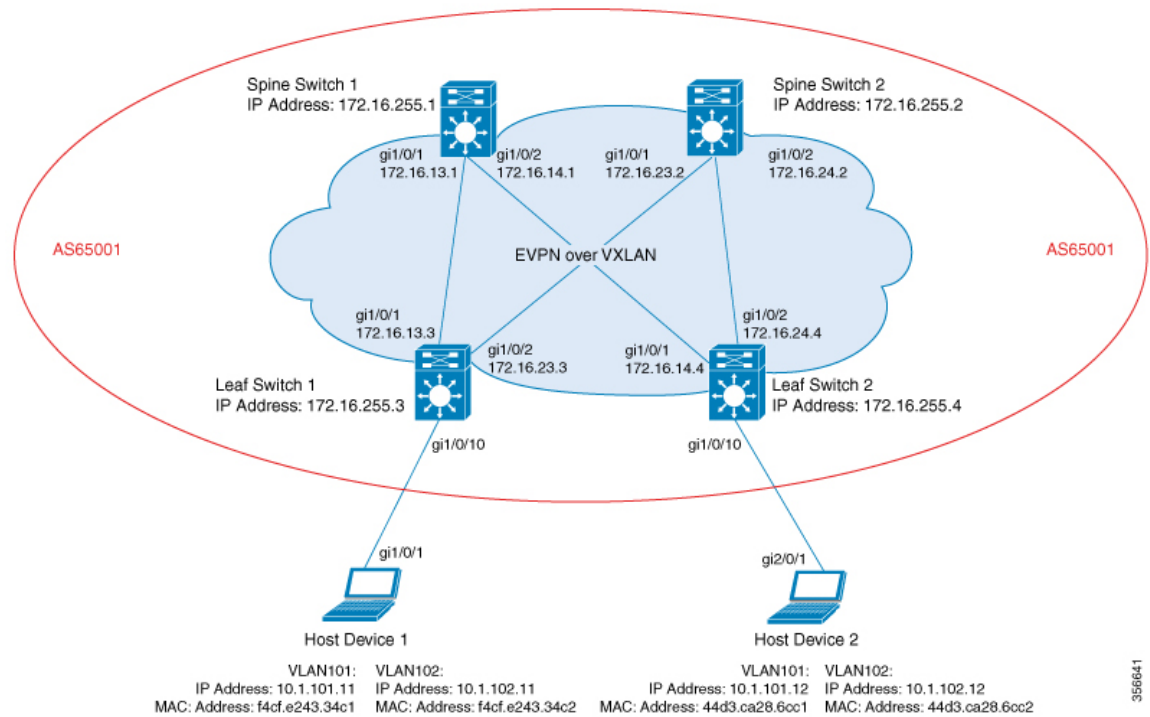
## Configuration Examples for Spine Switches in a BGP EVPN VXLAN Network

This section provides configuration examples for spine switches for the different deployments of spine and leaf switches in a BGP EVPN VXLAN fabric.

## Configuration Example for Spine Switches Using iBGP when the Spine Switches and Leaf Switches are in the Same Autonomous System

This section provides an example to show how spine switches are configured in a BGP EVPN VXLAN fabric using iBGP when the spine switches and leaf switches are in the same autonomous system. The example shows how to configure spine switches and verify the configuration for the topology shown below:

Figure 5: BGP EVPN VXLAN Fabric with the Spine Switches and Leaf Switches in the Same Autonomous System



The topology shows an EVPN VXLAN network with two leaf switches (VTEP 1 and VTEP 2) and two spine switches (spine switch 1 and spine switch 2). The entire BGP EVPN VXLAN fabric (which includes spine switch 1, spine switch 2, Leaf switch 1, and leaf switch 2) is in autonomous system AS65001. Anycast RP is configured on both the spine switches. Spine switch 1 and spine switch 2 are not route reflector clients to each other. Multicast Source Discovery Protocol (MSDP) is configured between spine switch 1 and spine switch 2 for source synchronization. Protocol Independent Multicast (PIM) is enabled on the interfaces that connect leaf switches and spine switches. Static RP is configured in the network and the underlay network uses multicast forwarding mechanism to forward BUM traffic.

The following tables provide sample configurations for the devices in the topology above.

Table 9: Configuring Spine Switch 1 and Spine Switch 2 using iBGP when the Spine Switches and the Leaf Switches are in the same Autonomous System

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.14.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.1 ! router bgp 65001 template peer-policy RR-PP route-reflector-client send-community both exit-peer-policy ! template peer-session RR-PS remote-as 65001 update-source Loopback0 exit-peer-session ! bgp router-id 172.16.255.1 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 neighbor 172.16.255.3 inherit peer-session RR-PS neighbor 172.16.255.4 inherit peer-session RR-PS ! address-family ipv4 exit-address-family ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! interface Loopback0 ip address 172.16.255.2 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.2 255.255.255.255 ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.23.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.2 ! router bgp 65001 template peer-policy RR-PP route-reflector-client send-community both exit-peer-policy ! template peer-session RR-PS remote-as 65001 update-source Loopback0 exit-peer-session ! bgp router-id 172.16.255.2 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.3 inherit peer-session RR-PS neighbor 172.16.255.4 inherit peer-session RR-PS ! address-family ipv4 exit-address-family ! </pre>



Spine Switch 1	Spine Switch 2
<pre> address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.3 activate neighbor 172.16.255.3 send-community extended neighbor 172.16.255.3 inherit peer-policy RR-PP neighbor 172.16.255.4 activate neighbor 172.16.255.4 send-community extended neighbor 172.16.255.4 inherit peer-policy RR-PP exit-address-family ! ip pim rp-address 172.16.255.255 ip msdp peer 172.16.254.2 connect-source Loopback1 remote-as 65001 ip msdp cache-sa-state ! end  Spine-01#                     </pre>	<pre> address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.3 activate neighbor 172.16.255.3 send-community extended neighbor 172.16.255.3 inherit peer-policy RR-PP neighbor 172.16.255.4 activate neighbor 172.16.255.4 send-community extended neighbor 172.16.255.4 inherit peer-policy RR-PP exit-address-family ! ip pim rp-address 172.16.255.255 ip msdp peer 172.16.254.1 connect-source Loopback1 remote-as 65001 ip msdp cache-sa-state ! end  Spine-02#                     </pre>

Table 10: Configuring Leaf Switch 1 and Leaf Switch 2 using iBGP when the Spine Switches and the Leaf Switches are in the same Autonomous System

Leaf Switch 1	Leaf Switch 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type static ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.4 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! </pre>

Leaf Switch 1	Leaf Switch 2
<pre> interface GigabitEthernet1/0/2 no switchport ip address 172.16.23.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport mode trunk ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ipv6 enable no autostate ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 10102 ingress-replication member vni 50901 vrf green ! router ospf 1 router-id 172.16.255.3 ! router bgp 65001 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! </pre>	<pre> interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport mode trunk ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ipv6 enable no autostate ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 50901 vrf green member vni 10102 ingress-replication ! router ospf 1 router-id 172.16.255.4 ! router bgp 65001 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! </pre>

Leaf Switch 1	Leaf Switch 2
<pre> address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 172.16.255.255 ! end Leaf-01# </pre>	<pre> address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 172.16.255.255 ! end Leaf-02# </pre>

The following examples provide sample outputs for **show** commands on the devices in the topology configured in the preceding tables:

### Spine Switch 1

The following example shows the output for the **show ip ospf neighbor** command on spine switch 1:

```
Spine-01# show ip ospf neighbor
```

Neighbor ID	Pri	State		Dead Time	Address	Interface
172.16.255.4	0	FULL/	-	00:00:39	172.16.14.4	GigabitEthernet1/0/2
172.16.255.3	0	FULL/	-	00:00:30	172.16.13.3	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on spine switch 1:

```
Spine-01# show bgp l2vpn evpn summary
```

```

BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 46, main routing table version 46
18 network entries using 6192 bytes of memory
38 path entries using 7904 bytes of memory
14/13 BGP path/bestpath attribute entries using 4032 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
12 BGP extended community entries using 640 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 18848 total bytes of memory
BGP activity 27/9 prefixes, 49/11 paths, scan interval 60 secs
18 networks peaked at 17:16:59 May 24 2020 UTC (22:49:24.588 ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.2	4	65001	1318	1314	46	0	0	19:39:19	18
172.16.255.3	4	65001	1517	1536	46	0	0	22:49:32	9
172.16.255.4	4	65001	1297	1310	46	0	0	19:23:05	11

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 1 for route type 2 and the IP address of host device 1:

```
Spine-01# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
```

```

BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 4

```

```
Paths: (2 available, best #2, table EVPN-BGP-Table)
```

```
Advertised to update-groups:
```

```

1          2
Refresh Epoch 1
Local
172.16.254.3 (metric 2) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
    Router MAC:10B3.D56A.8FC8
  Originator: 172.16.255.3, Cluster list: 172.16.255.2
  rx pathid: 0, tx pathid: 0
  net: 0x7F54CCA547D0, path: 0x7F54CCA63D70, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on May 24 2020 20:42:55 UTC
Refresh Epoch 2
Local, (Received from a RR-client)
172.16.254.3 (metric 2) (via default) from 172.16.255.3 (172.16.255.3)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
    Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F54CCA547D0, path: 0x7F54CCA64AF0, pathext: 0x7F54CA789BA8
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on May 24 2020 17:16:50 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 1 for route type 2 and the IP address of host device 2:

```

Spine-01# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 42
Paths: (2 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    1          2
Refresh Epoch 2
Local, (Received from a RR-client)
172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
    Router MAC:7C21.0DBD.9548
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F54CCA53E30, path: 0x7F54CCA63428, pathext: 0x7F54CA7898A8
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on May 24 2020 20:43:18 UTC
Refresh Epoch 1
Local
172.16.254.4 (metric 2) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
    Router MAC:7C21.0DBD.9548
  Originator: 172.16.255.4, Cluster list: 172.16.255.2
  rx pathid: 0, tx pathid: 0
  net: 0x7F54CCA53E30, path: 0x7F54CCA64280, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on May 24 2020 20:28:04 UTC

```

The following example shows the output for the **show ip pim neighbor** command on spine switch 1:

```
Spine-01# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface          Uptime/Expires    Ver   DR
Address
172.16.13.3       GigabitEthernet1/0/1  1d22h/00:01:41    v2    1 / DR S P G
172.16.14.4       GigabitEthernet1/0/2  4w5d/00:01:24    v2    1 / DR S P G
```

The following example shows the output for the **show ip pim rp map** command on spine switch 1:

```
Spine-01# show ip pim rp map
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)
```

The following example shows the output for the **show ip rpf** command on spine switch 1:

```
Spine-01# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
  RPF interface: Loopback2
  RPF neighbor: ? (172.16.255.255) - directly connected
  RPF route/mask: 172.16.255.255/32
  RPF type: multicast (connected)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base
```

The following example shows the output for the **show ip msdp summary** command on spine switch 1:

```
Spine-01# show ip msdp summary
MSDP Peer Status Summary
Peer Address      AS      State    Uptime/  Reset SA   Peer Name
                  Downtime Count Count
172.16.254.2     65001  Up       22:37:35 0       2       ?
```

The following example shows the output for the **show ip msdp sa-cache** command on spine switch 1:

```
Spine-01# show ip msdp sa-cache
MSDP Source-Active Cache - 2 entries
(172.16.254.3, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:00:29/00:05:30, Peer 172.16.254.2
(172.16.254.4, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:00:17/00:05:43, Peer 172.16.254.2
```

The following example shows the output for the **show ip mroute** command on spine switch 1:

```
Spine-01# show ip mroute 225.0.0.10
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
```

```

G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 00:01:04/stopped, RP 172.16.255.255, flags: SP
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list: Null

(172.16.254.4, 225.0.0.101), 00:00:51/00:02:08, flags: PA
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
Outgoing interface list: Null

(172.16.254.3, 225.0.0.101), 00:01:04/00:01:55, flags: PA
Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.13.3
Outgoing interface list: Null

```

## Spine Switch 2

The following example shows the output for the **show ip ospf neighbor** command on spine switch 2:

```
Spine-02# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.255.4	0	FULL/ -	00:00:39	172.16.24.4	GigabitEthernet1/0/2
172.16.255.3	0	FULL/ -	00:00:35	172.16.23.3	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on spine switch 2:

```
Spine-02# show bgp l2vpn evpn summary
```

```

BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 28, main routing table version 28
18 network entries using 6192 bytes of memory
38 path entries using 7904 bytes of memory
14/13 BGP path/bestpath attribute entries using 4032 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
12 BGP extended community entries using 640 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 18848 total bytes of memory
BGP activity 36/18 prefixes, 58/20 paths, scan interval 60 secs
18 networks peaked at 16:03:20 May 24 2020 UTC (1d00h ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	1327	1331	28	0	0	19:51:26	18
172.16.255.3	4	65001	1307	1322	28	0	0	19:35:35	9
172.16.255.4	4	65001	1316	1334	28	0	0	19:51:36	11

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 2 for route type 2 and the IP address of host device 1:

```

Spine-02# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 24
Paths: (2 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    2          3
  Refresh Epoch 2
  Local, (Received from a RR-client)
    172.16.254.3 (metric 2) (via default) from 172.16.255.3 (172.16.255.3)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
        Router MAC:10B3.D56A.8FC8
      rx pathid: 0, tx pathid: 0x0
      net: 0x7FEFE69D6638, path: 0x7FEFE45FED18, pathext: 0x7FEFE6645CC0
      flags: net: 0x0, path: 0x3, pathext: 0x81
      Updated on May 24 2020 20:43:24 UTC
  Refresh Epoch 1
  Local
    172.16.254.3 (metric 2) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
        Router MAC:10B3.D56A.8FC8
      Originator: 172.16.255.3, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0
      net: 0x7FEFE69D6638, path: 0x7FEFE45FF738, pathext: 0x0
      flags: net: 0x0, path: 0x3, pathext: 0x0
      Updated on May 24 2020 20:27:33 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 2 for route type 2 and the IP address of host device 2:

```

Spine-02# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 10
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    2          3
  Refresh Epoch 1
  Local
    172.16.254.4 (metric 2) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
        Router MAC:7C21.0DBD.9548
      Originator: 172.16.255.4, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0
      net: 0x7FEFE69D64D8, path: 0x7FEFE45FE730, pathext: 0x0
      flags: net: 0x0, path: 0x3, pathext: 0x0
      Updated on May 24 2020 20:43:46 UTC
  Refresh Epoch 1
  Local, (Received from a RR-client)
    172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
        Router MAC:7C21.0DBD.9548
      rx pathid: 0, tx pathid: 0x0
      net: 0x7FEFE69D64D8, path: 0x7FEFE45FF660, pathext: 0x7FEFE6645B40
      flags: net: 0x0, path: 0x3, pathext: 0x81
      Updated on May 24 2020 20:27:22 UTC

```



The following example shows the output for the **show ip pim neighbor** command on spine switch 2:

```
Spine-02# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor      Interface      Uptime/Expires    Ver   DR
Address
172.16.23.3   GigabitEthernet1/0/1    6w3d/00:01:21    v2    1 / DR S P G
172.16.24.4   GigabitEthernet1/0/2    1d22h/00:01:18    v2    1 / DR S P G
```

The following example shows the output for the **show ip pim rp map** command on spine switch 2:

```
Spine-02# show ip pim rp map
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)
```

The following example shows the output for the **show ip rpf** command on spine switch 2:

```
Spine-02# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
  RPF interface: Loopback2
  RPF neighbor: ? (172.16.255.255) - directly connected
  RPF route/mask: 172.16.255.255/32
  RPF type: multicast (connected)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base
```

The following example shows the output for the **show ip msdp summary** command on spine switch 2:

```
Spine-02# show ip msdp summary
MSDP Peer Status Summary
Peer Address      AS      State    Uptime/  Reset SA    Peer Name
                  Downtime Count Count
172.16.254.1     65001  Up       22:41:13 3        2        ?
```

The following example shows the output for the **show ip msdp sa-cache** command on spine switch 2:

```
Spine-02# show ip msdp sa-cache
MSDP Source-Active Cache - 2 entries
(172.16.254.3, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:04:09/00:05:57, Peer 172.16.254.1
(172.16.254.4, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:03:56/00:05:57, Peer 172.16.254.1
```

The following example shows the output for the **show ip mroute** command on spine switch 2:

```
Spine-02# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
```

```

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 5w6d/00:03:16, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d22h/00:03:10
    GigabitEthernet1/0/1, Forward/Sparse, 5w6d/00:02:55

(172.16.254.4, 225.0.0.101), 00:00:13/00:02:46, flags: TA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 00:00:13/00:03:16

(172.16.254.3, 225.0.0.101), 00:00:23/00:02:36, flags: A
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:00:23/00:03:10

```

## Leaf Switch 1

The following example shows the output for the **show ip ospf neighbor** command on leaf switch 1:

```
Leaf-01# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.255.2	0	FULL/ -	00:00:34	172.16.23.2	GigabitEthernet1/0/2
172.16.255.1	0	FULL/ -	00:00:30	172.16.13.1	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on leaf switch 1:

```
Leaf-01# show bgp l2vpn evpn summary
```

```

BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 11429, main routing table version 11429
27 network entries using 9288 bytes of memory
36 path entries using 7488 bytes of memory
15/15 BGP path/bestpath attribute entries using 4320 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
12 BGP extended community entries using 624 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 21800 total bytes of memory
BGP activity 398/365 prefixes, 4243/4201 paths, scan interval 60 secs
89 networks peaked at 20:32:14 Apr 21 2020 UTC (4w5d ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	261	242	11429	0	0	03:28:13	9
172.16.255.2	4	65001	31	16	11429	0	0	00:02:08	9

The following example shows the output for the **show bgp l2vpn evpn route-type 2** command on leaf switch 1 for route type 2 and the IP address of host device 2:

```
Leaf-01# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 11423
Paths: (1 available, best #1, table evi_101)
  Not advertised to any peer
  Refresh Epoch 1
  Local, imported path from [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24
(global)
    172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
      Originator: 172.16.255.4, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F575DB9FAB0, path: 0x7F575FD77698, pathext: 0x7F575DBD5B48, exp_net:
0x7F575DBA3B50
      flags: net: 0x0, path: 0x40000000000003, pathext: 0x81
      Updated on May 24 2020 20:40:59 UTC
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 11414
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    172.16.254.4 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
      Origin incomplete, metric 0, localpref 100, valid, internal
      EVPN ESI: 000000000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
      Originator: 172.16.255.4, Cluster list: 172.16.255.2
      rx pathid: 0, tx pathid: 0
      net: 0x7F575DBA3B50, path: 0x7F575FD77E30, pathext: 0x0
      flags: net: 0x0, path: 0x3, pathext: 0x0
      Updated on May 24 2020 20:40:37 UTC
  Refresh Epoch 1
  Local
    172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 000000000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
      Originator: 172.16.255.4, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F575DBA3B50, path: 0x7F575FD769F0, pathext: 0x7F575DBD5D88
      flags: net: 0x0, path: 0x3, pathext: 0x81
      Updated on May 24 2020 20:40:59 UTC
```

The following example shows the output for the **show ip pim neighbor** command on leaf switch 1:

```
Leaf-01# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface                Uptime/Expires    Ver  DR
Address
172.16.13.1       GigabitEthernet1/0/1    1d03h/00:01:21    v2   1 / S P G
```

```
172.16.23.2      GigabitEthernet1/0/2      6w2d/00:01:25      v2      1 / S P G
```

The following example shows the output for the **show ip pim rp mapping** command on leaf switch 1:

```
Leaf-01# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)
```

The following example shows the output for the **show ip ro** command on leaf switch 1:

```
Leaf-01# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 172.16.13.1 on GigabitEthernet1/0/1, 1d03h ago
  Routing Descriptor Blocks:
  * 172.16.23.2, from 172.16.255.2, 4w5d ago, via GigabitEthernet1/0/2
    Route metric is 2, traffic share count is 1
    172.16.13.1, from 172.16.255.1, 1d03h ago, via GigabitEthernet1/0/1
    Route metric is 2, traffic share count is 1
```

The following example shows the output for the **show ip rpf** command on leaf switch 1:

```
Leaf-01# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
  RPF interface: GigabitEthernet1/0/2
  RPF neighbor: ? (172.16.23.2)
  RPF route/mask: 172.16.255.255/32
  RPF type: unicast (ospf 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

The following example shows the output for the **show ip mroute** command on leaf switch 1:

```
Leaf-01# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 6w2d/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
```

```

Outgoing interface list:
  Tunnel0, Forward/Sparse-Dense, 6w2d/00:01:57

(172.16.254.4, 225.0.0.101), 00:00:49/00:02:10, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:00:49/00:02:10

(172.16.254.3, 225.0.0.101), 00:01:01/00:01:58, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:01:01/00:03:27

```

## Leaf Switch 2

The following example shows the output for the **show ip ospf neighbor** command on leaf switch 2:

```
Leaf-02# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.255.2	0	FULL/ -	00:00:34	172.16.24.2	GigabitEthernet1/0/2
172.16.255.1	0	FULL/ -	00:00:35	172.16.14.1	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on leaf switch 2:

```
Leaf-02# show bgp l2vpn evpn summary
```

```

BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 168, main routing table version 168
25 network entries using 8600 bytes of memory
36 path entries using 7488 bytes of memory
16/15 BGP path/bestpath attribute entries using 4608 bytes of memory
2 BGP rinfo entries using 80 bytes of memory
13 BGP extended community entries using 664 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 21440 total bytes of memory
BGP activity 70/39 prefixes, 168/124 paths, scan interval 60 secs
31 networks peaked at 15:56:08 May 24 2020 UTC (05:05:36.264 ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	45	31	168	0	0	00:16:18	9
172.16.255.2	4	65001	54	48	168	0	0	00:32:42	9

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 2 for route type 2 and the IP address of host device 1:

```

Leaf-02# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
  version 163
Paths: (2 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    172.16.254.3 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8

```

```

Router MAC:10B3.D56A.8FC8
Originator: 172.16.255.3, Cluster list: 172.16.255.1
rx pathid: 0, tx pathid: 0x0
net: 0x7F84B9145020, path: 0x7F84BB3355F8, pathext: 0x7F84BB5B4318
flags: net: 0x0, path: 0x3, pathext: 0x81
Updated on May 24 2020 20:45:25 UTC
Refresh Epoch 1
Local
  172.16.254.3 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  Originator: 172.16.255.3, Cluster list: 172.16.255.2
  rx pathid: 0, tx pathid: 0
  net: 0x7F84B9145020, path: 0x7F84BB333948, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on May 24 2020 20:45:03 UTC
BGP routing table entry for [2][172.16.254.4:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 166
Paths: (1 available, best #1, table evi_101)
Not advertised to any peer
Refresh Epoch 2
Local, imported path from [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24
(global)
  172.16.254.3 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  Originator: 172.16.255.3, Cluster list: 172.16.255.1
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F84B9145700, path: 0x7F84BB334008, pathext: 0x7F84BB5B3A18, exp_net:
0x7F84B9145020
  flags: net: 0x0, path: 0x40000000000003, pathext: 0x81
  Updated on May 24 2020 20:45:25 UTC

```

The following example shows the output for the **show ip pim neighbor** command on leaf switch 2:

```

Leaf-02# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor      Interface                Uptime/Expires    Ver   DR
Address
172.16.14.1   GigabitEthernet1/0/1    4w5d/00:01:26    v2    1 / S P G
172.16.24.2   GigabitEthernet1/0/2    1d03h/00:01:20    v2    1 / S P G

```

The following example shows the output for the **show ip pim rp map** command on leaf switch 2:

```

Leaf-02# show ip pim rp map
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)

```

The following example shows the output for the **show ip ro** command on leaf switch 2:

```
Leaf-02# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 172.16.14.1 on GigabitEthernet1/0/1, 05:12:11 ago
Routing Descriptor Blocks:
  * 172.16.24.2, from 172.16.255.2, 05:12:11 ago, via GigabitEthernet1/0/2
    Route metric is 2, traffic share count is 1
  172.16.14.1, from 172.16.255.1, 05:12:11 ago, via GigabitEthernet1/0/1
    Route metric is 2, traffic share count is 1
```

The following example shows the output for the **show ip mroute** command on leaf switch 2:

```
Leaf-02# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 3d07h/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 3d07h/00:00:38

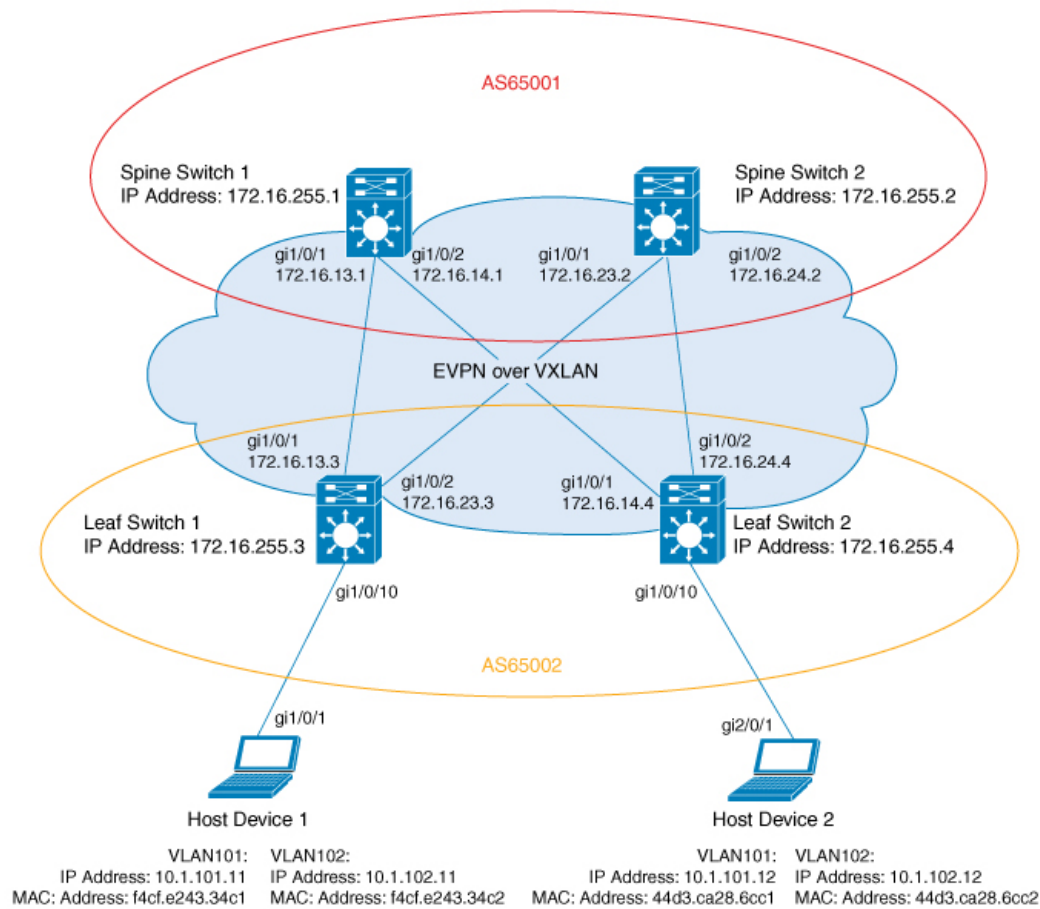
(172.16.254.4, 225.0.0.101), 00:00:09/00:02:50, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:00:09/00:03:20

(172.16.254.3, 225.0.0.101), 00:00:28/00:02:31, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:00:28/00:02:31
```

## Configuration Example for Spine Switches Using eBGP when the Spine Switches are in One Autonomous System and the Leaf Switches are in a Different Autonomous System

This section provides an example to show how spine switches are configured in a BGP EVPN VXLAN fabric using eBGP when the spine switches are in one autonomous system and the leaf switches are in a different autonomous system. The example shows how to configure spine switches and verify the configuration for the topology shown below:

Figure 6: BGP EVPN VXLAN Fabric with the Spine Switches in One Autonomous System and the Leaf Switches in a Different Autonomous System



The topology shows an EVPN VXLAN network with two leaf switches (leaf switch 1 and leaf switch 2) and two spine switches (spine switch 1 and spine switch 2). Spine switch 1 and spine switch 2 are in autonomous system AS65001. Leaf switch 1 and leaf switch 2 are in autonomous system AS65002. Spine switch 1 and spine switch 2 are BGP route servers and are not route reflector clients to each other. Multicast Source Discovery Protocol (MSDP) is configured between spine switch 1 and spine switch 2 for source synchronization. Protocol Independent Multicast (PIM) is enabled on the interfaces that connect leaf switches and spine switches. Static RP is configured in the network and the underlay network uses multicast forwarding mechanism to forward BUM traffic.



**Note** You must run the **neighbor ip-address allows-in** command in the L2VPN EVPN address family configuration mode on the leaf switches to allow processing of BGP updates that have a different autonomous system number.





---

**Note** You must manually run the **no bgp default route-target filter** command in router configuration mode on the spine switches.

---



---

**Note** You must configure eBGP multihop on the leaf and spine switches for the fabric to function.

---

The following tables provide sample configurations for the devices in the topology above.

Table 11: Configuring Spine Switch 1 and Spine Switch 2 using eBGP when the Spine Switches are in one Autonomous System and the Leaf Switches are in a Different Autonomous System

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.14.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.1 ! router bgp 65001 bgp router-id 172.16.255.1 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 neighbor 172.16.255.3 remote-as 65002 neighbor 172.16.255.3 ebgp-multihop 255 neighbor 172.16.255.3 update-source Loopback0 neighbor 172.16.255.4 remote-as 65002 neighbor 172.16.255.4 ebgp-multihop 255 neighbor 172.16.255.4 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! interface Loopback0 ip address 172.16.255.2 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.2 255.255.255.255 ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.23.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.2 ! router bgp 65001 bgp router-id 172.16.255.2 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.3 remote-as 65002 neighbor 172.16.255.3 ebgp-multihop 255 neighbor 172.16.255.3 update-source Loopback0 neighbor 172.16.255.4 remote-as 65002 neighbor 172.16.255.4 ebgp-multihop 255 neighbor 172.16.255.4 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>

Spine Switch 1	Spine Switch 2
<pre> address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-map BGP-NHU out neighbor 172.16.255.3 activate neighbor 172.16.255.3 send-community extended neighbor 172.16.255.3 route-map BGP-NHU out neighbor 172.16.255.4 activate neighbor 172.16.255.4 send-community both neighbor 172.16.255.4 route-map BGP-NHU out exit-address-family ! ip pim rp-address 172.16.255.255 ip msdp peer 172.16.254.2 connect-source Loopback1 remote-as 65001 ip msdp cache-sa-state ! route-map BGP-NHU permit 10 set ip next-hop unchanged ! end  Spine-01#                     </pre>	<pre> address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-map BGP-NHU out neighbor 172.16.255.3 activate neighbor 172.16.255.3 send-community both neighbor 172.16.255.3 route-map BGP-NHU out neighbor 172.16.255.4 activate neighbor 172.16.255.4 send-community both neighbor 172.16.255.4 route-map BGP-NHU out exit-address-family ! ip pim rp-address 172.16.255.255 ip msdp peer 172.16.254.1 connect-source Loopback1 remote-as 65001 ip msdp cache-sa-state ! route-map BGP-NHU permit 10 set ip next-hop unchanged ! end  Spine-02#                     </pre>

**Table 12: Configuring Leaf Switch 1 and Leaf Switch 2 using eBGP when the Spine Switches are in one Autonomous System and the Leaf Switches are in a Different Autonomous System**

Leaf Switch 1	Leaf Switch 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type static ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.4 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! </pre>

Leaf Switch 1	Leaf Switch 2
<pre> interface GigabitEthernet1/0/2 no switchport ip address 172.16.23.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport mode trunk ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ipv6 enable no autostate ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 10102 ingress-replication member vni 50901 vrf green ! router ospf 1 router-id 172.16.255.3 ! router bgp 65002 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 ebgp-multihop 255 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 ebgp-multihop 255 neighbor 172.16.255.2 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 allowas-in neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 allowas-in exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family </pre>	<pre> interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport mode trunk ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ipv6 enable no autostate ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 50901 vrf green member vni 10102 ingress-replication ! router ospf 1 router-id 172.16.255.4 ! router bgp 65002 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 ebgp-multihop 255 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 ebgp-multihop 255 neighbor 172.16.255.2 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 allowas-in neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 allowas-in exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family </pre>

Leaf Switch 1	Leaf Switch 2
<pre>! address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 172.16.255.255 ! end Leaf-01#</pre>	<pre>! address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 172.16.255.255 ! end Leaf-02#</pre>

The following examples provide sample outputs for **show** commands on the devices in the topology configured in the preceding tables:

### Spine Switch 1

The following example shows the output for the **show ip ospf neighbor** command on spine switch 1:

```
Spine-01# show ip ospf neighbor
```

Neighbor ID	Pri	State		Dead Time	Address	Interface
172.16.255.4	0	FULL/	-	00:00:33	172.16.14.4	GigabitEthernet1/0/2
172.16.255.3	0	FULL/	-	00:00:34	172.16.13.3	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on spine switch 1:

```
Spine-01# show bgp l2vpn evpn summary
```

```
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 75, main routing table version 75
18 network entries using 6192 bytes of memory
38 path entries using 7904 bytes of memory
27/13 BGP path/bestpath attribute entries using 7776 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
12 BGP extended community entries using 640 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 22536 total bytes of memory
BGP activity 18/0 prefixes, 76/38 paths, scan interval 60 secs
18 networks peaked at 20:34:25 May 27 2020 UTC (5d18h ago)
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.2	4	65001	9196	9183	75	0	0	5d18h	18
172.16.255.3	4	65002	8446	8456	75	0	0	5d07h	9
172.16.255.4	4	65002	8446	8447	75	0	0	5d07h	11

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 1 for route type 2 and the IP address of host device 2:

```
Spine-01# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
```

```
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 72
```

```
Paths: (2 available, best #1, table EVPN-BGP-Table)
```

```
Advertised to update-groups:
```

```

4          5
Refresh Epoch 2
65002
172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
  Origin incomplete, metric 0, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
    Router MAC:7C21.0DBD.9548
    rx pathid: 0, tx pathid: 0x0
    net: 0x7F789AD67240, path: 0x7F789AD76820, pathext: 0x7F789AD88298
    flags: net: 0x0, path: 0x3, pathext: 0x81
    Updated on May 28 2020 07:29:30 UTC
Refresh Epoch 1
65002
172.16.254.4 (metric 2) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
    Router MAC:7C21.0DBD.9548
    rx pathid: 0, tx pathid: 0
    net: 0x7F789AD67240, path: 0x7F789AD76EE0, pathext: 0x0
    flags: net: 0x0, path: 0x3, pathext: 0x0
    Updated on May 28 2020 07:27:54 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 1 for route type 2 and the IP address of host device 2:

```

Spine-01# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
  version 40
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    4          5
Refresh Epoch 1
65002
172.16.254.3 (metric 2) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
    Router MAC:10B3.D56A.8FC8
    rx pathid: 0, tx pathid: 0
    net: 0x7F789AD67EA0, path: 0x7F789AD77678, pathext: 0x0
    flags: net: 0x0, path: 0x3, pathext: 0x0
    Updated on May 28 2020 07:29:03 UTC
Refresh Epoch 1
65002
172.16.254.3 (metric 2) (via default) from 172.16.255.3 (172.16.255.3)
  Origin incomplete, metric 0, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
    Router MAC:10B3.D56A.8FC8
    rx pathid: 0, tx pathid: 0x0
    net: 0x7F789AD67EA0, path: 0x7F789AD77FC0, pathext: 0x7F789AD88598
    flags: net: 0x0, path: 0x3, pathext: 0x81
    Updated on May 28 2020 07:27:47 UTC

```

The following example shows the output for the **show ip pim neighbor** command on spine switch 1:

```

Spine-01# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

```

```

P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
L - DR Load-balancing Capable
Neighbor      Interface      Uptime/Expires  Ver  DR
Address
172.16.13.3   GigabitEthernet1/0/1   5d19h/00:01:44  v2   1 / DR S P G
172.16.14.4   GigabitEthernet1/0/2   5d19h/00:01:36  v2   1 / DR S P G

```

The following example shows the output for the **show ip pim rp mapping** command on spine switch 1:

```

Spine-01# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
RP: 172.16.255.255 (?)

```

The following example shows the output for the **show ip ro** command on spine switch 1:

```

Spine-01# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Loopback2
    Route metric is 0, traffic share count is 1

```

The following example shows the output for the **show ip msdp summary** command on spine switch 1:

```

Spine-01# show ip msdp summary
MSDP Peer Status Summary
Peer Address      AS      State      Uptime/  Reset SA      Peer Name
                  Downtime Count Count
172.16.254.2      65001  Up         5d19h    0      2      ?

```

The following example shows the output for the **show ip msdp sa-cache** command on spine switch 1:

```

Spine-01# show ip msdp sa-cache
MSDP Source-Active Cache - 2 entries
(172.16.254.3, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:04:01/00:05:23, Peer 172.16.254.2
(172.16.254.4, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:03:39/00:05:26, Peer 172.16.254.2

```

The following example shows the output for the **show ip mroute** command on spine switch 1:

```

Spine-01# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,

```



```

G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 00:04:02/stopped, RP 172.16.255.255, flags: SP
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list: Null

(172.16.254.4, 225.0.0.101), 00:00:34/00:02:25, flags: PA
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
Outgoing interface list: Null

(172.16.254.3, 225.0.0.101), 00:00:46/00:02:13, flags: PA
Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.13.3
Outgoing interface list: Null

```

## Spine Switch 2

The following example shows the output for the **show ip ospf neighbor** command on spine switch 2:

```
Spine-02# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.255.4	0	FULL/ -	00:00:37	172.16.24.4	GigabitEthernet1/0/2
172.16.255.3	0	FULL/ -	00:00:32	172.16.23.3	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on spine switch 2:

```

Spine-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 91, main routing table version 91
18 network entries using 6192 bytes of memory
38 path entries using 7904 bytes of memory
27/13 BGP path/bestpath attribute entries using 7776 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
12 BGP extended community entries using 640 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 22536 total bytes of memory
BGP activity 20/2 prefixes, 76/38 paths, scan interval 60 secs
18 networks peaked at 20:36:02 May 27 2020 UTC (5d18h ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   9183   9196     91    0    0 5d18h    18
172.16.255.3  4      65002   8443   8442     91    0    0 5d07h     9
172.16.255.4  4      65002   8442   8446     91    0    0 5d07h    11

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 2 for host device 1:

```

Spine-02# bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,

```

```

version 74
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    3          4
Refresh Epoch 1
65002
  172.16.254.4 (metric 2) (via default) from 172.16.255.1 (172.16.255.1)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
  Router MAC:7C21.0DBD.9548
  rx pathid: 0, tx pathid: 0
  net: 0x7FB64B5D07C0, path: 0x7FB64B5DFA08, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on May 28 2020 07:30:01 UTC
Refresh Epoch 1
65002
  172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
  Origin incomplete, metric 0, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
  Router MAC:7C21.0DBD.9548
  rx pathid: 0, tx pathid: 0x0
  net: 0x7FB64B5D07C0, path: 0x7FB64B5E01A0, pathext: 0x7FB64B5F1498
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on May 28 2020 07:28:25 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 2 for host device 2:

```

Spine-02# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 88
Paths: (2 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    3          4
Refresh Epoch 2
65002
  172.16.254.3 (metric 2) (via default) from 172.16.255.3 (172.16.255.3)
  Origin incomplete, metric 0, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0x0
  net: 0x7FB64B5D1580, path: 0x7FB64B5E0D70, pathext: 0x7FB64B5F19D8
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on May 28 2020 07:29:33 UTC
Refresh Epoch 1
65002
  172.16.254.3 (metric 2) (via default) from 172.16.255.1 (172.16.255.1)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65002:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0
  net: 0x7FB64B5D1580, path: 0x7FB64B5E0AE8, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on May 28 2020 07:28:18 UTC

```

The following example shows the output for the **show ip pim neighbor** command on spine switch 2:

```
Spine-02# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface          Uptime/Expires    Ver   DR
Address
172.16.23.3       GigabitEthernet1/0/1  5d19h/00:01:33   v2    1 / DR S P G
172.16.24.4       GigabitEthernet1/0/2  5d19h/00:01:18   v2    1 / DR S P G
```

The following example shows the output for the **show ip pim rp mapping** command on spine switch 2:

```
Spine-02# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)
```

The following example shows the output for the **show ip ro** command on spine switch 2:

```
Spine-02# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Loopback2
    Route metric is 0, traffic share count is 1
```

The following example shows the output for the **show ip msdp summary** command on spine switch 2:

```
Spine-02# show ip msdp summary
MSDP Peer Status Summary
Peer Address    AS    State    Uptime/  Reset SA    Peer Name
                AS    State    Downtime Count Count
172.16.254.1    65001 Up       5d19h   0      2      ?
```

The following example shows the output for the **show ip msdp sa-cache** command on spine switch 2:

```
Spine-02# show ip msdp sa-cache
MSDP Source-Active Cache - 2 entries
(172.16.254.3, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:04:07/00:05:17, Peer 172.16.254.1
(172.16.254.4, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:03:45/00:05:20, Peer 172.16.254.1
```

The following example shows the output for the **show ip mroute** command on spine switch 2:

```
Spine-02# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
```

```

N - Received BGP Shared-Tree Prune, n - BGP C-Route suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 5d19h/00:03:21, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 5d19h/00:03:15
    GigabitEthernet1/0/1, Forward/Sparse, 5d19h/00:03:21

(172.16.254.4, 225.0.0.101), 00:00:40/00:02:19, flags: A
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 00:00:40/00:03:21

(172.16.254.3, 225.0.0.101), 00:00:52/00:02:07, flags: A
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:00:52/00:03:15

```

## Leaf Switch 1

The following example shows the output for the **show ip ospf neighbor** command on leaf switch 1:

```
Leaf-01# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.255.2	0	FULL/ -	00:00:38	172.16.23.2	GigabitEthernet1/0/2
172.16.255.1	0	FULL/ -	00:00:32	172.16.13.1	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on leaf switch 1:

```
Leaf-01# show bgp l2vpn evpn summary
```

```

BGP router identifier 172.16.255.3, local AS number 65002
BGP table version is 32, main routing table version 32
27 network entries using 9288 bytes of memory
38 path entries using 7904 bytes of memory
16/15 BGP path/bestpath attribute entries using 4608 bytes of memory
1 BGP AS-PATH entries using 40 bytes of memory
13 BGP extended community entries using 664 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 22504 total bytes of memory
BGP activity 395/362 prefixes, 918/872 paths, scan interval 60 secs
27 networks peaked at 13:15:47 May 26 2020 UTC (1w0d ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	8457	8446	32	0	0	5d07h	9
172.16.255.2	4	65001	8443	8444	32	0	0	5d07h	11

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 1 for route type 2 and the IP address of host device 2:

```

Leaf-01# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 22
Paths: (1 available, best #1, table evi_101)
  Not advertised to any peer
  Refresh Epoch 1
  65001 65002, imported path from
[2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24 (global)
  172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
    Origin incomplete, localpref 100, valid, external, best
    EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
    Extended Community: RT:1:1 RT:65002:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
    rx pathid: 0, tx pathid: 0x0
    net: 0x7F575E47B150, path: 0x7F575E1EF800, pathext: 0x7F575E201C08, exp_net:
0x7F575E479470
    flags: net: 0x0, path: 0x40000000000003, pathext: 0x81
    Updated on May 28 2020 07:25:32 UTC
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 10
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    19
  Refresh Epoch 2
  65001 65002
    172.16.254.4 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
      Origin incomplete, localpref 100, valid, external
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65002:101 ENCAP:8
        Router MAC:7C21.0DBD.9548
      rx pathid: 0, tx pathid: 0
      net: 0x7F575E479470, path: 0x7F575E1EFD10, pathext: 0x0
      flags: net: 0x0, path: 0x3, pathext: 0x0
      Updated on May 28 2020 07:26:48 UTC
  Refresh Epoch 1
  65001 65002
    172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, localpref 100, valid, external, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65002:101 ENCAP:8
        Router MAC:7C21.0DBD.9548

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 1 for route type 2 and the IP address of host device 1:

```

Leaf-01# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 4
Paths: (1 available, best #1, table evi_101)
  Advertised to update-groups:
    19
  Refresh Epoch 1
  Local
  :: (via default) from 0.0.0.0 (172.16.255.3)
    Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
    EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
    Extended Community: RT:1:1 RT:65002:101 ENCAP:8
      Router MAC:10B3.D56A.8FC8
  Local irb vxlan vtep:
    vrf:green, l3-vni:50901
    local router mac:10B3.D56A.8FC8
    core-irb interface:Vlan901
    vtep-ip:172.16.254.3

```

```

rx pathid: 0, tx pathid: 0x0
net: 0x7F575E479B50, path: 0x7F575E1F0580, pathext: 0x7F575E201CC8
flags: net: 0x0, path: 0x4000028000003, pathext: 0x81
Updated on May 28 2020 07:25:30 UTC

```

The following example shows the output for the **show ip pim neighbor** command on leaf switch 1:

```

Leaf-01# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor      Interface      Uptime/Expires    Ver    DR
Address
172.16.13.1   GigabitEthernet1/0/1    5d19h/00:01:38    v2     1 / S P G
172.16.23.2   GigabitEthernet1/0/2    5d19h/00:01:17    v2     1 / S P G

```

The following example shows the output for the **show ip pim rp mapping** command on leaf switch 1:

```

Leaf-01# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)

```

The following example shows the output for the **show ip ro** command on leaf switch 1:

```

Leaf-01# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 172.16.23.2 on GigabitEthernet1/0/2, 5d19h ago
  Routing Descriptor Blocks:
    172.16.23.2, from 172.16.255.2, 5d19h ago, via GigabitEthernet1/0/2
      Route metric is 2, traffic share count is 1
    * 172.16.13.1, from 172.16.255.1, 5d19h ago, via GigabitEthernet1/0/1
      Route metric is 2, traffic share count is 1

```

The following example shows the output for the **show ip rpf** command on leaf switch 1:

```

Leaf-01# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
  RPF interface: GigabitEthernet1/0/2
  RPF neighbor: ? (172.16.23.2)
  RPF route/mask: 172.16.255.255/32
  RPF type: unicast (ospf 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base

```

The following example shows the output for the **show ip mroute** command on leaf switch 1:

```

Leaf-01# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,

```

```

G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 7w4d/stopped, RP 172.16.255.255, flags: SJCFx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
Tunnel0, Forward/Sparse-Dense, 1w0d/00:00:40

(172.16.254.4, 225.0.0.101), 00:01:22/00:01:37, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
Tunnel0, Forward/Sparse-Dense, 00:01:22/00:01:37

(172.16.254.3, 225.0.0.101), 00:01:35/00:01:24, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
GigabitEthernet1/0/2, Forward/Sparse, 00:01:35/00:02:53
    
```

## Leaf Switch 2

The following example shows the output for the **show ip ospf neighbor** command on leaf switch 2:

```
Leaf-02# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.255.2	0	FULL/ -	00:00:34	172.16.24.2	GigabitEthernet1/0/2
172.16.255.1	0	FULL/ -	00:00:31	172.16.14.1	GigabitEthernet1/0/1

The following example shows the output for the **show bgp l2vpn evpn summary** command on leaf switch 2:

```
Leaf-02# show bgp l2vpn evpn summary
```

```

BGP router identifier 172.16.255.4, local AS number 65002
BGP table version is 28, main routing table version 28
25 network entries using 8600 bytes of memory
34 path entries using 7072 bytes of memory
16/15 BGP path/bestpath attribute entries using 4608 bytes of memory
1 BGP AS-PATH entries using 40 bytes of memory
13 BGP extended community entries using 664 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 20984 total bytes of memory
BGP activity 199/168 prefixes, 638/596 paths, scan interval 60 secs
25 networks peaked at 13:20:44 May 26 2020 UTC (1w0d ago)
    
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	8449	8447	28	0	0	5d07h	9
172.16.255.2	4	65001	8448	8443	28	0	0	5d07h	7

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 2 for route type 2 and the IP address of host device 1:

```

Leaf-02# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 4
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    7
  Refresh Epoch 2
  65001 65002
    172.16.254.3 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, localpref 100, valid, external
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65002:101 ENCAP:8
      Router MAC:10B3.D56A.8FC8
      rx pathid: 0, tx pathid: 0
      net: 0x7F84BB3C4290, path: 0x7F84BB49BF98, pathext: 0x0
      flags: net: 0x0, path: 0x3, pathext: 0x0
      Updated on May 28 2020 07:31:42 UTC
  Refresh Epoch 1
  65001 65002
    172.16.254.3 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
      Origin incomplete, localpref 100, valid, external, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65002:101 ENCAP:8
      Router MAC:10B3.D56A.8FC8
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F84BB3C4290, path: 0x7F84BB49D9C0, pathext: 0x7F84BB594138
      flags: net: 0x0, path: 0x3, pathext: 0x81
      Updated on May 28 2020 07:31:37 UTC
BGP routing table entry for [2][172.16.254.4:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 20
Paths: (1 available, best #1, table evi_101)
  Not advertised to any peer
  Refresh Epoch 1
  65001 65002, imported path from
  [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24 (global)
    172.16.254.3 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
      Origin incomplete, localpref 100, valid, external, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65002:101 ENCAP:8
      Router MAC:10B3.D56A.8FC8

```

The following example shows the output for the **show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12** command on leaf switch 2 for route type 2 and the IP address of host device 2:

```

Leaf-02# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 10
Paths: (1 available, best #1, table evi_101)
  Advertised to update-groups:
    7
  Refresh Epoch 1
  Local
  :: (via default) from 0.0.0.0 (172.16.255.4)
    Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
    EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
    Extended Community: RT:1:1 RT:65002:101 ENCAP:8
    Router MAC:7C21.0DBD.9548
  Local irb vxlan vtep:
    vrf:green, l3-vni:50901
    local router mac:7C21.0DBD.9548
    core-irb interface:Vlan901
    vtep-ip:172.16.254.4
    rx pathid: 0, tx pathid: 0x0

```



```
net: 0x7F84BB3C4970, path: 0x7F84BB49CDF0, pathext: 0x7F84BB593CB8
flags: net: 0x0, path: 0x4000028000003, pathext: 0x81
Updated on May 28 2020 07:30:04 UTC
```

The following example shows the output for the **show ip pim neighbor** command on leaf switch 2:

```
Leaf-02# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface                Uptime/Expires    Ver  DR
Address
172.16.14.1       GigabitEthernet1/0/1     5d19h/00:01:22   v2   1 / S P G
172.16.24.2       GigabitEthernet1/0/2     5d19h/00:01:27   v2   1 / S P G
```

The following example shows the output for the **show ip pim rp mapping** command on leaf switch 2:

```
Leaf-02# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
RP: 172.16.255.255 (?)
```

The following example shows the output for the **show ip ro** command on leaf switch 2:

```
Leaf-02# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 172.16.24.2 on GigabitEthernet1/0/2, 5d19h ago
  Routing Descriptor Blocks:
    172.16.24.2, from 172.16.255.2, 5d19h ago, via GigabitEthernet1/0/2
      Route metric is 2, traffic share count is 1
    * 172.16.14.1, from 172.16.255.1, 5d19h ago, via GigabitEthernet1/0/1
      Route metric is 2, traffic share count is 1
```

The following example shows the output for the **show ip rpf** command on leaf switch 2:

```
Leaf-02# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
  RPF interface: GigabitEthernet1/0/2
  RPF neighbor: ? (172.16.24.2)
  RPF route/mask: 172.16.255.255/32
  RPF type: unicast (ospf 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

The following example shows the output for the **show ip mroute** command on leaf switch 2:

```
Leaf-02# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
```

```

Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 1w5d/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 1w5d/00:00:06

(172.16.254.4, 225.0.0.101), 00:01:56/00:01:03, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:01:56/00:02:32

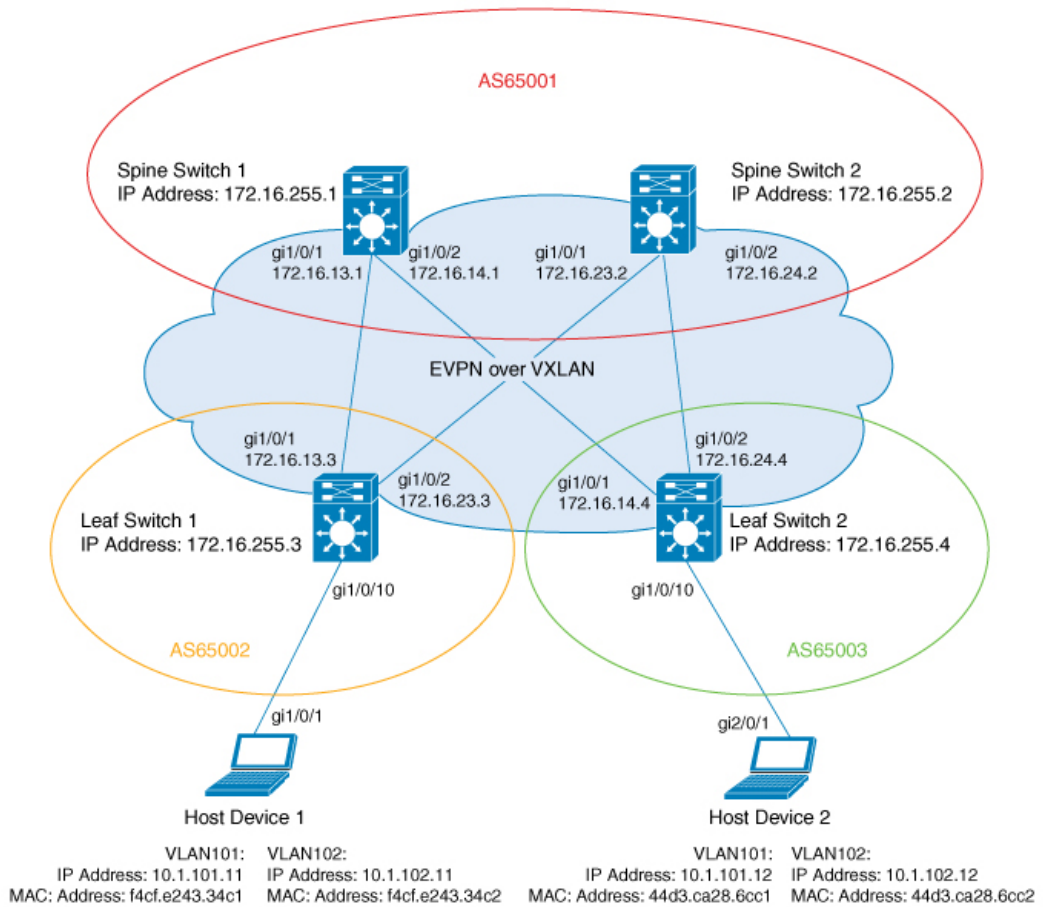
(172.16.254.3, 225.0.0.101), 00:02:09/00:00:50, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:02:09/00:00:50

```

## Configuration Example for Spine Switches Using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System

This section provides an example to show how spine switches are configured in a BGP EVPN VXLAN fabric using eBGP when the spine switches are in one autonomous system and the leaf switches are in a different autonomous system. The example shows how to configure spine switches and verify the configuration for the topology shown below:

Figure 7: BGP EVPN VXLAN Fabric with the Spine Switches in one Autonomous System and each Leaf Switch in a Different Autonomous System



The topology shows an EVPN VXLAN network with two leaf switches (leaf switch 1 and leaf switch 2) and two spine switches (spine switch 1 and spine switch 2). Spine switch 1 and spine switch 2 are in autonomous system AS65001. Leaf switch 1 is in autonomous system AS65002. Leaf switch 2 is in autonomous system AS65003. Spine switch 1 and spine switch 2 are BGP route servers and are not route reflector clients to each other. Multicast Source Discovery Protocol (MSDP) is configured between spine switch 1 and spine switch 2 for source synchronization. Protocol Independent Multicast (PIM) is enabled on the interfaces that connect leaf switches and spine switches. Static RP is configured in the network and the underlay network uses multicast forwarding mechanism to forward BUM traffic.



**Note** You must run the **rewrite-evpn-rt-asn** command in the L2VPN EVPN address family configuration mode on the leaf switches to allow processing of BGP updates that have a different autonomous system number.



**Note** You must manually run the **no bgp default route-target filter** command in router configuration mode on the spine switches.



---

**Note** You must configure eBGP multihop on the leaf and spine switches for the fabric to function.

---

The following tables provide sample configurations for the devices in the topology above.

**Table 13: Configuring Spine Switch 1 and Spine Switch 2 using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System**

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.1 255.255.255.255 ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.14.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.1 ! router bgp 65001 bgp router-id 172.16.255.1 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 neighbor 172.16.255.3 remote-as 65002 neighbor 172.16.255.3 ebgp-multihop 255 neighbor 172.16.255.3 update-source Loopback0 neighbor 172.16.255.4 remote-as 65003 neighbor 172.16.255.4 ebgp-multihop 255 neighbor 172.16.255.4 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! interface Loopback0 ip address 172.16.255.2 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.2 255.255.255.255 ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.23.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.2 ! router bgp 65001 bgp router-id 172.16.255.2 bgp log-neighbor-changes no bgp default ipv4-unicast no bgp default route-target filter neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.3 remote-as 65002 neighbor 172.16.255.3 ebgp-multihop 255 neighbor 172.16.255.3 update-source Loopback0 neighbor 172.16.255.4 remote-as 65003 neighbor 172.16.255.4 ebgp-multihop 255 neighbor 172.16.255.4 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>

**Configuration Example for Spine Switches Using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System**

Spine Switch 1	Spine Switch 2
<pre> address-family l2vpn evpn rewrite-evpn-rt-asn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-map BGP-NHU out neighbor 172.16.255.3 activate neighbor 172.16.255.3 send-community both neighbor 172.16.255.3 route-map BGP-NHU out neighbor 172.16.255.4 activate neighbor 172.16.255.4 send-community both neighbor 172.16.255.4 route-map BGP-NHU out exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ip msdp peer 172.16.254.2 connect-source Loopback1 remote-as 65001 ip msdp cache-sa-state ! route-map BGP-NHU permit 10 set ip next-hop unchanged ! end Spine-01# </pre>	<pre> address-family l2vpn evpn rewrite-evpn-rt-asn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-map BGP-NHU out neighbor 172.16.255.3 activate neighbor 172.16.255.3 send-community both neighbor 172.16.255.3 route-map BGP-NHU out neighbor 172.16.255.4 activate neighbor 172.16.255.4 send-community both neighbor 172.16.255.4 route-map BGP-NHU out exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ip msdp peer 172.16.254.1 connect-source Loopback1 remote-as 65001 ip msdp cache-sa-state ! route-map BGP-NHU permit 10 set ip next-hop unchanged ! end Spine-02# </pre>

**Table 14: Configuring Leaf Switch 1 and Leaf Switch 2 using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System**

Leaf Switch 1	Leaf Switch 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type static ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.4 255.255.255.255 ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! </pre>

**Configuration Example for Spine Switches Using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System**

Leaf Switch 1	Leaf Switch 2
<pre> interface GigabitEthernet1/0/2 no switchport ip address 172.16.23.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ipv6 enable no autostate ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 10102 ingress-replication member vni 50901 vrf green ! router ospf 1 router-id 172.16.255.3 ! router bgp 65002 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 ebgp-multihop 255 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 ebgp-multihop 255 neighbor 172.16.255.2 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn rewrite-evpn-rt-asn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! </pre>	<pre> interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ipv6 enable no autostate ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 member vni 50901 vrf green member vni 10102 ingress-replication ! router ospf 1 router-id 172.16.255.4 ! router bgp 65003 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 ebgp-multihop 255 neighbor 172.16.255.1 update-source Loopback0 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 ebgp-multihop 255 neighbor 172.16.255.2 update-source Loopback0 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn rewrite-evpn-rt-asn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! </pre>



Leaf Switch 1	Leaf Switch 2
<pre> address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ! end Leaf-01#                     </pre>	<pre> address-family ipv6 vrf green redistribute connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 172.16.255.255 ! end Leaf-02#                     </pre>

The following examples provide sample outputs for **show** commands on the devices in the topology configured in the preceding tables:

### Spine Switch 1

The following example shows the output for the **show ip ospf neighbor** command on spine switch 1:

```

Spine-01# show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address          Interface
172.16.255.4     0    FULL/ -         00:00:34   172.16.14.4     GigabitEthernet1/0/2
172.16.255.3     0    FULL/ -         00:00:38   172.16.13.3     GigabitEthernet1/0/1
    
```

The following example shows the output for the **show bgp l2vpn evpn summary** command on spine switch 1:

```

Spine-01# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 19, main routing table version 19
18 network entries using 6192 bytes of memory
38 path entries using 7904 bytes of memory
45/15 BGP path/bestpath attribute entries using 12960 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
24 BGP extended community entries using 1280 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 28384 total bytes of memory
BGP activity 94/76 prefixes, 293/255 paths, scan interval 60 secs
18 networks peaked at 21:10:53 Jun 4 2020 UTC (2d23h ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2  4      65001   28     27     19     0    0 00:08:49    18
172.16.255.3  4      65002   35     27     19     0    0 00:08:54     9
172.16.255.4  4      65003   34     27     19     0    0 00:08:54    11
    
```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 1 for route type 2 and the IP address of host device 2:

```

Spine-01# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 10
Paths: (2 available, best #2, table EVPN-BGP-Table)
    
```

```

Advertised to update-groups:
  11      13
Refresh Epoch 1
65003
  172.16.254.4 (metric 2) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
  Router MAC:7C21.0DBD.9548
  rx pathid: 0, tx pathid: 0
  net: 0x7F7898C7FEF0, path: 0x7F7898C8E578, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on Jun 7 2020 20:42:32 UTC
Refresh Epoch 2
65003
  172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
  Origin incomplete, metric 0, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
  Router MAC:7C21.0DBD.9548
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F7898C7FEF0, path: 0x7F7898C8E728, pathext: 0x7F7898CAE8E0
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on Jun 7 2020 20:41:30 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 1 for route type 2 and the IP address of host device 1:

```

Spine-01# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 4
Paths: (2 available, best #2, table EVPN-BGP-Table)
Advertised to update-groups:
  11      13
Refresh Epoch 1
65002
  172.16.254.3 (metric 2) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0
  net: 0x7F7898C7F290, path: 0x7F7898C8FEC8, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on Jun 7 2020 20:42:32 UTC
Refresh Epoch 2
65002
  172.16.254.3 (metric 2) (via default) from 172.16.255.3 (172.16.255.3)
  Origin incomplete, metric 0, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65001:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F7898C7F290, path: 0x7F7898C8E218, pathext: 0x7F7898CAEE20
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on Jun 7 2020 20:41:30 UTC

```

The following example shows the output for the **show ip pim neighbor** command on spine switch 1:

```

Spine-01# show ip pim neighbor
PIM Neighbor Table

```

```

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor      Interface      Uptime/Expires      Ver      DR
Address
172.16.13.3   GigabitEthernet1/0/1  1w4d/00:01:37      v2       1 / DR S P G
172.16.14.4   GigabitEthernet1/0/2  1w4d/00:01:39      v2       1 / DR S P G

```

The following example shows the output for the **show ip pim rp mapping** command on spine switch 1:

```

Spine-01# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)

```

The following example shows the output for the **show ip ro** command on spine switch 1:

```

Spine-01# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Loopback2
    Route metric is 0, traffic share count is 1

```

The following example shows the output for the **show ip msdp summary** command on spine switch 1:

```

Spine-01# show ip msdp summary
MSDP Peer Status Summary
Peer Address      AS      State      Uptime/  Reset SA      Peer Name
                  Downtime Count Count
172.16.254.2      65001  Up         1w4d     0      2      ?

```

The following example shows the output for the **show ip msdp sa-cache** command on spine switch 1:

```

Spine-01# show ip msdp sa-cache
MSDP Source-Active Cache - 2 entries
(172.16.254.3, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:01:07/00:05:06, Peer 172.16.254.2
(172.16.254.4, 225.0.0.101), RP 172.16.255.255, BGP/AS 0, 00:00:45/00:05:14, Peer 172.16.254.2

```

The following example shows the output for the **show ip rpf** command on spine switch 1:

```

Spine-01# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
  RPF interface: Loopback2
  RPF neighbor: ? (172.16.255.255) - directly connected
  RPF route/mask: 172.16.255.255/32
  RPF type: multicast (connected)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base

```

The following example shows the output for the **show ip mroute** command on spine switch 1:

```

Spine-01# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 00:01:07/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(172.16.254.4, 225.0.0.101), 00:00:45/00:02:14, flags: PA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null

(172.16.254.3, 225.0.0.101), 00:01:07/00:01:52, flags: PA
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.13.3
  Outgoing interface list: Null

```

## Spine Switch 2

The following example shows the output for the **show ip ospf neighbor** command on spine switch 2:

```

Spine-02# show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address        Interface
172.16.255.4    0    FULL/ -         00:00:32   172.16.24.4   GigabitEthernet1/0/2
172.16.255.3    0    FULL/ -         00:00:34   172.16.23.3   GigabitEthernet1/0/1

```

The following example shows the output for the **show bgp l2vpn evpn summary** command on spine switch 2:

```

Spine-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 19, main routing table version 19
18 network entries using 6192 bytes of memory
38 path entries using 7904 bytes of memory
45/15 BGP path/bestpath attribute entries using 12960 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
24 BGP extended community entries using 1280 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 28384 total bytes of memory
BGP activity 56/38 prefixes, 244/206 paths, scan interval 60 secs
18 networks peaked at 21:11:25 Jun 4 2020 UTC (2d23h ago)

Neighbor      V      AS  MsgRcvd  MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001    27      28      19    0    0 00:08:54    18

```

```

172.16.255.3    4      65002    30      27      19      0      0 00:08:54      9
172.16.255.4    4      65003    30      27      19      0      0 00:08:54     11

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 2 for route type 2 and the IP address of host device 2:

```

Spine-02# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 10
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    9          10
  Refresh Epoch 1
  65003
    172.16.254.4 (metric 2) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
      rx pathid: 0, tx pathid: 0
      net: 0x7FB6494C8550, path: 0x7FB64B6D21A8, pathext: 0x0
      flags: net: 0x0, path: 0x3, pathext: 0x0
      Updated on Jun 7 2020 20:43:06 UTC
  Refresh Epoch 2
  65003
    172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
      rx pathid: 0, tx pathid: 0x0
      net: 0x7FB6494C8550, path: 0x7FB64B6D3870, pathext: 0x7FB6494D8788
      flags: net: 0x0, path: 0x3, pathext: 0x81
      Updated on Jun 7 2020 20:42:08 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on spine switch 2 for route type 2 and the IP address of host device 1:

```

Spine-02# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 4
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    9          10
  Refresh Epoch 1
  65002
    172.16.254.3 (metric 2) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:10B3.D56A.8FC8
      rx pathid: 0, tx pathid: 0
      net: 0x7FB6494C86B0, path: 0x7FB64B6D25E0, pathext: 0x0
      flags: net: 0x0, path: 0x3, pathext: 0x0
      Updated on Jun 7 2020 20:43:06 UTC
  Refresh Epoch 2
  65002
    172.16.254.3 (metric 2) (via default) from 172.16.255.3 (172.16.255.3)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:10B3.D56A.8FC8

```

```

rx pathid: 0, tx pathid: 0x0
net: 0x7FB6494C86B0, path: 0x7FB64B6D31B0, pathext: 0x7FB6494D8CC8
flags: net: 0x0, path: 0x3, pathext: 0x81
Updated on Jun 7 2020 20:42:08 UTC

```

The following example shows the output for the **show ip pim neighbor** command on spine switch 2:

```

Spine-02# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface          Uptime/Expires    Ver   DR
Address
172.16.23.3       GigabitEthernet1/0/1  00:34:48/00:01:27 v2    1 / DR S P G
172.16.24.4       GigabitEthernet1/0/2  1w4d/00:01:36    v2    1 / DR S P G

```

The following example shows the output for the **show ip pim rp mapping** command on spine switch 2:

```

Spine-02# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)

```

The following example shows the output for the **show ip ro** command on spine switch 2:

```

Spine-02# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Loopback2
    Route metric is 0, traffic share count is 1

```

The following example shows the output for the **show ip msdp summary** command on spine switch 2:

```

Spine-02# show ip msdp summary
MSDP Peer Status Summary
Peer Address      AS      State      Uptime/  Reset SA  Peer Name
                  Downtime Count Count
172.16.254.1     65001  Up         1w4d    0        2        ?

```

The following example shows the output for the **show ip msdp sa-cache** command on spine switch 2:

```

Spine-02# show ip msdp sa-cache
RPF information for ? (172.16.255.255)
  RPF interface: Loopback2
  RPF neighbor: ? (172.16.255.255) - directly connected
  RPF route/mask: 172.16.255.255/32
  RPF type: multicast (connected)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base

```

The following example shows the output for the **show ip mroute** command on spine switch 2:

```
Spine-02# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 1w4d/00:03:27, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 00:34:36/00:03:22
    GigabitEthernet1/0/2, Forward/Sparse, 2d23h/00:03:27

(172.16.254.4, 225.0.0.101), 00:00:50/00:02:09, flags: A
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 00:00:50/00:03:22

(172.16.254.3, 225.0.0.101), 00:01:11/00:01:47, flags: A
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:01:11/00:03:27
```

### Leaf Switch 1

The following example shows the output for the **show ip ospf neighbor** command on leaf switch 1:

```
Leaf-01# show ip ospf neighbor
Neighbor ID      Pri  State           Dead Time   Address         Interface
172.16.255.2     0    FULL/ -         00:00:31    172.16.23.2    GigabitEthernet1/0/2
172.16.255.1     0    FULL/ -         00:00:34    172.16.13.1    GigabitEthernet1/0/1
```

The following example shows the output for the **show bgp l2vpn evpn summary** command on leaf switch 1:

```
Leaf-01# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.3, local AS number 65002
BGP table version is 99, main routing table version 99
27 network entries using 9288 bytes of memory
36 path entries using 7488 bytes of memory
22/15 BGP path/bestpath attribute entries using 6336 bytes of memory
1 BGP AS-PATH entries using 40 bytes of memory
18 BGP extended community entries using 944 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 24096 total bytes of memory
BGP activity 483/450 prefixes, 1123/1081 paths, scan interval 60 secs
```

27 networks peaked at 13:15:47 May 26 2020 UTC (1w5d ago)

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	27	34	99	0	0	00:08:30	9
172.16.255.2	4	65001	27	29	99	0	0	00:08:25	9

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 1 for route type 2 and the IP address of host device 2:

```
Leaf-01# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 93
Paths: (1 available, best #1, table evi_101)
  Not advertised to any peer
  Refresh Epoch 1
  65001 65003, imported path from
[2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24 (global)
  172.16.254.4 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
    Origin incomplete, localpref 100, valid, external, best
    EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
    Extended Community: RT:1:1 RT:65002:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
    rx pathid: 0, tx pathid: 0x0
    net: 0x7F575E4795D0, path: 0x7F575E1EFC38, pathext: 0x7F575E201308, exp_net:
0x7F575E47AA70
    flags: net: 0x0, path: 0x40000000000003, pathext: 0x81
    Updated on Jun 7 2020 20:40:17 UTC
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 84
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    21
  Refresh Epoch 1
  65001 65003
    172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, localpref 100, valid, external
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65002:101 ENCAP:8
```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 1 for route type 2 and the IP address of host device 1:

```
Leaf-01# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 4
Paths: (1 available, best #1, table evi_101)
  Advertised to update-groups:
    21
  Refresh Epoch 1
  Local
  :: (via default) from 0.0.0.0 (172.16.255.3)
    Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
    EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
    Extended Community: RT:1:1 RT:65002:101 ENCAP:8
      Router MAC:10B3.D56A.8FC8
    Local irb vxlan vtep:
      vrf:green, l3-vni:50901
      local router mac:10B3.D56A.8FC8
      core-irb interface:Vlan901
      vtep-ip:172.16.254.3
    rx pathid: 0, tx pathid: 0x0
    net: 0x7F575E47ABD0, path: 0x7F575E1F13D8, pathext: 0x7F575E201968
```



```
flags: net: 0x0, path: 0x4000028000003, pathext: 0x81
Updated on Jun 4 2020 21:26:02 UTC
```

The following example shows the output for the **show l2vpn evpn mac ip** command on leaf switch 1:

```
Leaf-01# show l2vpn evpn mac ip
-----
IP Address          EVI   VLAN  MAC Address      Next Hop(s)
-----
10.1.101.11        101   101   f4cf.e243.34c1  Gi1/0/10:101
10.1.101.12        101   101   44d3.ca28.6cc1  172.16.254.4
10.1.102.11        102   102   f4cf.e243.34c2  Gi1/0/10:102
10.1.102.12        102   102   44d3.ca28.6cc2  172.16.254.4
```

The following example shows the output for the **show ip pim neighbor** command on leaf switch 1:

```
Leaf-01# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface          Uptime/Expires    Ver  DR
Address
172.16.13.1       GigabitEthernet1/0/1  1w4d/00:01:17    v2   1 / S P G
172.16.23.2       GigabitEthernet1/0/2  00:34:19/00:01:24 v2   1 / S P G
```

The following example shows the output for the **show ip pim rp mapping** command on leaf switch 1:

```
Leaf-01# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)
```

The following example shows the output for the **show ip ro** command on leaf switch 1:

```
Leaf-01# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 172.16.23.2 on GigabitEthernet1/0/2, 00:34:08 ago
  Routing Descriptor Blocks:
    172.16.23.2, from 172.16.255.2, 00:34:08 ago, via GigabitEthernet1/0/2
      Route metric is 2, traffic share count is 1
    * 172.16.13.1, from 172.16.255.1, 1w4d ago, via GigabitEthernet1/0/1
      Route metric is 2, traffic share count is 1
```

The following example shows the output for the **show ip rpf** command on leaf switch 1:

```
Leaf-01# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
  RPF interface: GigabitEthernet1/0/2
  RPF neighbor: ? (172.16.23.2)
  RPF route/mask: 172.16.255.255/32
  RPF type: unicast (ospf 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

The following example shows the output for the **show ip mroute** command on leaf switch 1:

```
Leaf-01# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 8w2d/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 1w5d/00:01:01

(172.16.254.4, 225.0.0.101), 00:00:21/00:02:38, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:00:21/00:02:38

(172.16.254.3, 225.0.0.101), 00:00:43/00:02:46, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:00:43/00:02:46
```

## Leaf Switch 2

The following example shows the output for the **show ip ospf neighbor** command on leaf switch 2:

```
Leaf-02# show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address          Interface
172.16.255.2     0    FULL/ -         00:00:36   172.16.24.2     GigabitEthernet1/0/2
172.16.255.1     0    FULL/ -         00:00:31   172.16.14.1     GigabitEthernet1/0/1
```

The following example shows the output for the **show bgp l2vpn evpn summary** command on leaf switch 2:

```
Leaf-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.4, local AS number 65003
BGP table version is 83, main routing table version 83
25 network entries using 8600 bytes of memory
36 path entries using 7488 bytes of memory
23/15 BGP path/bestpath attribute entries using 6624 bytes of memory
1 BGP AS-PATH entries using 40 bytes of memory
19 BGP extended community entries using 984 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 23736 total bytes of memory
BGP activity 95/64 prefixes, 207/163 paths, scan interval 60 secs
25 networks peaked at 21:31:21 Jun 4 2020 UTC (2d23h ago)
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	27	34	83	0	0	00:08:40	9
172.16.255.2	4	65001	27	29	83	0	0	00:08:35	9

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 2 for route type 2 and the IP address of host device 2:

```
Leaf-02# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.12
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.12]/24,
version 4
Paths: (1 available, best #1, table evi_101)
  Advertised to update-groups:
    2
  Refresh Epoch 1
  Local
  :: (via default) from 0.0.0.0 (172.16.255.4)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65003:101 ENCAP:8
  Router MAC:7C21.0DBD.9548
  Local irb vxlan vtep:
    vrf:green, l3-vni:50901
    local router mac:7C21.0DBD.9548
    core-irb interface:Vlan901
    vtep-ip:172.16.254.4
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F84B8F2D778, path: 0x7F84BB3149F0, pathext: 0x7F84BB526788
  flags: net: 0x0, path: 0x4000028000003, pathext: 0x81
  Updated on Jun 4 2020 21:30:20 UTC
```

The following example shows the output for the **show bgp l2vpn evpn route-type** command on leaf switch 2 for route type 2 and the IP address of host device 1:

```
Leaf-02# show bgp l2vpn evpn route-type 2 0 f4cfe24334c1 10.1.101.11
BGP routing table entry for [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 72
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Advertised to update-groups:
    2
  Refresh Epoch 1
  65001 65002
  172.16.254.3 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
  Origin incomplete, localpref 100, valid, external
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65003:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0
  net: 0x7F84B8F2E958, path: 0x7F84BB313FD0, pathext: 0x0
  flags: net: 0x0, path: 0x3, pathext: 0x0
  Updated on Jun 7 2020 20:44:45 UTC
  Refresh Epoch 1
  65001 65002
  172.16.254.3 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65003:101 ENCAP:8
  Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F84B8F2E958, path: 0x7F84BB313178, pathext: 0x7F84BB526548
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on Jun 7 2020 20:44:44 UTC
```

```

BGP routing table entry for [2][172.16.254.4:101][0][48][F4CFE24334C1][32][10.1.101.11]/24,
version 78
Paths: (1 available, best #1, table evi_101)
  Not advertised to any peer
  Refresh Epoch 1
  65001 65002, imported path from
[2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24 (global)
  172.16.254.3 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, localpref 100, valid, external, best
  EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
  Extended Community: RT:1:1 RT:65003:101 ENCAP:8
    Router MAC:10B3.D56A.8FC8
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F84B8F2D358, path: 0x7F84BB314258, pathext: 0x7F84BB5265A8, exp_net:
0x7F84B8F2E958
  flags: net: 0x0, path: 0x40000000000003, pathext: 0x81
  Updated on Jun 7 2020 20:44:44 UTC

```

The following example shows the output for the **show l2vpn evpn mac ip** command on leaf switch 2:

```

Leaf-02# show l2vpn evpn mac ip
IP Address          EVI    VLAN  MAC Address      Next Hop(s)
-----
10.1.101.11         101    101   f4cf.e243.34c1   172.16.254.3
10.1.101.12         101    101   44d3.ca28.6cc1   Gi1/0/10:101
10.1.102.11         102    102   f4cf.e243.34c2   172.16.254.3
10.1.102.12         102    102   44d3.ca28.6cc2   Gi1/0/10:102

```

The following example shows the output for the **show ip pim neighbor** command on leaf switch 2:

```

Leaf-02# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor      Interface      Uptime/Expires    Ver  DR
Address
172.16.14.1   GigabitEthernet1/0/1   1w4d/00:01:42    v2   1 / S P G
172.16.24.2   GigabitEthernet1/0/2   1w4d/00:01:19    v2   1 / S P G

```

The following example shows the output for the **show ip pim rp mapping** command on leaf switch 2:

```

Leaf-02# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 172.16.255.255 (?)

```

The following example shows the output for the **show ip ro** command on leaf switch 2:

```

Leaf-02# show ip ro 172.16.255.255
Routing entry for 172.16.255.255/32
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 172.16.14.1 on GigabitEthernet1/0/1, 3d00h ago
  Routing Descriptor Blocks:
  * 172.16.24.2, from 172.16.255.2, 3d00h ago, via GigabitEthernet1/0/2
    Route metric is 2, traffic share count is 1

```

```
172.16.14.1, from 172.16.255.1, 3d00h ago, via GigabitEthernet1/0/1
Route metric is 2, traffic share count is 1
```

The following example shows the output for the **show ip rpf** command on leaf switch 2:

```
Leaf-02# show ip rpf 172.16.255.255
RPF information for ? (172.16.255.255)
RPF interface: GigabitEthernet1/0/2
RPF neighbor: ? (172.16.24.2)
RPF route/mask: 172.16.255.255/32
RPF type: unicast (ospf 1)
Doing distance-preferred lookups across tables
RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

The following example shows the output for the **show ip mroute** command on leaf switch 2:

```
Leaf-02# show ip mroute 225.0.0.101
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry,
* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.0.0.101), 2w3d/stopped, RP 172.16.255.255, flags: SJCFx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
Outgoing interface list:
Tunnel0, Forward/Sparse-Dense, 2d23h/00:01:40

(172.16.254.4, 225.0.0.101), 00:00:31/00:02:58, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
GigabitEthernet1/0/2, Forward/Sparse, 00:00:31/00:02:58

(172.16.254.3, 225.0.0.101), 00:00:52/00:02:07, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
Outgoing interface list:
Tunnel0, Forward/Sparse-Dense, 00:00:52/00:02:07
```

**Configuration Example for Spine Switches Using eBGP when the Spine Switches are in one Autonomous System and each Leaf Switch is in a Different Autonomous System**



## CHAPTER 6

# Configuring DHCP Relay in a BGP EVPN VXLAN Fabric

---

- [Restrictions for DHCP Relay in a BGP EVPN VXLAN Fabric, on page 151](#)
- [Information About DHCP Relay in a BGP EVPN VXLAN Fabric, on page 151](#)
- [How to Configure DHCP Relay in a BGP EVPN VXLAN Fabric, on page 153](#)
- [Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric, on page 157](#)

## Restrictions for DHCP Relay in a BGP EVPN VXLAN Fabric

- DHCP relay in a BGP EVPN VXLAN fabric is supported in the following scenarios only when VRF-Lite is configured on the border VTEP and the border VTEP is connected to the DHCP server through an external router.
  - DHCP client in the tenant VRF and DHCP server in the Layer 3 default VRF
  - DHCP client in the tenant VRF and DHCP server in a different tenant VRF
  - DHCP client in the tenant VRF and DHCP server in a non-default non-VXLAN VRF
- DHCPv6 relay is not supported.

## Information About DHCP Relay in a BGP EVPN VXLAN Fabric

Networks use DHCP relay to forward DHCP packets between host devices and a DHCP server. In a BGP EVPN VXLAN fabric, you can configure a VTEP as a relay agent to provide DHCP relay services in a multi-tenant VXLAN environment.

When a network uses DHCP relay, DHCP messages move through the same switch in both directions. DHCP relay generally uses the gateway IP address (GiAddr) for scope selection and DHCP response messages. In a BGP EVPN VXLAN fabric that has distributed IP anycast gateway enabled, DHCP messages can return to any switch that hosts the respective GiAddr.

Deploying DHCP relay in an EVPN VXLAN network requires a different method for scope selection and a unique IP address for each switch in the network. The unique Loopback interface for a switch becomes the GiAddr that a switch uses to respond to the correct switch. DHCP option 82, also referred to as DHCP option VPN, is used for scope selection based on the Layer 2 VNI.

In a multi-tenant EVPN environment, DHCP relay uses the following sub-options of option 82:

- **Sub-Option 151(0x97)—Virtual Subnet Selection:**

The virtual subnet selection sub-option is used to convey VRF-related information to the DHCP server in an MPLS VPN and a VXLAN EVPN multi-tenant environment.

[RFC 6607](#) provides the definition for this sub-option.

- **Sub-Option 11(0xb)—Server ID Override**

The server identifier or server ID override sub-option allows the DHCP relay agent to specify a new value for the server ID option. The DHCP server inserts this new value in the reply packet. This sub-option allows the DHCP relay agent to act as the actual DHCP server. The DHCP relay agent begins to receive all the renew requests instead of the DHCP server. The server ID override sub-option contains the incoming interface IP address. The DHCP client accesses the DHCP relay agent using the incoming interface IP address. The DHCP client uses this information to send all the renew and release request packets to the DHCP relay agent. The DHCP relay agent adds all the appropriate sub-options and then forwards the renew and release request packets to the original DHCP server.

For this function, Cisco's proprietary implementation is sub-option 152(0x98). To implement the suboption and manage the function, run the **ip dhcp relay sub-option type cisco** command in global configuration mode on the VTEP that acts as the DHCP relay agent.

[RFC 5107](#) provides the definition for this sub-option.

- **Sub-Option 5(0x5)—Link Selection:**

The link selection sub-option provides a mechanism to separate the subnet or link, on which the DHCP client resides, from the GiAddr. The DHCP server uses this mechanism to communicate with the DHCP relay agent. The DHCP relay agent sets the sub-option to the correct subscriber subnet. The DHCP server then uses this value to assign an IP address different from the GiAddr. The DHCP relay agent sets the GiAddr to its own IP address to ensure that it is possible to forward the DHCP messages over the network.

For this function, Cisco's proprietary implementation is sub-option 150(0x96). To manage the function, run the **ip dhcp relay sub-option type cisco** command in global configuration mode on the VTEP that acts as the DHCP relay agent.

[RFC 3527](#) provides the definition for this sub-option.

## DHCP Relay on VTEPs

DHCP relay is generally configured on the default gateway that faces the DHCP client. You can configure a VTEP as a DHCP relay agent in different ways to automate IP addressing. The configuration depends on whether the DHCP server is present in the same network, the same VRF, or a different VRF compared to the DHCP client. When the DHCP server and DHCP client are in different VRFs, traffic is forwarded across the tenant or VRF boundaries.

The following are the common DHCP relay deployment scenarios for a BGP EVPN VXLAN fabric:

1. DHCP client in the tenant VRF and DHCP server in the Layer 3 default VRF
2. DHCP client in the tenant VRF and DHCP server in the same tenant VRF
3. DHCP client in the tenant VRF and DHCP server in a different tenant VRF
4. DHCP client in the tenant VRF and DHCP server in a non-default non-VXLAN VRF





**Note** The deployment scenarios 1, 3, and 4 are supported only when VRF-Lite is configured on the border VTEP and the border VTEP is connected to the DHCP server through an external router.

## How to Configure DHCP Relay in a BGP EVPN VXLAN Fabric

You must configure EVPN VXLAN Layer 2 and Layer 3 overlay networks before configuring BGP EVPN VXLAN interworking with DHCP relay. See [How to Configure EVPN VXLAN Integrated Routing and Bridging, on page 59](#) for detailed steps.

Perform the following set of procedures to configure BGP EVPN VLAN interworking with DHCP relay:

### Configuring DHCP Relay on a VTEP

To configure DHCP relay on a VTEP, perform the following steps:

#### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>ip dhcp relay information option vpn</b> <b>Example:</b> Device(config)# <b>ip dhcp relay information option vpn</b>	Enables the device to insert VPN suboptions into the DHCP relay agent information option in the messages forwarded to the DHCP server and sets the GiAddr on the outgoing interface towards the DHCP server.
<b>Step 4</b>	<b>ip dhcp relay information option</b> <b>Example:</b> Device(config)# <b>ip dhcp relay information option</b>	Enables the system to insert a DHCP relay agent information option in the messages forwarded to the DHCP server.
<b>Step 5</b>	<b>ip dhcp relay override gateway-ip-address link-selection</b> <b>Example:</b> Device(config)# <b>ip dhcp relay override giaddr link-selection</b>	Sets the gateway IP address as the IP address of the DHCP relay agent and configures the server to assign an IP address that is different from the GiAddr to the DHCP clients.

	Command or Action	Purpose
<b>Step 6</b>	<b>ip dhcp compatibility suboption server-override standard</b>  <b>Example:</b> Device(config)# ip dhcp compatibility suboption server-override standard	Configures the DHCP client to use the Internet Assigned Numbers Authority (IANA) standard relay agent server ID override suboption.
<b>Step 7</b>	<b>ip dhcp snooping vlan <i>vlan-id-list</i></b>  <b>Example:</b> Device(config)# ip dhcp snooping vlan 201-202	Enables DHCP snooping on the specified list of VLANs.
<b>Step 8</b>	<b>ip dhcp snooping</b>  <b>Example:</b> Device(config)# ip dhcp snooping	Enables DHCP snooping on the VTEP.
<b>Step 9</b>	<b>end</b>  <b>Example:</b> Device(config)# end	Returns to privileged EXEC mode.

## Configuring DHCP Relay on the Access SVI of a VTEP

Perform this procedure on all the VTEPs for each VLAN that is associated with the Layer 2 VNI configured in the EVPN VXLAN network.

To configure DHCP relay on the access SVI of a VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface vlan <i>vlan-id</i></b>  <b>Example:</b> Device(config)# interface Vlan 201	Enters interface configuration mode for the specified VLAN interface.  This VLAN interface acts as the GiAddr.
<b>Step 4</b>	<b>vrf forwarding <i>vrf-name</i></b>  <b>Example:</b> Device(config-if)# vrf forwarding green	Associates the VRF with the interface.  The interface must be associated with the same VRF for which the Layer 3 VNI has been configured for the EVPN VXLAN network.

	Command or Action	Purpose
<b>Step 5</b>	<b>ip dhcp relay information option vpn-id</b> <b>Example:</b> Device(config-if)# <b>ip dhcp relay information option vpn-id</b>	Enables the device to insert VPN suboptions into the DHCP relay agent information option in the messages forwarded to the DHCP server and sets the GiAddr on the outgoing interface towards the DHCP server.
<b>Step 6</b>	<b>ip dhcp relay source-interface Loopback loopback-interface-id</b> <b>Example:</b> Device(config-if)# <b>ip dhcp relay source-interface Loopback13</b>	Configures the specified Loopback interface as the source interface for DHCP relay messages. The DHCP relay agent uses the IP address of the source interface as the source IP address to relay messages.  <b>Note</b> The IP address configured on the Loopback interface must be unique per VTEP per VRF.
<b>Step 7</b>	<b>ip address ip-address</b> <b>Example:</b> Device(config-if)# <b>ip address 192.168.1.201 255.255.255.0</b>	Sets the IP address for the VLAN interface.
<b>Step 8</b>	<b>ip helper-address ip-address</b> <b>Example:</b> Device(config-if)# <b>ip helper-address 192.168.3.100</b>	Sets the DHCP IP helper address for the VLAN interface.
<b>Step 9</b>	<b>exit</b> <b>Example:</b> Device(config-if)# <b>exit</b>	Exits interface configuration mode and returns to global configuration mode.
<b>Step 10</b>	<b>end</b> <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring the Router Interface on the Border VTEP for DHCP Server Reachability

DHCP server reachability can be achieved through a physical Layer 3 interface or subinterface, or a Layer3 Portchannel interface.

To configure the router interface on the border VTEP for DHCP server reachability, perform the following steps:

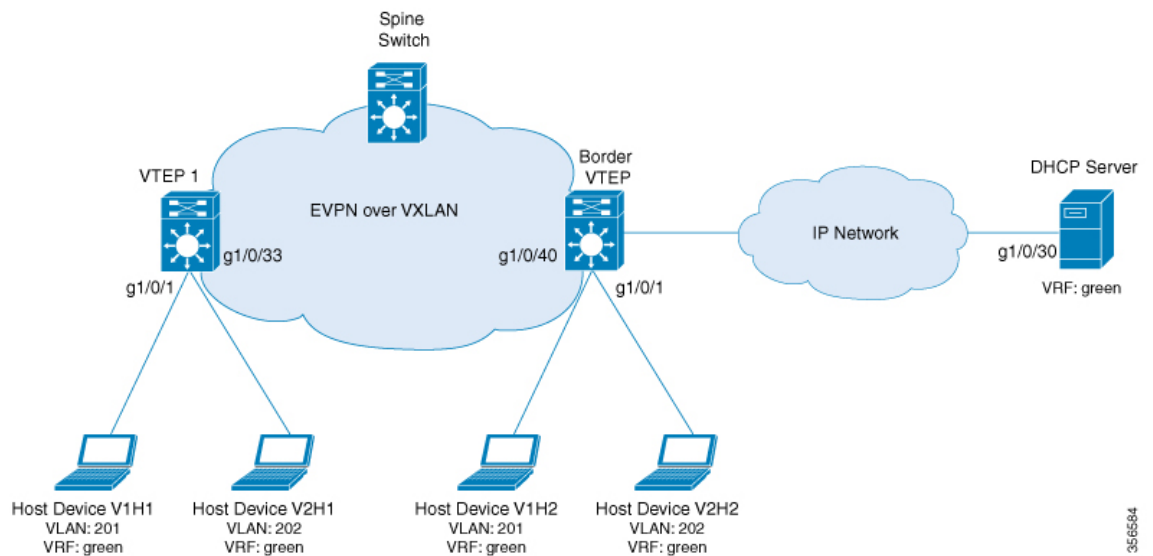
## Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface vlan</b> <i>vlan-id</i> <b>Example:</b> Device (config) # <b>interface vlan 203</b>	Enters interface configuration mode for the specified VLAN interface.
<b>Step 4</b>	<b>vrf forwarding</b> <i>vrf-name</i> <b>Example:</b> Device (config-if) # <b>vrf forwarding green</b>	Configures the SVI for the VLAN and associates the specified VRF with the interface.
<b>Step 5</b>	<b>ip address</b> <i>ip-address</i> <b>Example:</b> Device (config-if) # <b>ip address 192.168.3.203 255.255.255.0</b>	Configures the IP address for the VLAN.
<b>Step 6</b>	<b>ipv6 address</b> <i>ipv6-address</i> <b>Example:</b> Device (config-if) # <b>ipv6 address 2001:203::203/64</b>	Configures the IPv6 address for the VLAN.
<b>Step 7</b>	<b>ipv6 enable</b> <b>Example:</b> Device (config-if) # <b>ipv6 enable</b>	Enables IPv6 processing on the VLAN interface.
<b>Step 8</b>	<b>exit</b> <b>Example:</b> Device (config-if) # <b>exit</b>	Exits interface configuration mode and returns to global configuration mode.
<b>Step 9</b>	<b>interface</b> <i>interface-id</i> <b>Example:</b> Device (config) # <b>interface GigabitEthernet1/0/30</b>	Enters interface configuration mode for the specified interface.
<b>Step 10</b>	<b>switchport access vlan</b> <i>vlan-id</i> <b>Example:</b> Device (config-if) # <b>switchport access vlan 203</b>	Specifies the VLAN to be used as access VLAN when the interface is in access mode.

	Command or Action	Purpose
<b>Step 11</b>	<b>switchport mode access</b>  <b>Example:</b> Device(config-if)# <b>switchport mode access</b>	Configures the interface as an access interface.
<b>Step 12</b>	<b>exit</b>  <b>Example:</b> Device(config-if)# <b>exit</b>	Exits interface configuration mode and returns to global configuration mode.
<b>Step 13</b>	<b>end</b>  <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

## Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric

This section provides an example to show the configuration and verification of DHCP relay deployment in an EVPN VXLAN network. The example uses the following topology where the DHCP client and the DHCP server are in the same tenant VRF:



The illustration shows an EVPN VXLAN network with two VTEPs, VTEP 1 and Border VTEP. Border VTEP is connected to the DHCP server.

DHCP server reachability can be achieved through a physical Layer 3 interface or subinterface, or a Layer3 Portchannel interface. The example shown here deploys DHCP relay using an SVI interface and a switchport.

Table 15: Configuration Example for Deploying DHCP Relay in a BGP EVPN VXLAN Fabric when the DHCP Client and the DHCP Server are in the Same Tenant VRF

VTEP 1	Border VTEP
<pre>VTEP1# show running-config  &lt;snip: only dhcp relevant config is shown&gt; ip dhcp relay information option vpn ip dhcp relay information option ip dhcp compatibility suboption link-selection standard ip dhcp compatibility suboption server-override standard ip dhcp snooping vlan 201-202 ip dhcp snooping ! vlan configuration 200   member vni 5000 vlan configuration 201   member evpn-instance 1 vni 6000 vlan configuration 202   member evpn-instance 2 vni 7000 ! interface Loopback13   vrf forwarding green   ip address 10.1.13.13 255.255.255.0  interface Vlan200   description core svi for l3vni   vrf forwarding green   ip unnumbered Loopback0   ip pim sparse-mode   ipv6 enable   no autostate  interface Vlan201   vrf forwarding green   ip dhcp relay information option vpn-id   ip dhcp relay source-interface Loopback13   ip address 192.168.1.201 255.255.255.0   ip helper-address 192.168.3.100  interface Vlan202   vrf forwarding green   ip dhcp relay information option vpn-id   ip dhcp relay source-interface Loopback13   ip address 192.168.2.201 255.255.255.0   ip helper-address 192.168.3.100  interface nve10   no ip address   source-interface Loopback0   host-reachability protocol bgp   member vni 7000 mcast-group 231.1.1.1   member vni 6000 mcast-group 231.1.1.1   member vni 5000 vrf green</pre>	<pre>Border_VTEP# show running-config  &lt;snip: only dhcp relevant config is shown&gt; ip dhcp relay information option vpn ip dhcp relay information option ip dhcp relay override giaddr link-selection ip dhcp compatibility suboption server-override standard ip dhcp snooping vlan 201-202 ip dhcp snooping ! vlan configuration 200   member vni 5000 vlan configuration 201   member evpn-instance 1 vni 6000 vlan configuration 202   member evpn-instance 2 vni 7000 ! interface Loopback14   vrf forwarding green   ip address 10.1.14.14 255.255.255.0  interface Vlan200   description core svi for l3vni   vrf forwarding green   ip unnumbered Loopback0   ip pim sparse-mode   ipv6 enable   no autostate  interface Vlan201   vrf forwarding green   ip dhcp relay information option vpn-id   ip dhcp relay source-interface Loopback14   ip address 192.168.1.201 255.255.255.0   ip helper-address 192.168.3.100  interface Vlan202   vrf forwarding green   ip dhcp relay information option vpn-id   ip dhcp relay source-interface Loopback14   ip address 192.168.2.201 255.255.255.0   ip helper-address 192.168.3.100  interface nve10   no ip address   source-interface Loopback0   host-reachability protocol bgp   member vni 7000 mcast-group 231.1.1.1   member vni 6000 mcast-group 231.1.1.1   member vni 5000 vrf green</pre>

VTEP 1	Border VTEP
As VTEP 1 is not a border VTEP, DHCP server reachability is not configured on VTEP 1.	<pre>interface Vlan203 vrf forwarding green ip address 192.168.3.203 255.255.255.0 ipv6 address 2001:203::203/64 ipv6 enable end  interface GigabitEthernet1/0/30 description connected to DHCP server switchport access vlan 203 switchport mode access</pre>

The following examples provide sample outputs for the **show ip route vrf** command on VTEP 1 and Border VTEP to verify the reachability of the DHCP server from both VTEPs:

### VTEP 1

The following example shows the output for the **show ip route vrf** command on VTEP 1:

```
VTEP1# show ip route vrf green 192.168.3.100

Routing Table: green
Routing entry for 192.168.3.0/24
  Known via "bgp 10", distance 200, metric 0, type internal
  Last update from 10.2.2.20 on Vlan200, 18:28:43 ago
  Routing Descriptor Blocks:
    * 10.2.2.20 (default), from 10.5.5.50, 18:28:43 ago, via Vlan200
      opaque_ptr 0x7FEEA41D09C8
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: none
      MPLS Flags: NSF
```

### Border VTEP

The following example shows the output for the **show ip route vrf** command on VTEP 2:

```
Border_VTEP# show ip route vrf green 192.168.3.100

Routing Table: green
Routing entry for 192.168.3.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via bgp 10
  Advertised by bgp 10
  Routing Descriptor Blocks:
    * directly connected, via Vlan203
      Route metric is 0, traffic share count is 1
```

### Packet Capture for Spine Switch

The following example shows the packet capture details for the spine switch from the topology configured above:

```

6 12.749326 10.1.13.13 b^F^R 192.168.3.100 DHCP 449 DHCP Discover - Transaction ID
0x228f
7 12.750463 192.168.3.100 b^F^R 10.1.13.13 DHCP 447 DHCP Offer - Transaction ID
0x228f
8 12.755776 10.1.13.13 b^F^R 192.168.3.100 DHCP 467 DHCP Request - Transaction ID
0x228f
9 12.756701 192.168.3.100 b^F^R 10.1.13.13 DHCP 447 DHCP ACK - Transaction ID
0x228f
11 12.803031 00:59:dc:50:ae:42 b^F^R ff:ff:ff:ff:ff:ff ARP 110 Gratuitous ARP for
192.168.2.3 (Reply)
14 15.760480 00:59:dc:50:ae:42 b^F^R ff:ff:ff:ff:ff:ff ARP 110 Who has 192.168.2.201?
Tell 192.168.2.3
15 15.761058 38:0e:4d:9b:6a:42 b^F^R 00:59:dc:50:ae:42 ARP 110 192.168.2.201 is at
38:0e:4d:9b:6a:42

```

### Discover Packet Details for VTEP 1

The following example shows the packet discovery details for VTEP 1 from the topology configured above:

```

Frame 6: 449 bytes on wire (3592 bits), 449 bytes captured (3592 bits) on interface 0
Interface id: 0 (/tmp/epc_ws/wif_to_ts_pipe)
Interface name: /tmp/epc_ws/wif_to_ts_pipe
Encapsulation type: Ethernet (1)
Arrival Time: Mar 28, 2020 09:03:26.742700000 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1585386206.742700000 seconds
[Time delta from previous captured frame: 7.090744000 seconds]
[Time delta from previous displayed frame: 7.090744000 seconds]
[Time since reference or first frame: 12.749326000 seconds]
Frame Number: 6
Frame Length: 449 bytes (3592 bits)
Capture Length: 449 bytes (3592 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:udp:vxlan:eth:ethertype:ip:udp:bootp]
Ethernet II, Src: 00:a3:d1:5a:03:61 (00:a3:d1:5a:03:61), Dst: 38:0e:4d:9b:6a:45
(38:0e:4d:9b:6a:45)
Destination: 38:0e:4d:9b:6a:45 (38:0e:4d:9b:6a:45)
Address: 38:0e:4d:9b:6a:45 (38:0e:4d:9b:6a:45)
.... ..0. .... = LG bit: Globally unique address (factory default)
.... ..0 .... = IG bit: Individual address (unicast)
Source: 00:a3:d1:5a:03:61 (00:a3:d1:5a:03:61)
Address: 00:a3:d1:5a:03:61 (00:a3:d1:5a:03:61)
.... ..0. .... = LG bit: Globally unique address (factory default)
.... ..0 .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.1.1.10, Dst: 10.2.2.20
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 435
Identification: 0xc29c (49820)
Flags: 0x4000, Don't fragment
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment: Set
..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment offset: 0

```



```

Time to live: 253
Protocol: UDP (17)
Header checksum: 0xa27c [validation disabled]
[Header checksum status: Unverified]
Source: 10.1.1.10
Destination: 10.2.2.20
User Datagram Protocol, Src Port: 65294, Dst Port: 4789
Source Port: 65294
Destination Port: 4789
Length: 415
[Checksum: [missing]]
[Checksum Status: Not present]
[Stream index: 0]
Virtual eXtensible Local Area Network
Flags: 0x0800, VXLAN Network ID (VNI)
  0... .. = GBP Extension: Not defined
  .... ..0.. .. = Don't Learn: False
  .... 1... .. = VXLAN Network ID (VNI): True
  .... .. 0... = Policy Applied: False
  .000 .000 0.00 .000 = Reserved(R): 0x0000
Group Policy ID: 0
VXLAN Network Identifier (VNI): 5000
Reserved: 0
Ethernet II, Src: a0:f8:49:10:00:00 (a0:f8:49:10:00:00), Dst: 38:0e:4d:9b:6a:4a
(38:0e:4d:9b:6a:4a)
Destination: 38:0e:4d:9b:6a:4a (38:0e:4d:9b:6a:4a)
Address: 38:0e:4d:9b:6a:4a (38:0e:4d:9b:6a:4a)
  .... ..0. .... = LG bit: Globally unique address (factory default)
  .... ..0. .... = IG bit: Individual address (unicast)
Source: a0:f8:49:10:00:00 (a0:f8:49:10:00:00)
Address: a0:f8:49:10:00:00 (a0:f8:49:10:00:00)
  .... ..0. .... = LG bit: Globally unique address (factory default)
  .... ..0. .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.1.13.13, Dst: 192.168.3.100
0100 .... = Version: 4
... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  0000 00.. = Differentiated Services Codepoint: Default (0)
  ... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 385
Identification: 0x083f (2111)
Flags: 0x0000
  0... .. = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment offset: 0
Time to live: 254
Protocol: UDP (17)
Header checksum: 0xd812 [validation disabled]
[Header checksum status: Unverified]
Source: 10.1.13.13
Destination: 192.168.3.100
User Datagram Protocol, Src Port: 67, Dst Port: 67
Source Port: 67
Destination Port: 67
Length: 365
Checksum: 0x26ca [unverified]
[Checksum Status: Unverified]
[Stream index: 2]
Bootstrap Protocol (Discover)
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6

```

```

Hops: 1
Transaction ID: 0x0000228f
Seconds elapsed: 0
Bootp flags: 0x8000, Broadcast flag (Broadcast)
    1... .... .... .... = Broadcast flag: Broadcast
    .000 0000 0000 0000 = Reserved flags: 0x0000
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 10.1.13.13
Client MAC address: 00:59:dc:50:ae:42 (00:59:dc:50:ae:42)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type (Discover)
    Length: 1
    DHCP: Discover (1)
Option: (57) Maximum DHCP Message Size
    Length: 2
    Maximum DHCP Message Size: 1152
Option: (61) Client identifier
    Length: 27
    Type: 0
    Client Identifier: cisco-0059.dc50.ae42-V1202
Option: (12) Host Name
    Length: 12
    Host Name: host-switch1
Option: (55) Parameter Request List
    Length: 8
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (33) Static Route
    Parameter Request List Item: (150) TFTP Server Address
    Parameter Request List Item: (43) Vendor-Specific Information
Option: (60) Vendor class identifier
    Length: 8
    Vendor class identifier: ciscopnp
Option: (82) Agent Information Option
    Length: 44
    Option 82 Suboption: (1) Agent Circuit ID
        Length: 12
        Agent Circuit ID: 010a000800001b5801010000
    Option 82 Suboption: (2) Agent Remote ID
        Length: 8
        Agent Remote ID: 0006a0f84910bc80
    Option 82 Suboption: (151) VRF name/VPN ID
        Length: 6
        VRF name:
    Option 82 Suboption: (5) Link selection
        Length: 4
        Link selection: 192.168.2.0
    Option 82 Suboption: (11) Server ID Override
        Length: 4
        Server ID Override: 192.168.2.201
Option: (255) End
    Option End: 255

```



## CHAPTER 7

# Configuring Tenant Routed Multicast

- [Restrictions for Tenant Routed Multicast](#) , on page 163
- [Information about Tenant Routed Multicast](#), on page 164
- [How to Configure Tenant Routed Multicast](#), on page 168
- [Configuration Examples for Tenant Routed Multicast](#), on page 173
- [Verifying Tenant Routed Multicast](#), on page 186

## Restrictions for Tenant Routed Multicast

- Only Layer 3 Tenant Routed Multicast (TRM) is supported.
- Layer 2 TRM is not supported.
- TRM works only on the Default Multicast Distribution Tree (MDT).
- In the underlay network, TRM is supported only in the PIM-SM mode.
- In the underlay network, the spine switch should be configured as the Rendezvous Point (RP).
- In the overlay network, TRM is supported in the PIM-SM and PIM-SSM mode.
- Only Layer 3 Tenant Routed Multicast (TRM) is supported.
- Layer 2 TRM is not supported.
- Layer 3 TRM is not supported for IPv6 traffic in the underlay networks.  
In the overlay networks, Layer 3 TRM is supported for both IPv4 and IPv6 traffic.
- TRM works only on the Default Multicast Distribution Tree (MDT).
- In the underlay network, TRM is supported only in the PIM-SM mode. In the overlay network, it is supported in the PIM-SM with Distributed Anycast-RP mode, PIM-SM with External-RP mode, and PIM-SSM mode.
- In the underlay network, the spine switch should be configured as the Rendezvous Point (RP).
- In the overlay network, each of the VTEPs must be configured as the RP for TRM in Distributed Anycast-RP mode.
- For External RP with EVPN AnycastGW, separate register-source loopback is needed in FHR

- E-BGP and I-BGP are supported.
- BGP router-reflector is also supported.

## Information about Tenant Routed Multicast

TRM enables multicast forwarding in a VXLAN fabric that uses a BGP-based EVPN control plane. TRM provides multi-tenancy aware multicast forwarding between senders and receivers within the same or different subnets local or across VTEPs.

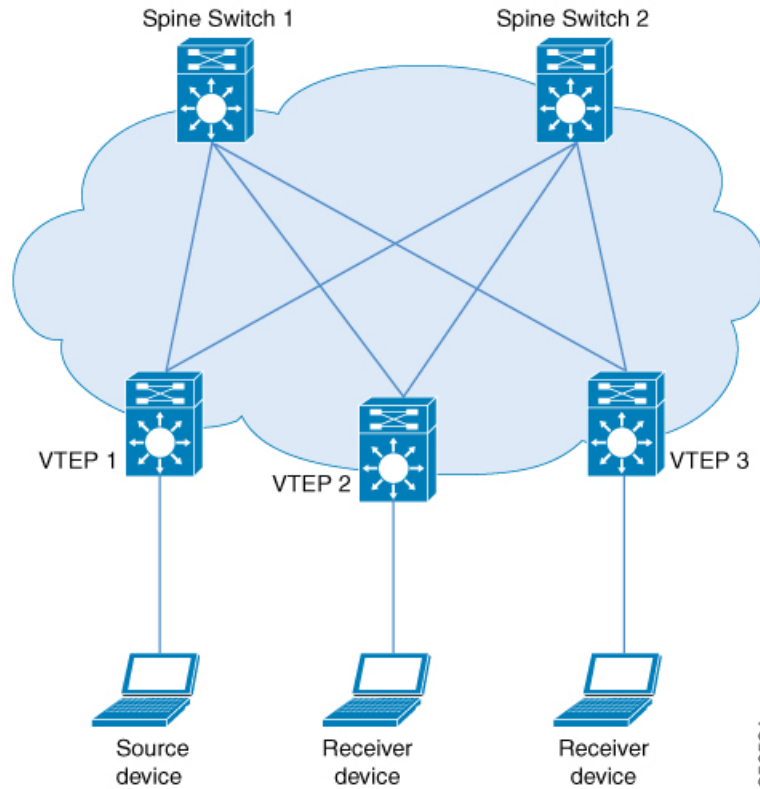
This feature brings the efficiency of multicast delivery to VXLAN overlay networks. TRM enables the delivery of a customer's IP multicast traffic in a multi-tenant fabric in an efficient and resilient manner. The delivery of TRM improves Layer-3 overlay multicast functionality in the networks.

With TRM enabled, multicast forwarding in the underlay is leveraged to replicate VXLAN-encapsulated routed multicast traffic. A default-MDT is built per-VRF. This is an addition to the existing multicast groups for broadcast and unknown unicast traffic in a Layer 2 Virtual Network Instance (VNI), and for Layer 2 multicast replication group. The individual multicast group addresses in the overlay are mapped to the respective underlay multicast address for replication and transport. The advantage of using a BGP-based approach is that it allows the BGP EVPN VXLAN fabric with TRM to operate as fully distributed Overlay Rendezvous-Point (RP), with the RP presence on every edge-device or VTEP.

A multicast-enabled data center fabric is typically part of an overall multicast network. Multicast sources, receivers, and multicast rendezvous points, might reside inside the data center but might also be inside the campus or externally reachable via the WAN. TRM allows a seamless integration with existing multicast networks.

For IPv4 and IPv6 multicast traffic, TRM uses BGP EVPN and MVPN routes to perform multicast routing.

Figure 8: Tenant Routed Multicast Topology



Source detection triggers advertising of EVPN route type 2 in the EVPN fabric. This EVPN route installed in Layer 3 RIB at a receiver VTEP is used as the RPF route towards the source. Thus, if the source is undetected, the RPF for the (S,G) entry is not found. In this case, either RPF remains NULL or a less specific route is installed if present in the RIB.

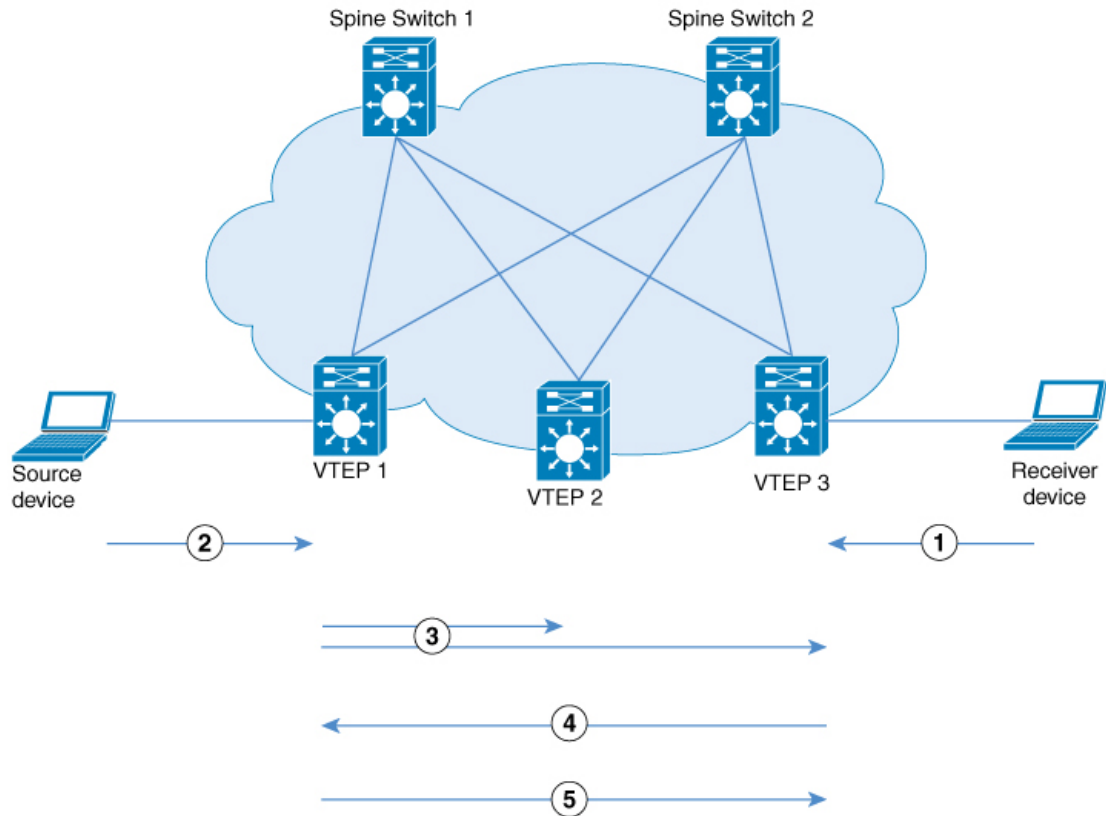
In EVPN-VXLAN network, TRM is supported in the overlay network in PIM-SM and PIM-SSM modes.

## PIM-SM with Distributed Anycast-RP Mode

In PIM-SM with Distributed Anycast-RP Mode, each of the VTEPs must be the RP in the overlay network for their respective groups. The rendezvous points in the underlay network must be configured on the spine switches. All the VTEPs do not need to be BGP peers. There can be BGP peering between the VTEPs and the spine switches with the spine switches acting as route reflectors.

When a VTEP discovers a source device, it sends Source A-D Routes (MVPN Route Type 5) to all the other VTEPs. Based on these Source A-D routes, the other VTEPs send (S,G) join requests as MVPN route type 7 to the source VTEP.

Figure 9: PIM-SM with Distributed Anycast-RP Mode



356532

1. IGMP Join for (\*,G) from Receiver Device to VTEP 3.
2. Data Traffic from Source Device to VTEP 1.
3. Source A-D Route for (S,G) from VTEP 1 to VTEP 2 and VTEP 3.
4. MVPN route type 7 from VTEP 3 to VTEP 1.
5. Data Traffic from VTEP 1 to VTEP 3.

In PIM-SM with Distributed Anycast-RP Mode, the following sequence of events occurs:

1. Receiver sends (\*,G) IGMP Join to VTEP 3. Since VTEP 3 is an RP, (\*,G) is created at VTEP 3.
2. The source device starts streaming data and (S,G) is created on VTEP 1.



**Note** When PIM-SM with Distributed Anycast-RP Mode is enabled, the first packet is dropped.

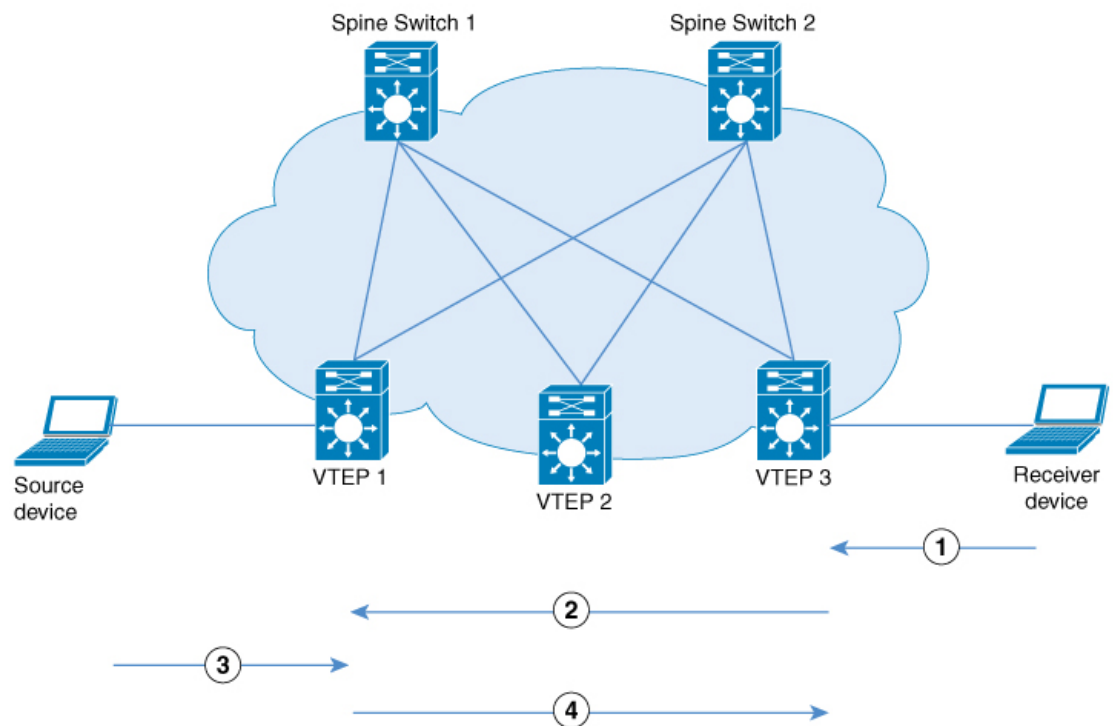
3. VTEP 1 performs self-source-registration since it is also an RP.
4. The source VTEP (VTEP 1) advertises Source A-D Routes (also called MVPN route type 5) for (S,G) to all the other VTEPs which are BGP peers in the MVPN address family.
5. VTEP 2 and VTEP 3 receive and install the Source A-D Routes for (S,G).

6. (S,G) is created at VTEP 3. VTEP 3 now has an overlay route for (S,G) and also has a unicast route to the source device from the EVPN Control plane. It then sends an MVPN route type 7 (S,G) BGP join to VTEP 1 and starts accepting traffic.
7. VTEP 1 receives and installs MVPN route type 7 from VTEP 3. It uses the Layer 3 VNI's SVI as the forwarding interface for (S,G) and starts forwarding traffic.

## PIM-SSM Mode

In PIM-SSM Mode, the Source A-D route (MVPN route type 5) is not needed for the multicast convergence to happen. The receiver VTEP does not wait to receive the Source A-D route to send the MVPN route type 7.

Figure 10: PIM-SSM Mode



1. IGMP Join for (S,G) from Receiver Device to VTEP 3.
2. MVPN route type 7.
3. Data Traffic for (S,G) from Source Device to VTEP 1.
4. Data Traffic from VTEP 1 to VTEP 3.

In PIM-SSM Mode, the following sequence of events occurs:

1. When the source device sends a unicast packet, VTEP 1 sends out EVPN routes to all the other VTEPs, letting them know that the packet is from the source device.
2. The receiver sends an (S,G) IGMP join towards VTEP 3 and an (S,G) entry is created.

3. VTEP 3 performs an RPF lookup for the source device. If the SVI of the Layer 3 VNI is found to be the RPF interface, VTEP 3 sends MVPN route type 7 towards VTEP 1.
4. VTEP 1 receives and installs the MVPN route type 7. VTEP 1 creates an (S,G) entry, using the Layer 3 VNI's SVI as the forwarding interface for (S,G).
5. The source device sends (S,G) data to VTEP 1 which starts forwarding the traffic to VTEP 3.

## How to Configure Tenant Routed Multicast

### Prerequisites to Configuring TRM

Before configuring TRM, ensure that EVPN VXLAN Layer 2 and Layer 3 Overlay networks have been configured. See [How to Configure EVPN VXLAN Integrated Routing and Bridging, on page 59](#) for detailed steps to configure Layer 2 and Layer 3 overlay networks.

Perform the following set of procedures to configure TRM in an EVPN VXLAN network:

## Configuring the Default Multicast Distribution Tree in the VRF

To configure the default MDT for TRM, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>vrf definition</b> <i>vrf-name</i> <b>Example:</b> Device(config)# <b>vrf definition green</b>	Names the VRF and enters VRF configuration mode.
<b>Step 4</b>	<b>mdt default vxlan</b> <i>group-address</i> <b>Example:</b> Device(config-vrf)# <b>mdt default vxlan 225.2.2.2</b>	Configures the multicast group address range for data MDT groups for a VRF in a VXLAN. <ul style="list-style-type: none"> <li>• This command creates a tunnel interface.</li> <li>• By default, the destination address of the tunnel header is the <i>group-address</i> argument.</li> </ul>
<b>Step 5</b>	<b>mdt auto-discovery vxlan</b> <b>Example:</b>	Enables VXLAN with BGP auto-discovery.



	Command or Action	Purpose
	<code>Device(config-vrf) # mdt auto-discovery vxlan</code>	
<b>Step 6</b>	<b>mdt overlay use-bgp spt-only</b> <b>Example:</b> <code>Device(config-vrf) # mdt overlay use-bgp spt-only</code>	Specifies BGP as the overlay protocol.
<b>Step 7</b>	<b>end</b> <b>Example:</b> <code>Device(config-vrf) # end</code>	Returns to privileged EXEC mode.

## Configuring Multicast Routing on the Overlay VRF

To enable multicast routing on the overlay VRF, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> <code>Device&gt; enable</code>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <code>Device# configure terminal</code>	Enters global configuration mode.
<b>Step 3</b>	<b>ip multicast-routing vrf vrf-name</b> <b>Example:</b> <code>Device(config) # ip multicast-routing vrf green</code>	Enables IP multicast forwarding on the overlay VRF.
<b>Step 4</b>	<b>end</b> <b>Example:</b> <code>Device(config) # end</code>	Returns to privileged EXEC mode.

## Configuring Multicast on Switch Virtual Interfaces for the Core-facing and Access-facing VLANs

To configure multicast on SVIs for the core-facing and access-facing VLANs on the VTEP, perform the following steps:

**Procedure**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface vlan</b> <i>core-facing-vlan-id</i> <b>Example:</b> Device(config)# <b>interface vlan 200</b>	Enters interface configuration mode for the specified VLAN.
<b>Step 4</b>	<b>ip pim sparse-mode</b> <b>Example:</b> Device(config-if) # <b>ip pim sparse-mode</b>	Enables multicast on the core-facing SVI.
<b>Step 5</b>	<b>exit</b> <b>Example:</b> Device(config-if) # <b>end</b>	Returns to privileged EXEC mode.
<b>Step 6</b>	<b>interface vlan</b> <i>access-facing-vlan-id</i> <b>Example:</b> Device(config)# <b>interface vlan 202</b>	Enters interface configuration mode for the specified VLAN.
<b>Step 7</b>	<b>ip pim sparse-mode</b> <b>Example:</b> Device(config-if) # <b>ip pim sparse-mode</b>	Enables multicast on the access-facing SVI where sources or receivers are connected.  Repeat this step for all the access-facing SVIs that are part of the Layer 2 VNI where sources and receivers are connected.
<b>Step 8</b>	<b>end</b> <b>Example:</b> Device(config-if) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring BGP with MVPN Address Family on VTEP

To configure BGP on a VTEP with MVPN address family, perform the following steps:

**Procedure**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b>	Enables privileged EXEC mode.

	Command or Action	Purpose
	<b>Example:</b> Device> <b>enable</b>	Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>router bgp <i>autonomous-system-number</i></b> <b>Example:</b> Device(config)# <b>router bgp 1</b>	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
<b>Step 4</b>	<b>address-family ipv4 mvpn</b> <b>Example:</b> Device(config-router)# <b>address-family ipv4 mvpn</b>	Specifies the MVPN address family and enters address family configuration mode.
<b>Step 5</b>	<b>neighbor <i>ip-address</i> activate</b> <b>Example:</b> Device(config-router-af) # <b>neighbor 10.2.2.20 activate</b>	Enables the exchange of information with a BGP neighbor.  Use the IP address of the spine switch as the neighbor IP address.
<b>Step 6</b>	<b>neighbor <i>ip-address</i> send-community extended</b> <b>Example:</b> Device(config-router-af) # <b>neighbor 10.2.2.20 send-community both</b>	Specifies the communities attribute sent to a BGP neighbor.  Use the IP address of the spine switch as the neighbor IP address.
<b>Step 7</b>	<b>neighbor <i>ip-address</i> advertisement-interval <i>seconds</i></b> <b>Example:</b> Device(config-router-af) # <b>neighbor 10.2.2.20 advertisement-interval 10</b>	(Optional) Sets the minimum route advertisement interval (RAI) between the sending of BGP routing updates.
<b>Step 8</b>	<b>exit-address-family</b> <b>Example:</b> Device(config-router-af) # <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 9</b>	<b>end</b> <b>Example:</b> Device(config-router) # <b>end</b>	Returns to privileged EXEC mode.

## Configuring RP for Underlay Network

To configure RP for the underlay network, perform the following steps:

**Procedure**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>ip pim rp-address ip-address-of-rp</b> <b>Example:</b> Device (config)# <b>ip pim rp-address</b> <b>&lt;rp-ip-address&gt;</b>	Configures the RP in the underlay network. For information about RP redundancy, see to <i>IP Multicast Routing Configuration Guide</i> .
<b>Step 4</b>	<b>end</b> <b>Example:</b> Device (config)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring RP and SSM for Overlay Network

To configure RP and SSM for the overlay network, perform the following steps:

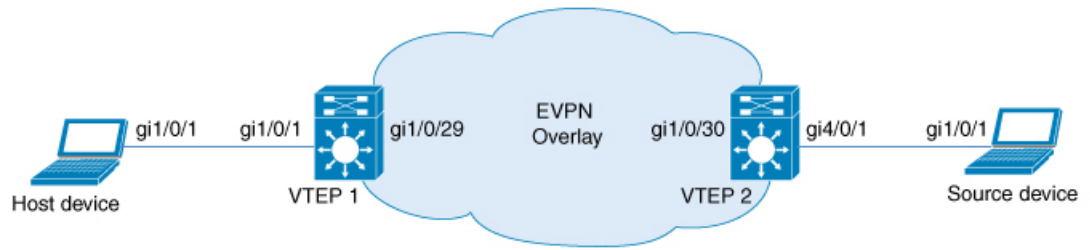
**Procedure**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface loopback-interface</b> <b>Example:</b> Device (config)# <b>interface Loopback 13</b>	Enters interface configuration mode for the specified Loopback interface.
<b>Step 4</b>	<b>vrf forwarding vrf-name</b> <b>Example:</b> Device (config-if)# <b>vrf forwarding green</b>	Configures forwarding table for the Loopback interface.
<b>Step 5</b>	<b>ip-address ip-address subnet-mask</b> <b>Example:</b>	Configures the IP address for the Loopback interface.

	Command or Action	Purpose
	Device(config-if)# <b>ip address 10.1.13.13</b> <b>255.255.255.255</b>	
<b>Step 6</b>	<b>ip pim sparse-mode</b>  <b>Example:</b> Device(config-if)# <b>ip pim sparse-mode</b>	Enables PIM sparse mode on the Loopback interface.  <b>Note</b> Enable PIM sparse mode only if EVPN VXLAN Layer 2 overlay network is also configured on the VTEP with underlay multicast as the mechanism for forwarding BUM traffic.
<b>Step 7</b>	<b>exit</b>  <b>Example:</b> Device(config-if)# <b>exit</b>	Returns to global configuration mode.
<b>Step 8</b>	<b>ip pim [vrf vrf-name ] ssm {default   range access-list }</b>  <b>Example:</b> Device(config)# <b>ip pim vrf green ssm default</b>	Configures an SSM range for TRM.  The <b>default</b> keyword defines the SSM range access list as 232/8.  The <b>range</b> keyword specifies the standard IP access list number or name that defines the SSM range.
<b>Step 9</b>	<b>ip pim vrf vrf-name rp-address</b> <i>loopback-address-of-vtep</i>  <b>Example:</b> Device(config)# <b>ip pim vrf green rp-address 10.1.13.13</b>	Configures the address of the loopback interface of the local VTEP as the PIM RP for the multicast group in PIM-SM with Distributed Anycast RP mode.  <b>Note</b> The loopback interface specified must be part of the same VRF.
<b>Step 10</b>	<b>end</b>  <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

## Configuration Examples for Tenant Routed Multicast

This section provides an example for TRM configuration. The following example shows a sample configuration for a VXLAN network with a receiver device and a source device connected to VTEP 1 and VTEP 2 respectively, with TRM enabled.



356534

*Table 16: Configuration Example for a VXLAN Network with the Source Device and Host Device connected to two VTEPs with TRM Enabled*

VTEP 1	VTEP 2
--------	--------

VTEP 1	VTEP 2
<pre> VTEP1# show running-config ! hostname VTEP1 ! vrf definition green  rd 103:2  !  address-family ipv4   mdt auto-discovery vxlan   mdt default vxlan 239.1.1.1   mdt overlay use-bgp spt-only   route-target export 103:2   route-target import 104:2   route-target export 103:2 stitching   route-target import 104:2 stitching  exit-address-family  !  ! ip routing ip multicast-routing ip multicast-routing vrf green ! ! l2vpn evpn  replication-type static  router-id Loopback0  default-gateway advertise  ! l2vpn evpn instance 1 vlan-based  encapsulation vxlan  route-target export 103:1  route-target import 104:1  ! l2vpn evpn instance 2 vlan-based  encapsulation vxlan  !  ! system mtu 9150 ! ! vlan configuration 200  member vni 5000 vlan configuration 201  member evpn-instance 1 vni 6000 vlan configuration 202  member evpn-instance 2 vni 7000 ! ! interface Loopback0  ip address 10.1.1.10 255.255.255.255  ip pim sparse-mode ! interface Loopback13  vrf forwarding green  ip address 10.1.13.13 255.255.255.0  ip pim sparse-mode  ipv6 enable ! ! interface GigabitEthernet1/0/1  description access interface  switchport mode trunk ! </pre>	<pre> VTEP2# show running-config ! hostname VTEP2 ! vrf definition green  rd 104:2  !  address-family ipv4   mdt auto-discovery vxlan   mdt default vxlan 239.1.1.1   mdt overlay use-bgp spt-only   route-target export 104:2   route-target import 103:2   route-target export 104:2 stitching   route-target import 103:2 stitching  exit-address-family  !  ! ip routing ip multicast-routing ip multicast-routing vrf green ! ! l2vpn evpn  replication-type static  router-id Loopback0  default-gateway advertise  ! l2vpn evpn instance 1 vlan-based  encapsulation vxlan  route-target export 104:1  route-target import 103:1  ! l2vpn evpn instance 2 vlan-based  encapsulation vxlan  !  ! system mtu 9150 ! ! vlan configuration 200  member vni 5000 vlan configuration 201  member evpn-instance 1 vni 6000 vlan configuration 202  member evpn-instance 2 vni 7000 ! ! interface Loopback0  ip address 10.2.2.20 255.255.255.255  ip pim sparse-mode ! interface Loopback14  vrf forwarding green  ip address 10.1.14.14 255.255.255.0  ip pim sparse-mode  ipv6 address 2001:200::14:14/128  ipv6 enable ! ! interface GigabitEthernet4/0/1  description access interface  switchport mode trunk ! </pre>



VTEP 1	VTEP 2
<pre> interface GigabitEthernet1/0/29   description core-underlay-interface   no switchport   ip address 172.16.1.29 255.255.255.0   ip pim sparse-mode ! interface Vlan200   description core svi for l3vni   vrf forwarding green   ip unnumbered Loopback0   ip pim sparse-mode   ipv6 enable   no autostate ! interface Vlan201   description vni 6000 default-gateway   vrf forwarding green   ip address 192.168.1.201 255.255.255.0   ip pim sparse-mode   ipv6 enable ! interface Vlan202   description vni 7000 default-gateway   vrf forwarding green   ip address 192.168.2.202 255.255.255.0   ip pim sparse-mode ! ! interface nve10   no ip address   source-interface Loopback0   host-reachability protocol bgp   member vni 6000 mcast-group 231.1.1.1   member vni 7000 mcast-group 231.1.1.1   member vni 5000 vrf green ! router ospf 1   router-id 10.1.1.10   network 10.1.1.0 0.0.0.255 area 0   network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10   bgp router-id interface Loopback0   bgp log-neighbor-changes   bgp update-delay 1   no bgp default ipv4-unicast   neighbor 10.2.2.20 remote-as 10   neighbor 10.2.2.20 update-source Loopback0 !   address-family ipv4   exit-address-family !   address-family ipv4 mvpn   neighbor 10.2.2.20 activate   neighbor 10.2.2.20 send-community both   exit-address-family !   address-family l2vpn evpn   neighbor 10.2.2.20 activate   neighbor 10.2.2.20 send-community both   exit-address-family ! </pre>	<pre> interface GigabitEthernet1/0/30   description core-underlay-interface   no switchport   ip address 172.16.1.30 255.255.255.0   ip pim sparse-mode ! interface Vlan200   description core svi for l3vni   vrf forwarding green   ip unnumbered Loopback0   ip pim sparse-mode   ipv6 enable   no autostate ! interface Vlan201   vrf forwarding green   ip address 192.168.1.201 255.255.255.0   ip pim sparse-mode   ipv6 address 2001:DB8:201::201/64   ipv6 enable ! interface Vlan202   description vni 7000 default-gateway   vrf forwarding green   ip address 192.168.2.202 255.255.255.0   ip pim sparse-mode ! ! interface nve10   no ip address   source-interface Loopback0   host-reachability protocol bgp   member vni 6000 mcast-group 231.1.1.1   member vni 7000 mcast-group 231.1.1.1   member vni 5000 vrf green ! router ospf 1   router-id 10.2.2.20   network 10.2.2.0 0.0.0.255 area 0   network 172.16.1.0 0.0.0.255 area 0 ! router bgp 10   bgp router-id interface Loopback0   bgp log-neighbor-changes   bgp update-delay 1   no bgp default ipv4-unicast   neighbor 10.1.1.10 remote-as 10   neighbor 10.1.1.10 update-source Loopback0 !   address-family ipv4   exit-address-family !   address-family ipv4 mvpn   neighbor 10.1.1.10 activate   neighbor 10.1.1.10 send-community both   exit-address-family !   address-family l2vpn evpn   neighbor 10.1.1.10 activate   neighbor 10.1.1.10 send-community both   exit-address-family ! </pre>

VTEP 1	VTEP 2
<pre> address-family ipv4 vrf green   advertise l2vpn evpn   redistribute connected   redistribute static exit-address-family ! ! ip pim rp-address 10.1.1.10 ip pim vrf green rp-address 10.1.13.13 ! ! end </pre>	<pre> address-family ipv4 vrf green   advertise l2vpn evpn   redistribute connected   redistribute static exit-address-family ! ! ip pim rp-address 10.1.1.10 ip pim vrf green rp-address 10.1.14.14 ! ! end </pre>

The following examples provide outputs for **show** commands on VTEP 1 and VTEP 2 in the topology configured above.

- [show nve peers, on page 178](#)
- [show l2vpn evpn peers vxlan, on page 179](#)
- [show ip igmp vrf green groups, on page 179](#)
- [show bgp ipv4 mvpn all, on page 179](#)
- [show ip mroute vrf green, on page 180](#)
- [show ip mfib vrf green, on page 181](#)
- [show ip mroute, on page 183](#)
- [show ip mfib, on page 184](#)

### show nve peers

#### VTEP 1

The following example shows the output for the **show nve peers** command on VTEP 1:

```

VTEP1# show nve peers
Interface  VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve10     5000     L3CP 10.2.2.20        380e.4d9b.6a4a 5000      UP   A/-/4 03:22:40
nve10     6000     L2CP 10.2.2.20          14           6000      UP   N/A   03:22:19
nve10     7000     L2CP 10.2.2.20          6            7000      UP   N/A   03:22:19

```

#### VTEP 2

The following example shows the output for the **show nve peers** command on VTEP 2:

```

VTEP2# show nve peers
Interface  VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve10     5000     L3CP 10.1.1.10          a0f8.4910.bce2 5000      UP   A/M/4 03:22:27
nve10     6000     L2CP 10.1.1.10          6            6000      UP   N/A   03:22:27
nve10     7000     L2CP 10.1.1.10          4            7000      UP   N/A   03:22:27

```

**show l2vpn evpn peers vxlan****VTEP 1**

The following example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 1:

```
VTEP1# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI    UP time
-----
nve10      6000            10.2.2.20       5          6000   01:34:50
nve10      7000            10.2.2.20       6          7000   01:34:50
```

**VTEP 2**

The following example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 2:

```
VTEP2# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes
eVNI      UP time
-----
nve10      6000            10.1.1.10       7          6000   01:35:23
nve10      7000            10.1.1.10       6          7000   01:35:23
```

**show ip igmp vrf green groups****VTEP 1**

The following example shows the output for the **show ip igmp vrf green groups** command on VTEP 1:

```
VTEP1# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address  Interface          Uptime    Expires    Last Reporter  Group Accounted
229.1.1.1     Vlan201           04:08:35  00:02:16  192.168.1.81
224.0.1.40    Loopback13        06:35:55  00:02:05  10.1.13.13
```

**VTEP 2**

The following example shows the output for the **show ip igmp vrf green groups** command on VTEP 2:

```
VTEP2# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address  Interface          Uptime    Expires    Last Reporter  Group Accounted
224.0.1.40    Loopback14        05:11:42  00:02:18  10.1.14.14
```

**show bgp ipv4 mvpn all****VTEP 1**

The following example shows the output for the **show bgp ipv4 mvpn all** command on VTEP 1:

```
VTEP1# show bgp ipv4 mvpn all
BGP table version is 22, local router ID is 10.1.1.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 103:2 (default for vrf green)
*>i  [5][103:2][192.168.2.88][229.1.1.1]/18
      10.2.2.20                0    100    0 ?
Route Distinguisher: 104:2
*>i  [5][104:2][192.168.2.88][229.1.1.1]/18
      10.2.2.20                0    100    0 ?
Route Distinguisher: 10.2.2.20:2
*>  [7][10.2.2.20:2][10][192.168.2.88/32][229.1.1.1/32]/22
      0.0.0.0                    32768 ?
```

## VTEP 2

The following example shows the output for the **show bgp ipv4 mvpn all** command on VTEP 2:

```
VTEP2# show bgp ipv4 mvpn all
BGP table version is 24, local router ID is 10.2.2.20
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 104:2 (default for vrf green)
*>  [5][104:2][192.168.2.88][229.1.1.1]/18
      0.0.0.0                    32768 ?
*>i  [7][104:2][10][192.168.2.88/32][229.1.1.1/32]/22
      10.1.1.10                0    100    0 ?
Route Distinguisher: 10.2.2.20:2
*>i  [7][10.2.2.20:2][10][192.168.2.88/32][229.1.1.1/32]/22
      10.1.1.10                0    100    0 ?
```

## show ip mroute vrf green

## VTEP 1

The following example shows the output for the **show ip mroute vrf green** command on VTEP 1:

```
VTEP1# show ip mroute vrf green IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
```

```

* - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 229.1.1.1), 04:11:11/stopped, RP 10.1.13.13, flags: SJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan201, Forward/Sparse, 04:11:11/00:02:40
(192.168.2.88, 229.1.1.1), 00:02:42/00:00:17, flags: gQ
  Incoming interface: Vlan200, RPF nbr 10.2.2.20
  Outgoing interface list:
    Vlan201, Forward/Sparse, 00:02:42/00:02:40
(*, 224.0.1.40), 04:44:21/00:02:34, RP 10.1.13.13, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback13, Forward/Sparse, 04:44:21/00:02:34

```

## VTEP 2

The following example shows the output for the **show ip mroute vrf green** command on VTEP 2:

```

VTEP2# show ip mroute vrf green
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 229.1.1.1), 00:53:58/stopped, RP 10.1.14.14, flags: SPF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null
(192.168.2.88, 229.1.1.1), 00:53:58/00:01:56, flags: FTGqx
  Incoming interface: Vlan202, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan200, Forward/Sparse, 00:03:06/stopped
(*, 224.0.1.40), 04:46:21/00:02:48, RP 10.1.14.14, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback14, Forward/Sparse, 04:46:21/00:02:48

```

## show ip mrib vrf green

## VTEP 1

The following example shows the output for the **show ip mrib vrf green** command on VTEP 1:

```

VTEP1# show ip mrib vrf green
Entry Flags:      C - Directly Connected, S - Signal, IA - Inherit A flag,
                  ET - Data Rate Exceeds Threshold, K - Keepalive

```

```

DDE - Data Driven Event, HW - Hardware Installed
ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
MS - MoFRR Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
NS - Negate Signalling, SP - Signal Present,
A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
MA - MFIB Accept, A2 - Accept backup,
RA2 - MRIB Accept backup, MA2 - MFIB Accept backup
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:  HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
VRF green
(*,224.0.0.0/4) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,224.0.1.40) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel9 Flags: A
  Loopback13 Flags: F IC NS
  Pkts: 0/0/0 Rate: 0 pps
(*,229.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel9 Flags: A
  Vlan201 Flags: F NS
  Pkts: 0/0/0 Rate: 0 pps
(192.168.2.88,229.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 117/0/126/0, Other: 0/0/0
  Tunnel9 Flags: A
  Vlan200, VXLAN Decap Flags: NS
  Vlan201 Flags: F NS
  Pkts: 0/0/0 Rate: 0 pps

```

## VTEP 2

The following example shows the output for the **show ip mfib vrf green** command on VTEP 2:

```

VTEP2# show ip mfib vrf green
Entry Flags: C - Directly Connected, S - Signal, IA - Inherit A flag,
ET - Data Rate Exceeds Threshold, K - Keepalive
DDE - Data Driven Event, HW - Hardware Installed
ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
MS - MoFRR Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
NS - Negate Signalling, SP - Signal Present,
A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
MA - MFIB Accept, A2 - Accept backup,
RA2 - MRIB Accept backup, MA2 - MFIB Accept backup
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:  HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
VRF green
(*,224.0.0.0/4) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,224.0.1.40) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

Tunnel5 Flags: A
Loopback14 Flags: F IC NS
  Pkts: 0/0/0 Rate: 0 pps
(*,229.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnel5 Flags: A
(192.168.2.88,229.1.1.1) Flags: HW
SW Forwarding: 56/0/100/0, Other: 1715/1699/16
HW Forwarding: 2306/0/122/0, Other: 0/0/0
Vlan202 Flags: A
Vlan200, VXLAN v4 Encap (5000, 239.1.1.1) Flags: F
  Pkts: 0/0/0 Rate: 0 pps

```

## show ip mroute

### VTEP 1

The following example shows the output for the **show ip mroute** command on VTEP 1:

```

VTEP1# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 239.1.1.1), 00:57:25/00:03:16, RP 10.1.1.10, flags: SJCx
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/29, Forward/Sparse, 00:57:17/00:03:16
    Tunnel0, Forward/Sparse, 00:57:25/stopped
(10.2.2.20, 239.1.1.1), 00:04:25/00:02:37, flags: Tx
  Incoming interface: GigabitEthernet1/0/29, RPF nbr 172.16.1.30
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:04:25/00:01:33
(*, 231.1.1.1), 00:57:25/00:03:02, RP 10.1.1.10, flags: SJCfX
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/29, Forward/Sparse, 00:56:28/00:03:02
    Tunnel0, Forward/Sparse, 00:57:25/stopped
(10.2.2.20, 231.1.1.1), 00:56:26/00:02:55, flags: JTx
  Incoming interface: GigabitEthernet1/0/29, RPF nbr 172.16.1.30
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:56:26/00:00:33
(10.1.1.10, 231.1.1.1), 00:57:23/00:03:03, flags: FTx
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/29, Forward/Sparse, 00:56:53/00:03:02
(*, 224.0.1.40), 00:57:25/00:02:46, RP 10.1.1.10, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0

```

```

Outgoing interface list:
  GigabitEthernet1/0/29, Forward/Sparse, 00:56:43/00:02:46
  Loopback0, Forward/Sparse, 00:57:25/00:02:44

```

## VTEP 2

The following example shows the output for the **show ip mroute** command on VTEP 2:

```

VTEP2# show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 239.1.1.1), 04:50:56/stopped, RP 10.1.1.10, flags: SJCFx
  Incoming interface: GigabitEthernet1/0/30, RPF nbr 172.16.1.29
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 04:50:56/00:02:03
    (10.2.2.20, 239.1.1.1), 00:04:51/00:02:44, flags: FTx
      Incoming interface: Loopback0, RPF nbr 0.0.0.0
      Outgoing interface list:
        GigabitEthernet1/0/30, Forward/Sparse, 00:04:49/00:02:37
        (*, 231.1.1.1), 00:58:51/stopped, RP 10.1.1.10, flags: SJCFx
          Incoming interface: GigabitEthernet1/0/30, RPF nbr 172.16.1.29
          Outgoing interface list:
            Tunnel0, Forward/Sparse, 00:58:51/00:01:08
            (10.1.1.10, 231.1.1.1), 00:58:16/00:01:05, flags: JTx
              Incoming interface: GigabitEthernet1/0/30, RPF nbr 172.16.1.29
              Outgoing interface list:
                Tunnel0, Forward/Sparse, 00:58:16/00:01:43
                (10.2.2.20, 231.1.1.1), 00:58:49/00:02:58, flags: FTx
                  Incoming interface: Loopback0, RPF nbr 0.0.0.0
                  Outgoing interface list:
                    GigabitEthernet1/0/30, Forward/Sparse, 00:58:49/00:02:46
                    (*, 224.0.1.40), 05:14:59/00:02:03, RP 10.1.1.10, flags: SJCL
                      Incoming interface: GigabitEthernet1/0/30, RPF nbr 172.16.1.29
                      Outgoing interface list:
                        Loopback0, Forward/Sparse, 05:14:58/00:02:03

```

## show ip mfib

## VTEP 1

The following example shows the output for the **show ip mfib** command on VTEP 1:

```

VTEP1# show ip mfib
Entry Flags:      C - Directly Connected, S - Signal, IA - Inherit A flag,
                  ET - Data Rate Exceeds Threshold, K - Keepalive

```



```

DDE - Data Driven Event, HW - Hardware Installed
ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
MS - MoFRR Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
NS - Negate Signalling, SP - Signal Present,
A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
MA - MFIB Accept, A2 - Accept backup,
RA2 - MRIB Accept backup, MA2 - MFIB Accept backup
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:  HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
Default
(*,224.0.0.0/4) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,224.0.1.40) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnel8 Flags: A
  GigabitEthernet1/0/29 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  Loopback0 Flags: F IC NS
    Pkts: 0/0/0 Rate: 0 pps
(*,231.1.1.1) Flags: C HW
  SW Forwarding: 1/0/206/0, Other: 4/4/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnel8 Flags: A
  GigabitEthernet1/0/29 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(10.1.1.10,231.1.1.1) Flags: HW
  SW Forwarding: 1/0/128/0, Other: 0/0/0
  HW Forwarding:  192/0/144/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/29 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(10.2.2.20,231.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  386/0/186/0, Other: 0/0/0
  GigabitEthernet1/0/29 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 26/0/150/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnel8 Flags: A
  GigabitEthernet1/0/29 Flags: F NS
    Pkts: 0/0/22 Rate: 0 pps
(10.2.2.20,239.1.1.1) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding:  162/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/29 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1 Rate: 0 pps

```

## VTEP 2

The following example shows the output for the **show ip mfib** command on VTEP 2:

```

VTEP2# show ip mfib
Entry Flags:      C - Directly Connected, S - Signal, IA - Inherit A flag,
                  ET - Data Rate Exceeds Threshold, K - Keepalive

```

```

DDE - Data Driven Event, HW - Hardware Installed
ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
MS - MoFRR Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
NS - Negate Signalling, SP - Signal Present,
A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
MA - MFIB Accept, A2 - Accept backup,
RA2 - MRIB Accept backup, MA2 - MFIB Accept backup
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:  HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
Default
(*,224.0.0.0/4) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,224.0.1.40) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  GigabitEthernet1/0/30 Flags: A NS
  Loopback0 Flags: F IC NS
  Pkts: 0/0/0 Rate: 0 pps
(*,231.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  1/0/146/0, Other: 0/0/0
  GigabitEthernet1/0/30 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0 Rate: 0 pps
(10.1.1.10,231.1.1.1) Flags: HW
  SW Forwarding: 1/0/128/0, Other: 0/0/0
  HW Forwarding:  192/0/156/0, Other: 0/0/0
  GigabitEthernet1/0/30 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1 Rate: 0 pps
(10.2.2.20,231.1.1.1) Flags: HW
  SW Forwarding: 3/0/194/0, Other: 1/1/0
  HW Forwarding:  397/0/174/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/30 Flags: F NS
  Pkts: 0/0/2 Rate: 0 pps
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  GigabitEthernet1/0/30 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0 Rate: 0 pps
(10.2.2.20,239.1.1.1) Flags: HW
  SW Forwarding: 3/0/150/0, Other: 1/1/0
  HW Forwarding:  160/0/156/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/30 Flags: F NS
  Pkts: 0/0/2 Rate: 0 pps

```

## Verifying Tenant Routed Multicast

The following table lists the **show** commands that are used to verify TRM:

Command	Purpose
<b>show nve peers</b>	Displays NVE interface state information for peer leaf switches.
<b>show l2vpn evpn peers vxlan</b>	Displays Layer 2 EVPN peer route counts in the VXLAN and up time.
<b>show ip igmp vrf green groups</b>	Displays the multicast groups with receivers that are directly connected to the router pertaining to the specific Multicast Virtual Routing and Forwarding (MVRF) instance and that were learned through IGMP.
<b>show bgp ipv4 mvpn all</b>	Displays the MVPN options for BGP MVPN C-route signaling.
<b>show ip mroute vrf green</b>	Displays the contents of the mroute table that pertain to a specific MVRF instance.
<b>show ip mfib vrf green</b>	Displays forwarding entries and interfaces in the IPv4 Multicast Forwarding Information Base (MFIB) associated with MVRF instances.
<b>show ip mroute</b>	Displays multicast routing table information.
<b>show ip mfib</b>	Displays the forwarding entries and interfaces in the IPv4 MFIB.





## CHAPTER 8

# Configuring EVPN VXLAN External Connectivity

- [Restrictions for EVPN VXLAN External Connectivity, on page 189](#)
- [Information About EVPN VXLAN External Connectivity, on page 189](#)
- [How to Configure EVPN VXLAN External Connectivity, on page 193](#)
- [Configuration Examples for EVPN VXLAN External Connectivity, on page 204](#)

## Restrictions for EVPN VXLAN External Connectivity

- External connectivity with VPLS networks is supported only when bridging is the mode of interworking between the two domains. Integrated routing and bridging (IRB) is not supported between a BGP EVPN VXLAN fabric and a VPLS network.
- External Connectivity with Layer 3 networks is supported only for IPv4 and IPv6 unicast traffic.
- External connectivity with an MVPN network is not supported for multicast traffic.
- Import of EVPN IP routes, which includes both route type 5 and route type 2 host routes, to global routing table is not supported.

## Information About EVPN VXLAN External Connectivity

External connectivity allows the movement of Layer 2 and Layer 3 traffic between an EVPN VXLAN network and an external network. It also enables the EVPN VXLAN network to exchange routes with the externally connected network. Routes within an EVPN VXLAN network are already shared between all the VTEPs or leaf switches. External connectivity uses the VTEPs on the periphery of the network to pass on these routes to an external Layer 2 or Layer 3 network. Similarly, the EVPN VXLAN network imports the reachability routes from the external network. External connectivity extends the Layer 2 or Layer 3 overlay network outside the VXLAN network. The process of extending a Layer 2 or Layer 3 network outside the EVPN VXLAN network is also known as handoff.

## Implementation of Border Nodes for EVPN VXLAN External Connectivity

Border nodes or border VTEPs are the devices through which you establish a connection between an EVPN VXLAN network and an external network. The border nodes sit on the periphery of the EVPN VXLAN

network and remain a part of the BGP EVPN VXLAN fabric. To enable external connectivity, you can implement the border nodes of an EVPN VXLAN network as either border leaf or border spine switches.

### Connectivity Through a Border Leaf Switch

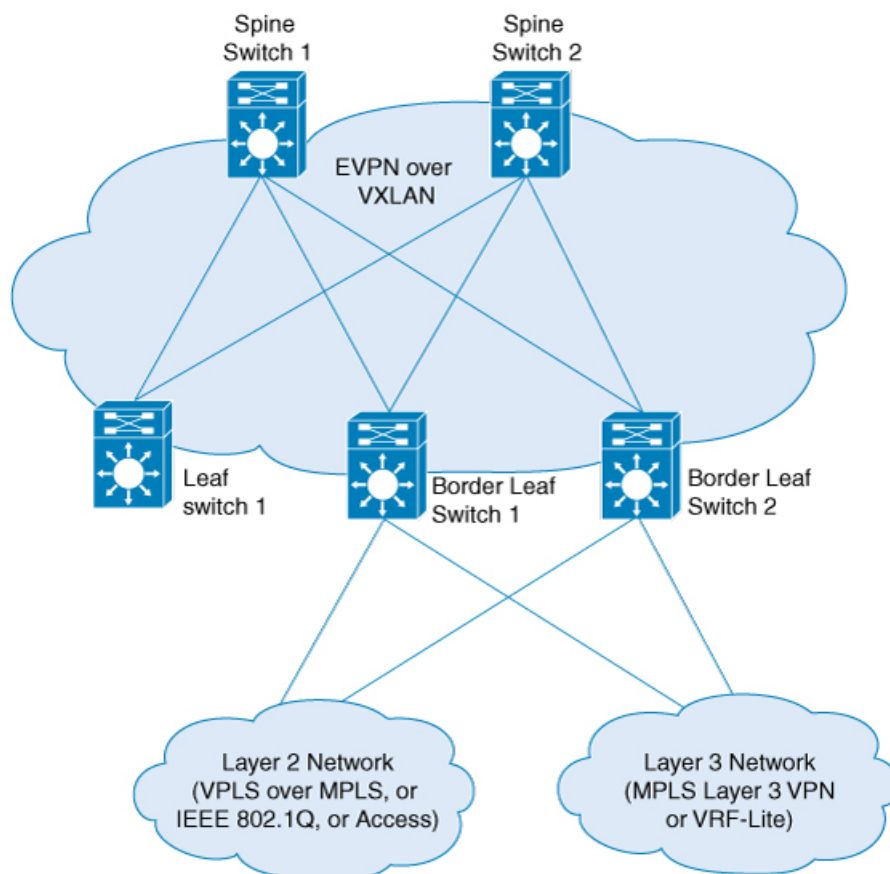
Leaf switches deployed as border nodes support the required control plane and data plane functionalities. Border leaf deployment ensures that the configuration on the spine switches is much simpler. Border leaf switches only allow communication between the external network and the VXLAN network, also known as north-south communication.



**Note** A border leaf switch can also be multiple switches functioning as a single logical system with Cisco StackWise Virtual configured.

The following figure shows border leaf external connectivity of an EVPN VXLAN network with external Layer 2 and Layer 3 networks.:

**Figure 11: EVPN VXLAN External Connectivity Through a Border Leaf Switch**



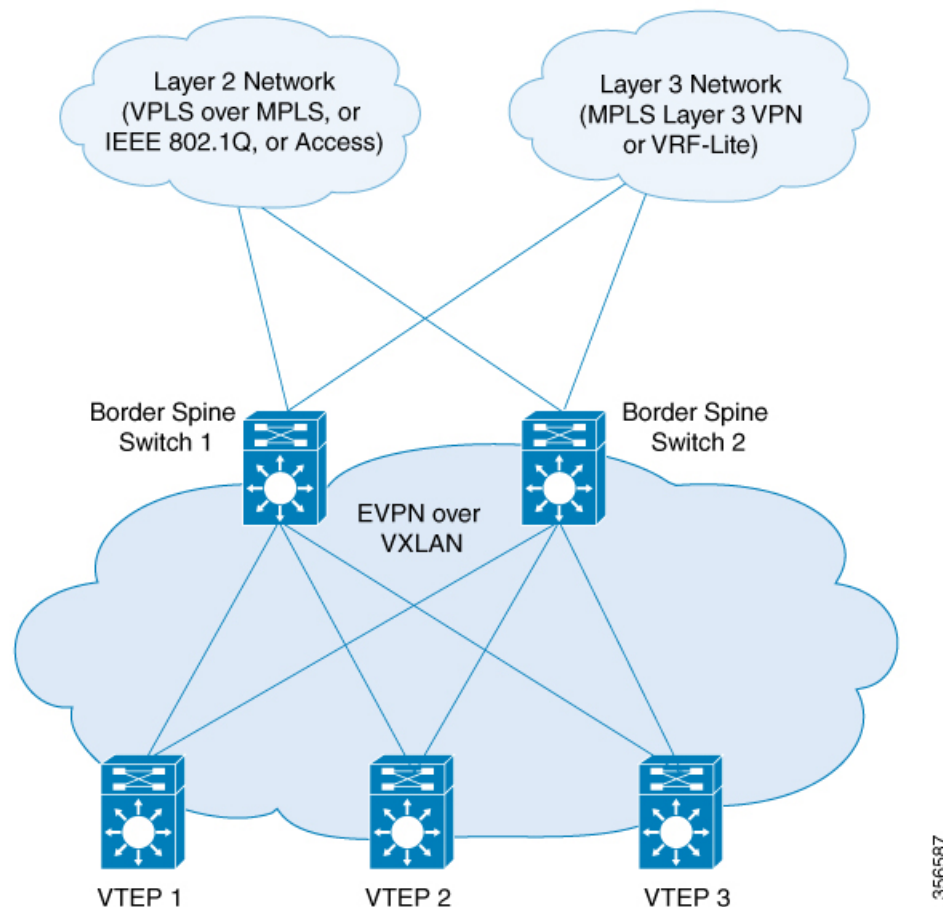
356586

### Connectivity Through a Border Spine Switch

Deploying spine switches as border nodes provides the advantage of optimizing the north-south communication with external resources. At the same time, border spine deployment allows the spine switches to support VXLAN control and data plane functionality. Border spine switches allow both north-south communication and east-west communication. East-west communication represents the communication within the nodes of the EVPN VXLAN network.

The following figure shows border spine external connectivity of an EVPN VXLAN network with external Layer 2 and Layer 3 networks.:

**Figure 12: EVPN VXLAN External Connectivity Through a Border Spine Switch**



## External Connectivity with Layer 3 Networks

Layer 3 external connectivity or handoff is established by connecting the border nodes of a BGP EVPN VXLAN fabric with an edge router from the external Layer 3 network. The border node acts as a VTEP to perform VXLAN encapsulation and decapsulation, but it also routes the traffic towards the edge routing device. The VXLAN-facing interface on the external Layer 3 network can be a switch virtual interface (SVI), or a Layer 3 interface, or a Layer 3 subinterface.

You can use Layer 3 external connectivity to achieve any of the following:

- Extend the logical isolation between VRFs or VLANs within the EVPN VXLAN network into the externally routed network. The external routed network can be a traditional non-VXLAN campus network, a datacenter, or a WAN.
- Provide shared access within the EVPN VXLAN network to a common external service such as the internet.

BGP EVPN VXLAN fabric supports Layer 3 external connectivity with VRF-Lite and MPLS Layer 3 VPN networks.

### Layer 3 External Connectivity with VRF-Lite

Using VRF allows for the use of multiple routing tables that are independent and isolated. VRF-Lite is a mechanism to extend the tenant Layer 3 VRF information beyond the BGP EVPN VXLAN Fabric. External connectivity with VRF-Lite or VRF handoff involves a two-box approach where the border node and the edge router are physically independent devices. With VRF-Lite handoff, the BGP EVPN VXLAN fabric extends the connectivity for different tenants externally on a hop-by-hop basis.

Once the border node learns external routes from the edge router, it advertises the prefixes inside the BGP EVPN VXLAN fabric as EVPN type 5 routes. This information is distributed to all the other VTEPs in the network. The border node also advertises EVPN routes to the external edge router. It sends the EVPN routes learned from the Layer 2 VPN EVPN address family to the IPv4 or IPv6 unicast address family.

### Layer 3 Multicast External Connectivity with MPLS Layer 3 VPN

Layer 3 external connectivity with an MPLS Layer 3 VPN network or MPLS handoff uses a single-box approach. The single-box approach combines the functionalities of an EVPN VXLAN border node and an MPLS PE router into a single physical device. The device is also known as a border PE node. The border PE node reoriginates IP prefixes from the EVPN address family of the BGP EVPN VXLAN fabric to the VPNv4 address family of the MPLS network. Likewise, the border PE node performs the corresponding function in the reverse direction. eBGP peering is necessary between the border PE node and the MPLS PE devices to ensure the connectivity.

MPLS handoff allows scalability for EVPN VXLAN networks that have a large number of tenants or VRFs. Scalability is not possible with VRF-Lite handoff.

In every VRF on a border VTEP, there are two sets of manually configured import and export route targets. The first set of import and export route targets is associated with the BGP neighbor in the BGP EVPN VXLAN fabric. This BGP neighbor uses the EVPN address family to exchange Layer 3 information. The second set of import and export route targets is associated with the BGP neighbor in the Layer 3 VPN network. This BGP neighbor uses either VPNv4 or VPNv6 unicast address families to exchange Layer 3 information. The separation of route targets allows you to configure both sets of route targets independently. In this way, a border VTEP in an EVPN VXLAN network effectively stitches the two sets of route targets. The route targets associated with the BGP neighbor in the Layer 3 VPN network are known as normal route targets. The route targets associated with the BGP neighbor in the BGP EVPN VXLAN fabric are known as stitching route targets.

## External Connectivity with Layer 2 Networks

Layer 2 external connectivity or handoff for an EVPN VXLAN network extends the Layer 2 domain outside of the network. BGP EVPN VXLAN fabric supports Layer 2 external connectivity with IEE 802.1Q, access, and VPLS over MPLS networks.



### Layer 2 External connectivity with IEEE 802.1Q or Access Networks

Layer 2 handoff to IEEE 802.1Q networks is achieved through a regular IEEE 802.1Q Trunk port configuration on the Switchport interfaces on the border nodes. You can also connect EVPN VXLAN networks to external access networks.

The commonly deployed scenario has EVPN enabled at the distribution layer and has the access layer switches connected with IEEE 802.1Q Trunk encapsulation. The IEEE 802.1Q Layer 2 traffic that comes from the access layer switches is mapped to the corresponding VLAN. The border node then bridges the traffic towards the destination with VXLAN encapsulation. The inner packet does not carry the IEEE 802.1Q tag. Instead, the VXLAN network identifier (VNI), which is the Layer 2 VNI in the VXLAN header, represents the broadcast domain. Similarly, the border nodes decapsulate the traffic from the BGP EVPN VXLAN fabric and bridge it with the corresponding IEEE 802.1Q tag to the access switches. The interface on the border VTEP that faces the external interface can be either an access or a Trunk port. The external interface can belong to either a Layer 2 switch or a firewall.



---

**Note** If you connect the network to an external Layer 2 switch through two border VTEPs, it represents a dual connection. In such cases, STP does not propagate over the BGP EVPN VXLAN fabric by default.

---

### Layer 2 External connectivity with VPLS over MPLS Network

External connectivity with VPLS networks or VPLS handoff is achieved when a border VTEP or multiple border VTEPs establish a connection with the VPLS network. The border nodes act as the provider edge (PE) devices in the VPLS network and as VTEPs in the EVPN VXLAN network.

BGP EVPN VXLAN supports VPLS handoff in the form of VPLS stitching through either an access VFI or an access pseudowire on the VLAN on the border VTEP.

The access pseudowires and the pseudowires in the access VFI function as the access ports in the EVPN VXLAN network. The BGP EVPN VXLAN fabric treats the MAC addresses learned on the pseudowires as locally learned MAC addresses. It advertises these MAC addresses within the fabric as EVPN type 2 routes. The pseudowires are in a different split horizon group compared to the EVPN VXLAN network. Therefore, BUM traffic floods between both the EVPN VXLAN and VPLS networks.

## How to Configure EVPN VXLAN External Connectivity

This section provides information about how to configure external connectivity between an EVPN VXLAN network and an external Layer 2 or Layer 3 network.



---

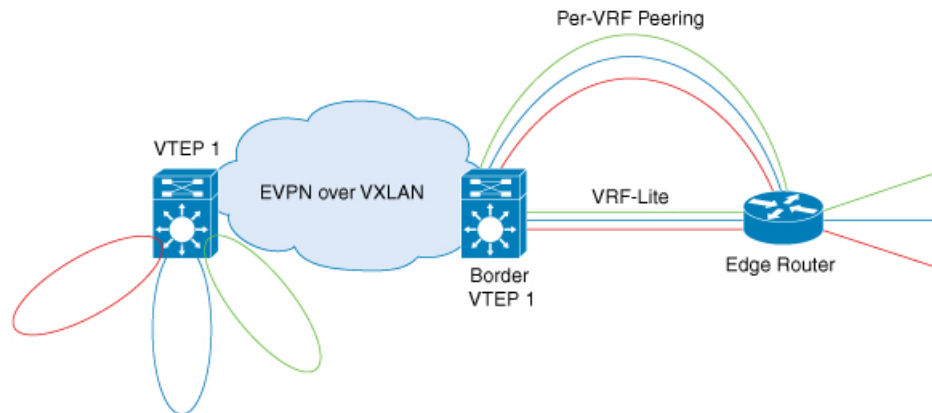
**Note** You must configure EVPN VXLAN Layer 2 and Layer 3 overlay networks before you configure external connectivity. See [How to Configure EVPN VXLAN Integrated Routing and Bridging, on page 59](#) for detailed steps.

---

## Enabling Layer 3 External Connectivity with VRF-Lite

The following figure shows a sample topology that illustrates Layer 3 external connectivity with VRF-Lite:

Figure 13: Layer 3 External Connectivity with VRF-Lite



To configure Layer 3 external connectivity with VRF-Lite, perform the following set of procedures:

- Configure the VRF on the border VTEP interface that faces the external router.
- Ensure that Layer 2 VPN EVPN is advertised as part of the BGP VRF configuration. See [Configuring BGP with EVPN and VRF Address Families on a VTEP](#), on page 64 for detailed steps.



**Note** Redistribution of the respective interior gateway protocol (IGP) is required in the BGP VRF address family to distribute the external prefixes into the BGP EVPN VXLAN fabric.

For more information about VRF-Lite, see *Contents* → *IP Routing Configuration Guide* → *Configuring VRF-lite* in the software configuration guide for the applicable release.

## Configuring the VRF on the Border VTEP Interface that Faces the External Router

To configure the VRF on the border VTEP interface that faces the external router, perform these steps:

### Procedure

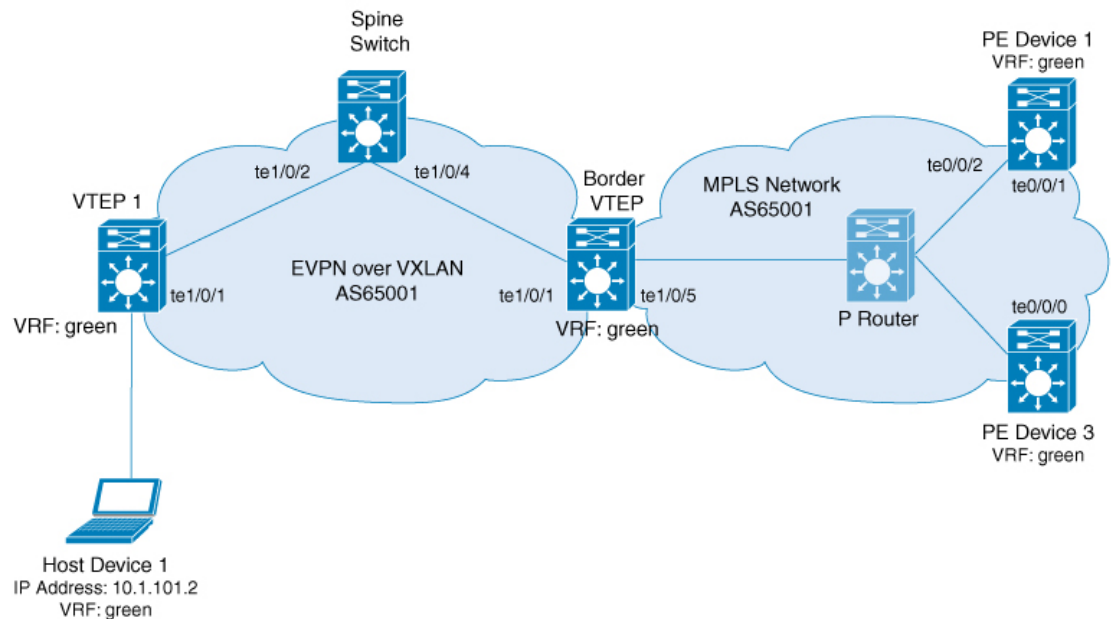
	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>interface-id</i> <b>Example:</b>	Enters the interface configuration mode for the specified interface.

	Command or Action	Purpose
	Device(config)# <b>interface</b> <b>GigabitEthernet1/0/30</b>	
<b>Step 4</b>	<b>vrf forwarding</b> <i>vrf-name</i>  <b>Example:</b> Device(config-if)# <b>vrf forwarding green</b>	Associates the VRF with the interface.  <b>Note</b> The interface must be associated with the same VRF for which the Layer 3 VNI has been configured for the EVPN VXLAN network.
<b>Step 5</b>	<b>ip address</b> <i>ip-address</i>  <b>Example:</b> Device(config-if)# <b>ip address</b> <b>192.168.3.203 255.255.255.0</b>	Configures the IP address for the interface.
<b>Step 6</b>	<b>end</b>  <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN

The following figure shows a sample topology that illustrates Layer 3 external connectivity with an MPLS Layer 3 VPN network:

Figure 14: Layer 3 External Connectivity with MPLS Layer 3 VPN



To enable EVPN VLAN Layer 3 external connectivity with MPLS Layer 3 VPN networks, perform the following set of procedures:

- Run the **mpls label mode all-vrfs protocol all-afs per-vrf** command in global configuration mode on the border VTEP.
- Configure BGP with reorigination of routes with a new route type for Layer 2 VPN, VPNv4, VPNv6 address families on the border VTEP.

## Configuring BGP on a Border VTEP for External Connectivity with MPLS Layer 3 VPN

To configure BGP on a border VTEP to establish external connectivity with an MPLS Layer 3 VPN network, perform this procedure:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>router bgp <i>autonomous-system-number</i></b> <b>Example:</b> Device (config)# <b>router bgp 1</b>	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
<b>Step 4</b>	<b>bgp log-neighbor-changes</b> <b>Example:</b> Device (config-router)# <b>bgp log-neighbor-changes</b>	(Optional) Enables the generation of logging messages when the status of a BGP neighbor changes. For more information, see Configuring BGP section of the <i>IP Routing Configuration Guide</i> .
<b>Step 5</b>	<b>bgp update-delay <i>time-period</i></b> <b>Example:</b> Device (config-router)# <b>bgp update-delay 1</b>	(Optional) Sets the maximum initial delay period before sending the first update. For more information, see Configuring BGP section of the <i>IP Routing Configuration Guide</i> .
<b>Step 6</b>	<b>bgp graceful-restart</b> <b>Example:</b> Device (config-router)# <b>bgp graceful-restart</b>	(Optional) Enables the BGP graceful restart capability for all BGP neighbors. For more information, see Configuring BGP section of the <i>IP Routing Configuration Guide</i> .
<b>Step 7</b>	<b>no bgp default ipv4-unicast</b> <b>Example:</b> Device (config-router)# <b>no bgp default ipv4-unicast</b>	(Optional) Disables default IPv4 unicast address family for BGP peering session establishment. For more information, see Configuring BGP section of the <i>IP Routing Configuration Guide</i> .

	Command or Action	Purpose
Step 8	<b>neighbor</b> <i>spine-ip-address</i> <b>remote-as</b> <i>number</i> <b>Example:</b> <pre>Device(config-router)# neighbor 172.16.255.1 remote-as 1</pre>	<p>Defines multiprotocol-BGP neighbors in the EVPN network.</p> <p>Use the IP address of the spine switch as the neighbor IP address. This configures the spine switch as a BGP neighbor.</p>
Step 9	<b>neighbor</b> <i>mpls-peer-ip-address</i> <b>remote-as</b> <i>number</i> <b>Example:</b> <pre>Device(config-router)# neighbor 172.16.255.103 remote-as 1</pre>	<p>Defines multiprotocol-BGP neighbors in the external MPLS network.</p> <p>Use the IP address of the external MPLS network peer as the neighbor IP address. This configures the external MPLS network peer as a BGP neighbor.</p>
Step 10	<b>neighbor</b> { <i>ip-address</i>   <i>group-name</i> } <b>update-source</b> <i>interface</i> <b>Example:</b> <pre>Device(config-router)# neighbor 172.16.255.1 update-source Loopback0</pre>	<p>Configures update source. Update source can be configured per neighbor or per peer-group.</p> <p>Use the IP address of the spine switch as the neighbor IP address.</p>
Step 11	<b>address-family</b> <i>l2vpn evpn</i> <b>Example:</b> <pre>Device(config-router)# address-family l2vpn evpn</pre>	<p>Specifies the L2VPN address family and enters address family configuration mode.</p>
Step 12	<b>import</b> <i>vpn4</i> <b>unicast re-originate</b> <b>Example:</b> <pre>Device(config-router-af)# import vpn4 unicast re-originate</pre>	<p>Reoriginates the VPNv4 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.</p>
Step 13	<b>import</b> <i>vpn6</i> <b>unicast re-originate</b> <b>Example:</b> <pre>Device(config-router-af)# import vpn6 unicast re-originate</pre>	<p>Reoriginates the VPNv6 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.</p>
Step 14	<b>neighbor</b> <i>ip-address</i> <b>activate</b> <b>Example:</b> <pre>Device(config-router-af)# neighbor 10.11.11.11 activate</pre>	<p>Enables the exchange information from a BGP neighbor.</p> <p>Use the IP address of the spine switch as the neighbor IP address.</p>
Step 15	<b>neighbor</b> <i>ip-address</i> <b>send-community</b> [ <b>both</b>   <b>extended</b>   <b>standard</b> ] <b>Example:</b> <pre>Device(config-router-af)# neighbor 10.11.11.11 send-community both</pre>	<p>Specifies the communities attribute sent to a BGP neighbor.</p> <p>Use the IP address of the spine switch as the neighbor IP address.</p>

	Command or Action	Purpose
		<p><b>Note</b> Use either of <b>extended</b> or <b>both</b> keywords. External connectivity cannot be established when you use the <b>standard</b> keyword.</p>
<b>Step 16</b>	<p><b>neighbor {ip-address   peer-group-name} next-hop-self [ all]</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# neighbor ip-address next-hop-self all</pre>	<p>Configures the router as the next hop for a BGP-speaking neighbor or peer group.</p> <p>The <b>all</b> keyword is mandatory when implementing external connectivity through iBGP, where the EVPN fabric and the MPLS network are in the same BGP autonomous system number.</p> <p>The <b>all</b> keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers.</p>
<b>Step 17</b>	<p><b>exit-address-family</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# exit-address-family</pre>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 18</b>	<p><b>address-family vpnv4</b></p> <p><b>Example:</b></p> <pre>Device(config-router)# address-family vpnv4</pre>	Specifies the VPNv4 address family and enters address family configuration mode.
<b>Step 19</b>	<p><b>import l2vpn evpn re-originate</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# import l2vpn evpn re-originate</pre>	Reoriginates the EVPN routes imported from the EVPN fabric into the VPNv4 address family as VPNv4 routes and distributes them to the external network.
<b>Step 20</b>	<p><b>neighbor ip-address activate</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# neighbor 172.16.255.103 activate</pre>	<p>Enables the exchange information from a BGP neighbor.</p> <p>Use the IP address of the external MPLS network router as the neighbor IP address.</p>
<b>Step 21</b>	<p><b>neighbor ip-address send-community [both   extended   standard]</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# neighbor 172.16.255.103 send-community both</pre>	<p>Specifies the communities attribute sent to a BGP neighbor.</p> <p>Use the IP address of the external MPLS network router as the neighbor IP address.</p> <p><b>Note</b> Use either of <b>extended</b> or <b>both</b> keywords. External connectivity cannot be established when you use the <b>standard</b> keyword.</p>

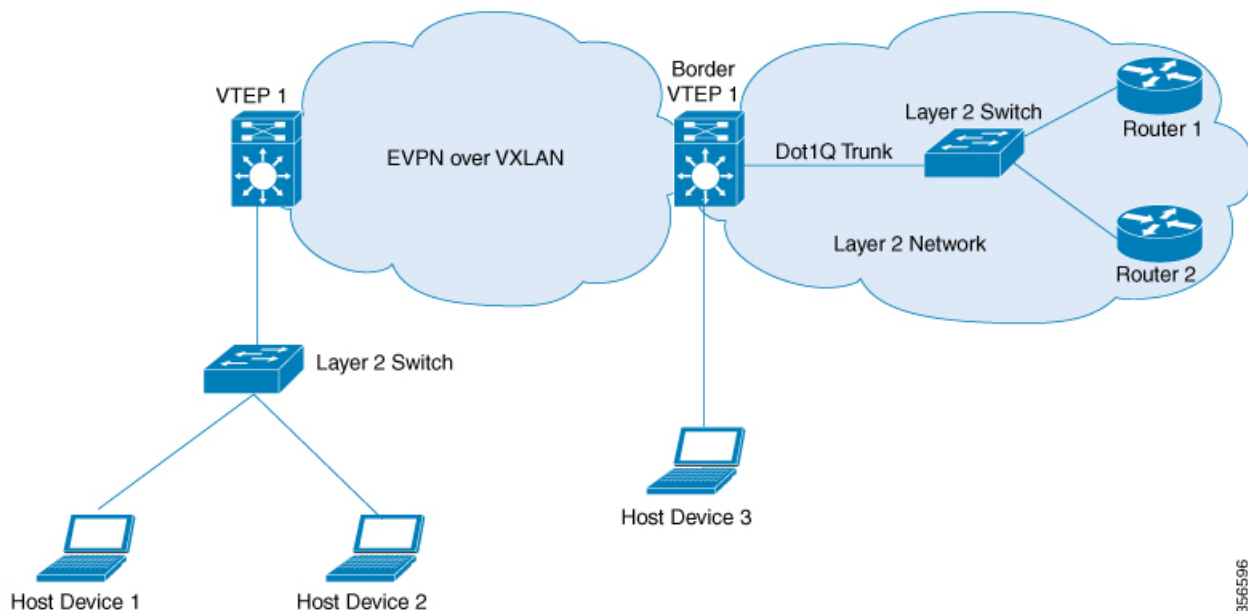
	Command or Action	Purpose
Step 22	<p><b>neighbor {ip-address   peer-group-name} next-hop-self [ all]</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# neighbor ip-address next-hop-self all</pre>	<p>Configures the router as the next hop for a BGP-speaking neighbor or peer group.</p> <p>The <b>all</b> keyword is mandatory when implementing external connectivity through iBGP, where the EVPN fabric and the MPLS network are in the same BGP autonomous system number.</p> <p>The <b>all</b> keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers.</p>
Step 23	<p><b>exit-address-family</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# exit-address-family</pre>	Exits address family configuration mode and returns to router configuration mode.
Step 24	<p><b>address-family vpnv6</b></p> <p><b>Example:</b></p> <pre>Device(config-router)# address-family vpnv6</pre>	Specifies the VPNv6 address family and enters address family configuration mode.
Step 25	<p><b>import l2vpn evpn re-originate</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# import l2vpn evpn re-originate</pre>	Reoriginates the EVPN routes imported from the EVPN fabric into the VPNv6 address family as VPNv6 routes and distributes them to the external network.
Step 26	<p><b>neighbor ip-address activate</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# neighbor 172.16.255.103 activate</pre>	<p>Enables the exchange information from a BGP neighbor.</p> <p>Use the IP address of the spine switch as the neighbor IP address.</p>
Step 27	<p><b>neighbor ip-address send-community [both   extended   standard]</b></p> <p><b>Example:</b></p> <pre>Device(config-router-af)# neighbor 172.16.255.103 send-community both</pre>	<p>Specifies the communities attribute sent to a BGP neighbor.</p> <p>Use the IP address of the spine switch as the neighbor IP address.</p> <p><b>Note</b> Use either of <b>extended</b> or <b>both</b> keywords. External connectivity cannot be established when you use the <b>standard</b> keyword.</p>
Step 28	<p><b>neighbor {ip-address   peer-group-name} next-hop-self [ all]</b></p> <p><b>Example:</b></p>	<p>Configures the router as the next hop for a BGP-speaking neighbor or peer group.</p> <p>The <b>all</b> keyword is mandatory when implementing external connectivity through</p>

	Command or Action	Purpose
	<pre>Device (config-router-af) # neighbor ip-address next-hop-self all</pre>	<p>iBGP, where the EVPN fabric and the MPLS network are in the same BGP autonomous system number.</p> <p>The <b>all</b> keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers.</p>
<b>Step 29</b>	<p><b>exit-address-family</b></p> <p><b>Example:</b></p> <pre>Device (config-router-af) # exit-address-family</pre>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 30</b>	<p><b>end</b></p> <p><b>Example:</b></p> <pre>Device (config-router) # end</pre>	Returns to privileged EXEC mode.

## Enabling Layer 2 External Connectivity with IEEE 802.1Q Networks

The following image shows a sample topology that illustrates Layer 2 external connectivity with an IEEE 802.1Q network:

Figure 15: Layer 2 External Connectivity with IEEE 802.1Q Networks



356596



You can also connect the EVPN VXLAN network to a firewall in place of the Layer 2 switch in the above image. To configure Layer 2 external connectivity with an IEEE 802.1Q network, perform the following steps on the external Layer 2 switch:

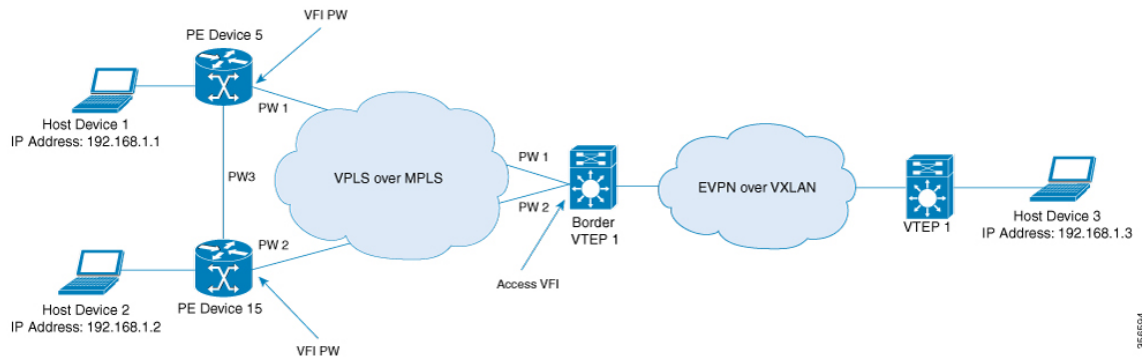
### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. Enter your password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface interface-id</b> <b>Example:</b> Device(config)# <b>interface</b> <b>GigabitEthernet4/0/1</b>	Enters interface configuration mode for the specified interface.  The specified interface must be the interface on the Layer 2 switch through which the EVPN VXLAN network communicates with the IEEE 802.1Q network.
<b>Step 4</b>	<b>switchport mode trunk</b> <b>Example:</b> Device(config-if)# <b>switchport mode trunk</b>	Configures the interface as a trunking VLAN Layer 2 interface.
<b>Step 5</b>	<b>switchport trunk allowed vlan vlan-list</b> <b>Example:</b> Device(config-if)# <b>switchport trunk</b> <b>allowed vlan 201,202</b>	Sets the list of VLANs that are allowed to transmit traffic from this interface in tagged format when the interface is in trunking mode.
<b>Step 6</b>	<b>end</b> <b>Example:</b> Device(config-if)# <b>end</b>	Returns to privileged EXEC mode.

## Enabling Layer 2 External Connectivity with a VPLS Network Through an Access VFI

The following illustration shows a single-homed VXLAN network connected to a VPLS over MPLS network through the access VFIs on the border VTEP:

Figure 16: Layer 2 External Connectivity with a VPLS Network Through an Access VFI

**Note**

We recommend you to use Cisco Catalyst 9500 Series - High Performance switches or Cisco Catalyst 9600 Series switches as border VTEPs when you configure Layer 2 external connectivity with a VPLS network.

We recommend you to configure Cisco Stackwise Virtual on the border VTEPs in order to achieve physical redundancy when you configure Layer 2 external connectivity with a VPLS network.

Perform the following set of procedures to enable Layer 2 external connectivity with VPLS networks through an access VFI interface:

1. Define the access VFI for the VTEPs.
2. Configure the access VFI as a member of the VLAN on the VTEPs.
3. Configure the EVPN instance as a member of the VLAN on the VTEPs.
4. Configure VPLS on the border VTEP.

## Defining an Access VFI on a Border VTEP

To configure an access facing VFI on the VLAN of a border VTEP, perform the following steps:

For more information on configuring VFIs, in the software configuration guide for the required release, go to *Contents* → *Multiprotocol Label Switching (MPLS) Configuration Guide* → *Configuring Virtual Private LAN Service (VPLS) and VPLS BGP-Based Autodiscovery*.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enters privileged EXEC mode. Enter password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 3</b>	<b>l2vpn vfi context</b> <i>vfi-name</i> <b>Example:</b> Device(config) # <b>l2vpn vfi context myVFI</b>	Establishes an Layer 2 VPN VFI between two or more separate networks, and enters VFI configuration mode.
<b>Step 4</b>	<b>vpn id</b> <i>vpn-id</i> <b>Example:</b> Device(config-vfi) # <b>vpn id 1</b>	Configures the VPN ID for the VFI.
<b>Step 5</b>	<b>member ip-address encapsulation mpls</b> <b>Example:</b> Device(config-vfi) # <b>member 10.12.12.5 encapsulation mpls</b>	Specifies the device that forms a point-to-point Layer 2 VPN VFI connection.
<b>Step 6</b>	Repeat step 5 for all devices that form a point-to-point Layer 2 VPN VFI connection.	
<b>Step 7</b>	<b>end</b> <b>Example:</b> Device(config-vfi) # <b>end</b>	Exits VFI configuration mode and enters privileged EXEC mode.

## Adding an Access VFI and an EVPN Instance as Members of the VLAN of a Border VTEP

To add an access VFI and an EVPN instance as members of the VLAN of a border VTEP, perform the following steps:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enters privileged EXEC mode. Enter password, if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>vlan configuration</b> <i>vlan-number</i> <b>Example:</b> Device(config) # <b>vlan configuration 11</b>	Enters VLAN feature configuration mode for the specified VLAN interface. Enter the VLAN number that is associated with the Layer 2 VNI configured in the EVPN VXLAN network.
<b>Step 4</b>	<b>member access-vfi</b> <i>vfi-name</i> <b>Example:</b>	Adds the access VFI as a member of the VLAN configuration.

	Command or Action	Purpose
	Device(config-vlan)# <b>member access-vfi myVFI</b>	
<b>Step 5</b>	<b>member evpn-instance</b> <i>evpn-instance-number</i> <b>vni</b> <i>l2-vni-number</i>  <b>Example:</b> Device(config-vlan)# <b>member evpn-instance 1 vni 6000</b>	Adds the EVPN instance as a member of the VLAN configuration.
<b>Step 6</b>	<b>end</b>  <b>Example:</b> Device(config-vlan)# <b>end</b>	Exits VLAN configuration mode and enters privileged EXEC mode.

## Configuring VPLS on a Border VTEP

To configure VPLS on a border VTEP, in the software configuration guide for the required release, see *Contents → Multiprotocol Label Switching (MPLS) Configuration Guide → Configuring Virtual Private LAN Service (VPLS) and VPLS BGP-Based Autodiscovery*.

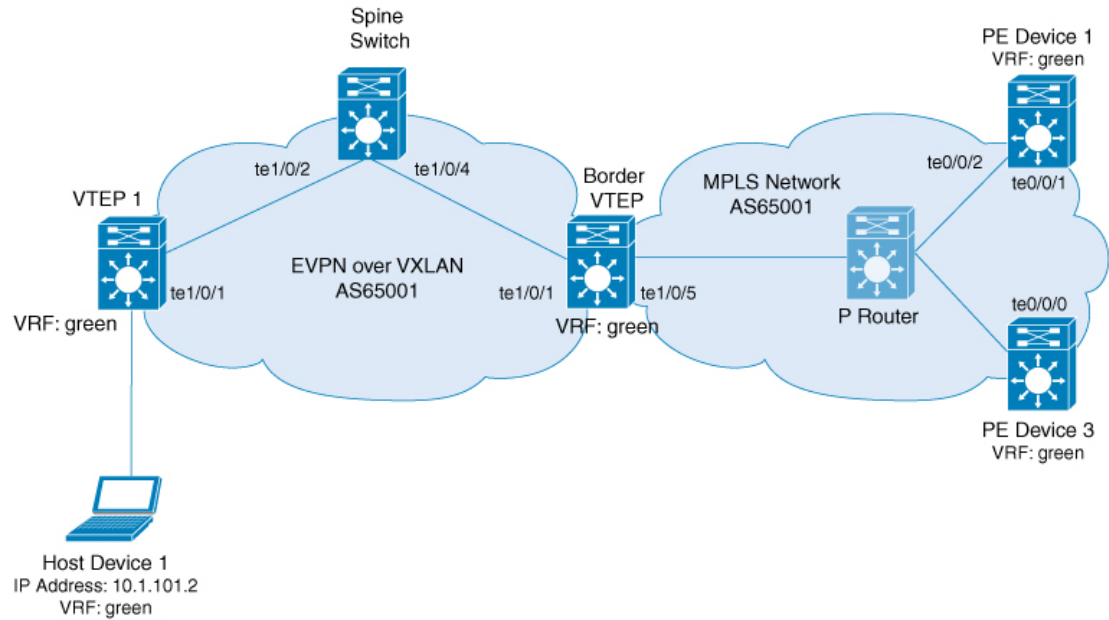
## Configuration Examples for EVPN VXLAN External Connectivity

The following section shows the configuration examples for EVPN VXLAN external connectivity to other technologies:

### Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP

This section provides an example to show how Layer 3 external connectivity with MPLS Layer 3 VPN is enabled for a BGP EVPN VXLAN fabric through iBGP. The example shows how to configure and verify Layer 3 external connectivity with MPLS Layer 3 VPN for the topology shown below:

Figure 17: Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP



The topology shows an EVPN VXLAN network with two VTEPS, VTEP 1 and border VTEP. Border VTEP is connected to an external PE device that belongs to an MPLS network. The BGP EVPN VXLAN fabric and the MPLS network are in the autonomous system number 65001. All the VTEPs, PE devices and, host devices are part of the VRF green. The following tables provide sample configurations for the devices in the topology above.

Table 17: Configuring Spine Switch, Border VTEP and PE Device 1 for Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP

Spine Switch	Border VTEP	PE Device 1
<pre> Spine_switch# show running-config hostname Spine_switch ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ip pim sparse-mode ! interface Loopback1 ip address 172.16.254.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/2 no switchport ip address 172.16.14.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/4 no switchport ip address 172.16.16.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.1 ! router bgp 65001 template peer-policy RR-PP route-reflector-client send-community both exit-peer-policy ! template peer-session RR-PS remote-as 65001 update-source Loopback0 exit-peer-session ! bgp router-id 172.16.255.1 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.4 inherit peer-session RR-PS neighbor 172.16.255.6 inherit peer-session RR-PS ! ! ! </pre>	<pre> Border_VTEP# show running-config hostname Border_VTEP !vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! mpls label mode all-vrfs protocol all-afs per-vrf ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.6 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.6 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/1 no switchport ip address 172.16.16.6 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 </pre>	<pre> PE_device_1# show running-config hostname PE_device_1 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 exit-address-family ! interface Loopback0 ip address 172.16.255.101 255.255.255.255 ! interface Loopback1 vrf forwarding green ip address 10.1.255.101 255.255.255.255 ! interface TenGigabitEthernet0/0/1 ip address 172.16.111.101 255.255.255.0 ip router isis cdp enable mpls ip isis network point-to-point ! interface TenGigabitEthernet0/0/2 ip address 172.16.106.101 255.255.255.0 ip router isis negotiation auto cdp enable mpls ip isis network point-to-point ! router isis net 49.0001.1720.1625.5101.00 is-type level-2-only metric-style wide passive-interface Loopback0 ! router bgp 65001 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.103 remote-as 65001 neighbor 172.16.255.103 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>







Table 18: Configuring VTEP 1 and PE Device 3 for Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP

VTEP 1	PE Device 3
<pre>VTEP_1# show running-config hostname VTEP_1 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0</pre>	<pre>PE_device_3# show running-config hostname PE_device_3 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 exit-address-family ! interface Loopback0 ip address 172.16.255.103 255.255.255.255 ! interface Loopback1 vrf forwarding green ip address 10.1.255.103 255.255.255.255 ! interface TenGigabitEthernet0/0/0 ip address 172.16.111.103 255.255.255.0 ip router isis cdp enable mpls ip isis network point-to-point ! router isis net 49.0001.1720.1625.5103.00 is-type level-2-only metric-style wide passive-interface Loopback0 ! router bgp 65001 template peer-policy RR-PP route-reflector-client send-community both exit-peer-policy ! template peer-session RR-PS remote-as 65001 update-source Loopback0 exit-peer-session ! bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.6 inherit peer-session RR-PS neighbor 172.16.255.101 inherit peer-session RR-PS ! address-family ipv4 exit-address-family ! ! ! ! !</pre>



The following examples provide sample outputs for **show** commands on VTEP 1 and border VTEP to verify external connectivity for the topology configured above:

### VTEP 1

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 5 on VTEP 1:

```
VTEP_1# show bgp l2vpn evpn route-type 5 0 10.1.255.103 32
BGP routing table entry for [5][1:1][0][32][10.1.255.103]/17, version 12
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 1
  Local
    172.16.254.6 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, VNI Label 50901, MPLS VPN
Label 0
  Extended Community: RT:1:1 ENCAP:8 Router MAC:0C75.BD67.EF48
  Originator: 172.16.255.103, Cluster list: 172.16.255.1, 172.16.255.6
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F84B914EF38, path: 0x7F84BAFD0E30, pathext: 0x7F84BB42E698
  flags: net: 0x100, path: 0x3, pathext: 0xA1
  Updated on May 20 2020 19:31:08 UTC
```

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 2 on VTEP 1:

```
VTEP_1# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.2
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.2]/24,
  version 17
Paths: (1 available, best #1, table evi_101)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (172.16.255.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
        Router MAC:7C21.0DBD.9548
      Local irb vxlan vtep:
        vrf:green, l3-vni:50901
        local router mac:7C21.0DBD.9548
        core-irb interface:Vlan901
        vtep-ip:172.16.254.4
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F84B914E858, path: 0x7F84BAFD09F8, pathext: 0x7F84BB42E4B8
      flags: net: 0x0, path: 0x4000028000003, pathext: 0x81
      Updated on May 20 2020 19:31:30 UTC
```

The following example shows the output for the **show ip route vrf** command on VTEP 1:

```
VTEP_1# show ip route vrf green

Routing Table: green
```

## Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP

```

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

```

Gateway of last resort is not set

```

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.1.101.0/24 is directly connected, Vlan101
L       10.1.101.1/32 is directly connected, Vlan101
C       10.1.102.0/24 is directly connected, Vlan102
L       10.1.102.1/32 is directly connected, Vlan102
B       10.1.255.101/32 [200/0] via 172.16.254.6, 00:21:47, Vlan901
B       10.1.255.103/32 [200/0] via 172.16.254.6, 00:21:47, Vlan901

```

## Border VTEP

The following example shows the output for the **show mpls ldp neighbor** command on border VTEP:

```

Border_VTEP# show mpls ldp neighbor
Peer LDP Ident: 172.16.111.101:0; Local LDP Ident 172.16.106.6:0
TCP connection: 172.16.111.101.26371 - 172.16.106.6.646
State: Oper; Msgs sent/rcvd: 86/69; Downstream
Up time: 00:32:14
LDP discovery sources:
  TenGigabitEthernet1/0/5, Src IP addr: 172.16.106.101
Addresses bound to peer LDP Ident:
  172.16.111.101 172.16.106.101 172.16.255.101

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 5 on border VTEP:

```

Border_VTEP# show bgp l2vpn evpn route-type 5 0 10.1.255.103 32
BGP routing table entry for [5][1:1][0][32][10.1.255.103]/17, version 7
Paths: (1 available, best #1, table EVPN-BGP-Table)
Flag: 0x100
Advertised to update-groups:
  1
Refresh Epoch 1
Local, (Received from a RR-client), imported path from base
  172.16.255.103 (metric 20) (via default) from 172.16.255.103 (172.16.255.103)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, local vtep: 172.16.254.6,
VNI Label 50901, MPLS VPN Label 23
  Extended Community: RT:1:1 ENCAP:8 Router MAC:0C75.BD67.EF48
  rx pathid: 0, tx pathid: 0x0
  net: 0x7FED6F808948, path: 0x7FED6D7EDA68, pathext: 0x7FED6D80DE40, exp_net:
0x7FED6F9BF070
  flags: net: 0x100, path: 0x7, pathext: 0xA1
  Updated on May 20 2020 19:22:47 UTC

```

The following example shows the output for the **show bgp vpv4 unicast all** command on border VTEP for the IP address of host device 1:

```
Border_VTEP# show bgp vpv4 unicast all 10.1.101.2
BGP routing table entry for 1:1:10.1.101.2/32, version 10
Paths: (1 available, best #1, table green)
  Advertised to update-groups:
    3
  Refresh Epoch 1
  Local, (Received from a RR-client), imported path from
[2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.2]/24 (global)
  172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  Extended Community: RT:1:1 ENCAP:8 Router MAC:7C21.0DBD.9548
  Originator: 172.16.255.4, Cluster list: 172.16.255.1
  Local vxlan vtep:
    vrf:green, vni:50901
    local router mac:0C75.BD67.EF48
    encap:8
    vtep-ip:172.16.254.6
    bdi:Vlan901
  Remote VxLAN:
    Topoid 0x4(vrf green)
    Remote Router MAC:7C21.0DBD.9548
    Encap 8
    Egress VNI 50901
    RTEP 172.16.254.4
  mpls labels in/out IPv4 VRF Aggr:34/nolabel
  rx pathid: 0, tx pathid: 0x0
  Updated on May 20 2020 19:23:11 UTC
```

## Spine Switch

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 5 on spine switch:

```
Spine_switch# show bgp l2vpn evpn route-type 5 0 10.1.255.103 32
BGP routing table entry for [5][1:1][0][32][10.1.255.103]/17, version 12
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local, (Received from a RR-client)
    172.16.254.6 (metric 2) (via default) from 172.16.255.6 (172.16.255.6)
    Origin incomplete, metric 0, localpref 100, valid, internal, best
    EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, VNI Label 50901, MPLS VPN
Label 0
  Extended Community: RT:1:1 ENCAP:8 Router MAC:0C75.BD67.EF48
  Originator: 172.16.255.103, Cluster list: 172.16.255.6
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F54CC99CEF8, path: 0x7F54CC9AD310, pathext: 0x7F54CC9C6998
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on May 20 2020 19:28:59 UTC
```

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 2 on spine switch:

## Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP

```
Spine_switch# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.2
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.2]/24,
version 14
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local, (Received from a RR-client)
    172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F54CC99CAD8, path: 0x7F54CC9AD088, pathext: 0x7F54CC9C68D8
      flags: net: 0x0, path: 0x3, pathext: 0x81
      Updated on May 20 2020 19:29:22 UTC
```

**PE Device 3**

The following example shows the output for the **show bgp vpnv4 unicast all** command on PE device 3 for the IP address of host device 1:

```
PE_device_3# show bgp vpnv4 unicast all 10.1.101.2
BGP routing table entry for 1:1:10.1.101.2/32, version 14
Paths: (1 available, best #1, table green)
  Advertised to update-groups:
    3
  Refresh Epoch 1
  Local, (Received from a RR-client)
    172.16.255.6 (metric 20) (via default) from 172.16.255.6 (172.16.255.6)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1 ENCAP:8 Router MAC:7C21.0DBD.9548
      Originator: 172.16.255.4, Cluster list: 172.16.255.6, 172.16.255.1
      mpls labels in/out nolabel/34
      rx pathid: 0, tx pathid: 0x0
      Updated on May 20 2020 11:27:25 UTC
```

The following example shows the output for the **show ip route vrf green** command on PE device 3:

```
PE_device_3# show ip route vrf green

Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PFR

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
```

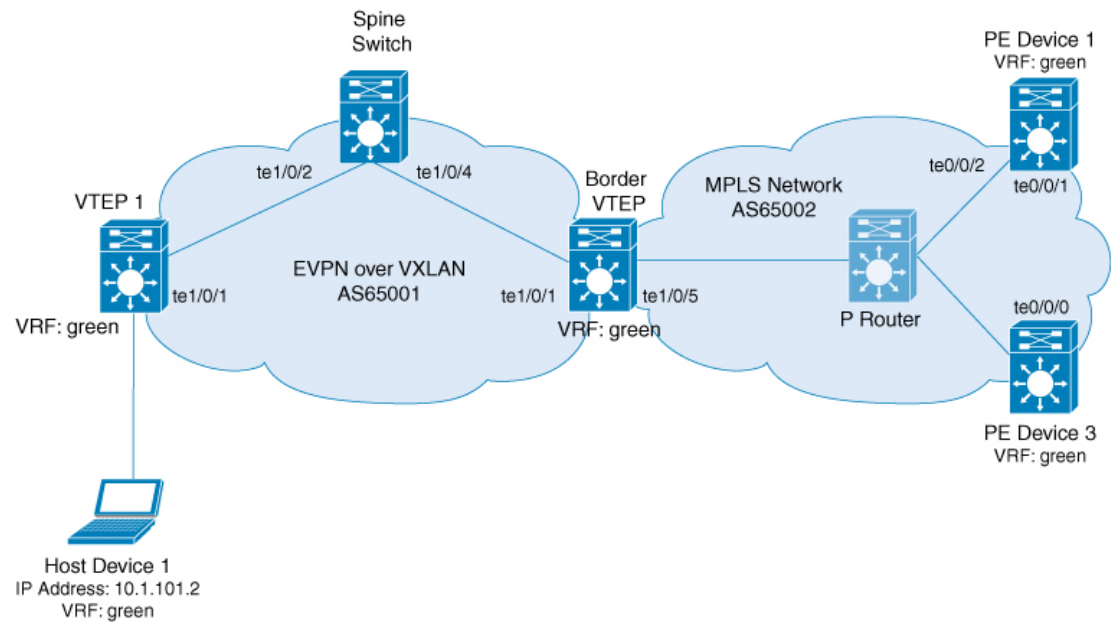
```

B      10.1.101.0/24 [200/0] via 172.16.255.6, 00:28:12
B      10.1.101.1/32 [200/0] via 172.16.255.6, 00:28:10
B      10.1.101.2/32 [200/0] via 172.16.255.6, 00:27:48
B      10.1.102.0/24 [200/0] via 172.16.255.6, 00:28:12
B      10.1.102.1/32 [200/0] via 172.16.255.6, 00:28:10
B      10.1.255.101/32 [200/0] via 172.16.255.101, 00:28:09
C      10.1.255.103/32 is directly connected, Loopback1
    
```

## Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP

This section provides an example to show how Layer 3 external connectivity with MPLS Layer 3 VPN is enabled for a BGP EVPN VXLAN fabric through eBGP. The example shows how to configure and verify Layer 3 external connectivity with MPLS Layer 3 VPN for the topology shown below:

Figure 18: Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP



The topology shows an EVPN VXLAN network with two VTEPs, VTEP 1 and border VTEP. Border VTEP is connected to an external PE device that belongs to an MPLS network. The BGP EVPN VXLAN fabric is in the autonomous system number 65001. The MPLS network is in the autonomous system number 65002. All the VTEPs, PE devices, and host devices are part of the VRF green. The following tables provide sample configurations for the devices in the topology above.

Table 19: Configuring Spine Switch, Border VTEP and PE Device 1 for Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP

Spine Switch	Border VTEP	PE Device 1
<pre> Spine_switch# show running-config hostname Spine_switch ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ip pim sparse-mode ! interface Loopback1 ip address 172.16.254.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback2 ip address 172.16.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/2 no switchport ip address 172.16.14.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/4 no switchport ip address 172.16.16.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 172.16.255.1 ! router bgp 65001 template peer-policy RR-PP route-reflector-client send-community both exit-peer-policy ! template peer-session RR-PS remote-as 65001 update-source Loopback0 exit-peer-session ! bgp router-id 172.16.255.1 bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.4 inherit peer-session RR-PS neighbor 172.16.255.6 inherit peer-session RR-PS ! address-family ipv4 exit-address-family ! </pre>	<pre> Border_VTEP# show running-config hostname Border_VTEP ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! mpls label mode all-vrfs protocol all-afs per-vrf ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.6 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.6 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/1 no switchport ip address 172.16.16.6 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 </pre>	<pre> PE_device_1# show running-config hostname PE_device_1 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 exit-address-family ! interface Loopback0 ip address 172.16.255.101 255.255.255.255 ! interface Loopback1 vrf forwarding green ip address 10.1.255.101 255.255.255.255 ! interface TenGigabitEthernet0/0/1 ip address 172.16.111.101 255.255.255.0 ip router isis cdp enable mpls ip isis network point-to-point ! interface TenGigabitEthernet0/0/2 ip address 172.16.106.101 255.255.255.0 negotiation auto cdp enable mpls bgp forwarding ! router isis net 49.0001.1720.1625.5101.00 is-type level-2-only metric-style wide passive-interface Loopback0 ! router bgp 65002 bgp log-neighbor-changes no bgp default ipv4-unicas no bgp default route-target filter neighbor 172.16.106.6 remote-as 65001 neighbor 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 ebgp-multihop 255 neighbor 172.16.255.6 update-source Loopback0 neighbor 172.16.255.103 remote-as 65002 neighbor 172.16.255.103 update-source Loopback0 </pre>







Table 20: Configuring VTEP 1 and PE Device 3 for Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP

VTEP 1	PE Device 3
<pre>VTEP_1# show running-config hostname VTEP_1! ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-family ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan replication-type ingress ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.4 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 !</pre>	<pre>PE_device_3# show running-config hostname PE_device_3 ! vrf definition green rd 1:1 ! address-family ipv4 route-target export 1:1 route-target import 1:1 exit-address-family ! address-family ipv6 route-target export 1:1 route-target import 1:1 exit-address-family ! interface Loopback0 ip address 172.16.255.103 255.255.255.255 ! interface Loopback1 vrf forwarding green ip address 10.1.255.103 255.255.255.255 ! interface TenGigabitEthernet0/0/0 ip address 172.16.111.103 255.255.255.0 ip router isis cdp enable mpls ip isis network point-to-point ! router isis net 49.0001.1720.1625.5103.00 is-type level-2-only metric-style wide passive-interface Loopback0 ! router bgp 65002 template peer-policy RR-PP route-reflector-client send-community both exit-peer-policy ! template peer-session RR-PS remote-as 65002 update-source Loopback0 exit-peer-session ! bgp log-neighbor-changes no bgp default ipv4-unicast neighbor 172.16.255.101 inherit peer-session RR-PS ! address-family ipv4 exit-address-family ! ! !</pre>



The following examples provide sample outputs for **show** commands on the devices to verify external connectivity for the topology configured above:

### VTEP 1

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 5 on VTEP 1:

```
VTEP_1# show bgp l2vpn evpn route-type 5 0 10.1.255.103 32
BGP routing table entry for [5][1:1][0][32][10.1.255.103]/17, version 36
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 1
  65002
    172.16.254.6 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, VNI Label 50901, MPLS VPN
Label 0
  Extended Community: RT:1:1 ENCAP:8 Router MAC:0C75.BD67.EF48
  Originator: 172.16.255.6, Cluster list: 172.16.255.1
  rx pathid: 0, tx pathid: 0x0
  net: 0x7F84BB35A5C8, path: 0x7F84B913E010, pathext: 0x7F84BB54A8A8
  flags: net: 0x0, path: 0x3, pathext: 0x81
  Updated on May 21 2020 13:56:28 UTC
```

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 2 on VTEP 1:

```
VTEP_1# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.2
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.2]/24,
version 37
Paths: (1 available, best #1, table evi_101)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    :: (via default) from 0.0.0.0 (172.16.255.4)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
      Router MAC:7C21.0DBD.9548
      Local irb vxlan vtep:
        vrf:green, l3-vni:50901
        local router mac:7C21.0DBD.9548
        core-irb interface:Vlan901
        vtep-ip:172.16.254.4
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F84BB35A468, path: 0x7F84B913DF38, pathext: 0x7F84BB54A848
      flags: net: 0x0, path: 0x4000028000003, pathext: 0x81
      Updated on May 21 2020 14:00:49 UTC
```

The following example shows the output for the **show ip route vrf** command on VTEP 1:

```
VTEP_1# show ip route vrf green

Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
```

## Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP

```

n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PFR

```

Gateway of last resort is not set

```

10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C    10.1.101.0/24 is directly connected, Vlan101
L    10.1.101.1/32 is directly connected, Vlan101
C    10.1.102.0/24 is directly connected, Vlan102
L    10.1.102.1/32 is directly connected, Vlan102
B    10.1.255.101/32 [200/0] via 172.16.254.6, 00:06:25, Vlan901
B    10.1.255.103/32 [200/0] via 172.16.254.6, 00:05:54, Vlan901

```

### Border VTEP

The following example shows the output for the **show bgp vpnv4 unicast all** command on border VTEP for the IP address of the external device:

```

Border_VTEP# show bgp vpnv4 uni all 10.1.255.103/32
BGP routing table entry for 1:1:10.1.255.103/32, version 9
Paths: (1 available, best #1, table green)
  Not advertised to any peer
  Refresh Epoch 1
  65002
    172.16.255.101 (via default) from 172.16.255.101 (172.16.255.101)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:1:1
      Local vxlan vtep:
        vrf:green, vni:50901
        local router mac:0C75.BD67.EF48
        encaps:8
        vtep-ip:172.16.254.6
        bdi:Vlan901
      mpls labels in/out nolabel/16
      rx pathid: 0, tx pathid: 0x0
      Updated on May 21 2020 13:48:09 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 5 on border VTEP:

```

Border_VTEP# show bgp l2vpn evpn route-type 5 0 10.1.255.103 32
BGP routing table entry for [5][1:1][0][32][10.1.255.103]/17, version 32
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  65002, imported path from base
    172.16.255.101 (via default) from 172.16.255.101 (172.16.255.101)
      Origin incomplete, localpref 100, valid, external, best
      EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, local vtep: 172.16.254.6,
VNI Label 50901, MPLS VPN Label 16
      Extended Community: RT:1:1 ENCAP:8 Router MAC:0C75.BD67.EF48
      rx pathid: 0, tx pathid: 0x0
      net: 0x7FED704944D0, path: 0x7FED704A4CA0, pathext: 0x7FED6DA6E250, exp_net:
0x7FED6F812678

```

```
flags: net: 0x0, path: 0x7, pathext: 0x81
Updated on May 21 2020 13:48:09 UTC
```

The following example shows the output for the **show mpls forwarding-table** command on border VTEP:

```
Border_VTEP# show mpls forwarding-table
Local   Outgoing Prefix          Bytes Label   Outgoing  Next Hop
Label   Label    or Tunnel Id    Switched     interface
16      No Label IPv4 VRF[V]     156         aggregate/green
17      Pop Label 172.16.106.101/32 \
                                     228         Te1/0/5    172.16.106.101
18      Pop Label 172.16.255.101/32 \
                                     0          Te1/0/5    172.16.106.101
```

The following example shows the output for the **show bgp vpnv4 unicast all** command on border VTEP for the IP address of host device 1:

```
Border_VTEP# show bgp vpnv4 uni all 10.1.101.2/32
BGP routing table entry for 1:1:10.1.101.2/32, version 10
Paths: (1 available, best #1, table green)
  Advertised to update-groups:
    1
  Refresh Epoch 4
  Local, imported path from [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.2]/24
(global)
  172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  Extended Community: RT:1:1 ENCAP:8 Router MAC:7C21.0DBD.9548
  Originator: 172.16.255.4, Cluster list: 172.16.255.1
  Local vxlan vtep:
    vrf:green, vni:50901
    local router mac:0C75.BD67.EF48
    encap:8
    vtep-ip:172.16.254.6
    bdi:Vlan901
  Remote VxLAN:
    Topoid 0x9(vrf green)
    Remote Router MAC:7C21.0DBD.9548
    Encap 8
    Egress VNI 50901
    RTEP 172.16.254.4
  mpls labels in/out IPv4 VRF Aggr:16/nolabel
  rx pathid: 0, tx pathid: 0x0
  Updated on May 21 2020 13:52:30 UTC
```

## Spine Switch

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 5 on spine switch:

```
Spine_switch# show bgp l2vpn evpn route-type 5 0 10.1.255.103 32
BGP routing table entry for [5][1:1][0][32][10.1.255.103]/17, version 23
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  65002, (Received from a RR-client)
    172.16.254.6 (metric 2) (via default) from 172.16.255.6 (172.16.255.6)
```

## Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP

```

Origin incomplete, metric 0, localpref 100, valid, internal, best
EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, VNI Label 50901, MPLS VPN
Label 0
Extended Community: RT:1:1 ENCAP:8 Router MAC:0C75.BD67.EF48
rx pathid: 0, tx pathid: 0x0
net: 0x7F54CC95FAB8, path: 0x7F54CCA542F8, pathext: 0x7F54CC9707B0
flags: net: 0x0, path: 0x3, pathext: 0x81
Updated on May 21 2020 13:54:20 UTC

```

The following example shows the output for the **show bgp l2vpn evpn route-type** command for route type 2 on spine switch:

```

Spine_switch# show bgp l2vpn evpn route-type 2 0 44d3ca286cc1 10.1.101.2
BGP routing table entry for [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.2]/24,
version 24
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local, (Received from a RR-client)
    172.16.254.4 (metric 2) (via default) from 172.16.255.4 (172.16.255.4)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10101, Label2 50901
      Extended Community: RT:1:1 RT:65001:101 ENCAP:8
        Router MAC:7C21.0DBD.9548
        rx pathid: 0, tx pathid: 0x0
        net: 0x7F54CC95F958, path: 0x7F54CCA54220, pathext: 0x7F54CC970750
        flags: net: 0x0, path: 0x3, pathext: 0x81
        Updated on May 21 2020 13:58:41 UTC

```

**PE Device 1**

The following example shows the output for the **show bgp vpnv4 unicast all** command on PE device 1 for the IP address of host device 1:

```

PE_device_1# show bgp vpnv4 unicast all 10.1.255.103/32
BGP routing table entry for 1:1:10.1.101.2/32, version 14
Paths: (1 available, best #1, table green)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  65001
    172.16.255.6 (via default) from 172.16.255.6 (172.16.255.6)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:1:1 ENCAP:8 Router MAC:7C21.0DBD.9548
      mpls labels in/out 22/16
      rx pathid: 0, tx pathid: 0x0
      Updated on May 21 2020 05:57:06 UTC

```

The following example shows the output for the **show ip route vrf** command on PE device 1:

```

PE_device_1# show ip route vrf green

Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

```



```

E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR

```

Gateway of last resort is not set

```

10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
B    10.1.101.0/24 [20/0] via 172.16.255.6, 00:28:09
B    10.1.101.1/32 [20/0] via 172.16.255.6, 00:28:09
B    10.1.101.2/32 [20/0] via 172.16.255.6, 00:23:17
B    10.1.102.0/24 [20/0] via 172.16.255.6, 00:28:09
B    10.1.102.1/32 [20/0] via 172.16.255.6, 00:28:09
C    10.1.255.101/32 is directly connected, Loopback1
B    10.1.255.103/32 [200/0] via 172.16.255.103, 00:28:09

```

### PE Device 3

The following example shows the output for the **show bgp vpnv4 unicast all** command on PE device 3 for the IP address of host device 1:

```

PE_device_3# show bgp vpnv4 unicast all 10.1.101.2/32
BGP routing table entry for 1:1:10.1.101.2/32, version 14
Paths: (1 available, best #1, table green)
  Not advertised to any peer
  Refresh Epoch 1
  65001, (Received from a RR-client)
    172.16.255.101 (metric 10) (via default) from 172.16.255.101 (172.16.255.101)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1 ENCAP:8 Router MAC:7C21.0DBD.9548
      mpls labels in/out nolabel/22
      rx pathid: 0, tx pathid: 0x0
      Updated on May 21 2020 05:56:46 UTC

```

The following example shows the output for the **show ip route vrf** command on PE device 3:

```

PE_device_3# show ip route vrf green

Routing Table: green
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

```

Gateway of last resort is not set

```

10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks

```

## Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP

```
B      10.1.101.0/24 [200/0] via 172.16.255.101, 00:29:09
B      10.1.101.1/32 [200/0] via 172.16.255.101, 00:29:09
B      10.1.101.2/32 [200/0] via 172.16.255.101, 00:24:17
B      10.1.102.0/24 [200/0] via 172.16.255.101, 00:29:09
B      10.1.102.1/32 [200/0] via 172.16.255.101, 00:29:09
B      10.1.255.101/32 [200/0] via 172.16.255.101, 00:29:09
C      10.1.255.103/32 is directly connected, Loopback1
```



## CHAPTER 9

# Troubleshooting BGP EVPN VXLAN

- [Troubleshooting Scenarios for BGP EVPN VXLAN, on page 227](#)
- [Troubleshooting Broadcast, Unknown Unicast, Multicast Traffic Forwarding, on page 228](#)
- [Troubleshooting Unicast Forwarding Between VTEPs in the Same VLAN Through a Layer 2 VNI, on page 232](#)
- [Troubleshooting Unicast Forwarding Between VTEPs in Different VLANs Through a Layer 3 VNI, on page 244](#)
- [Troubleshooting Unicast Forwarding Between a VXLAN Network and an IP Network, on page 257](#)

## Troubleshooting Scenarios for BGP EVPN VXLAN

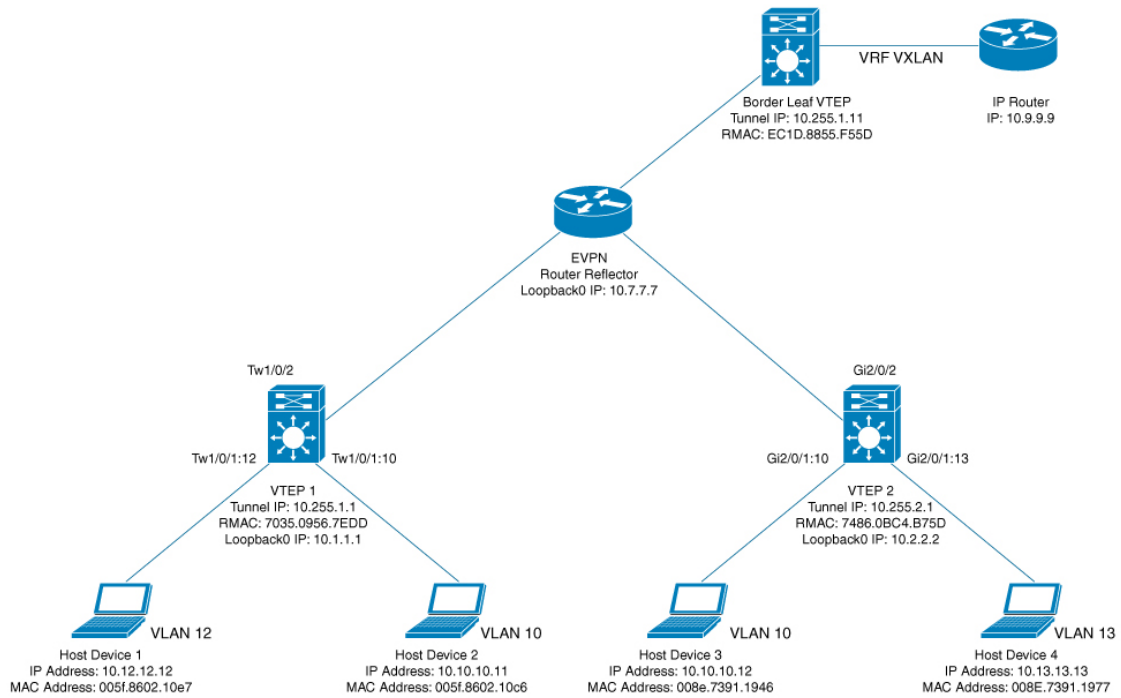
This document provides information about the various troubleshooting scenarios that are applicable to BGP EVPN VXLAN and how to troubleshoot each scenario.

In this troubleshooting document, comments have been added at the end of certain lines of the outputs of **show** commands. This has been done to highlight or explain a specific aspect of that line of output. If a comment begins in a new line, then it refers to the line of output that precedes the comment. The following notation has been used throughout the document to highlight the comments inside the outputs of **show** commands:

```
<<- Text highlighted in this format inside a command's output represents a comment.  
This is done for explanation purpose only and is not part of the command's output.
```

The following is a sample EVPN VXLAN topology with two access facing VTEPs (VTEP 1 and VTEP 2) and a border leaf VTEP connected in a VXLAN network through an EVPN route reflector. Each of the access facing VTEPs has two host devices connected to it and the border leaf VTEP is connected to an external IP network. All the troubleshooting scenarios in this document are explained using this topology.

Figure 19: EVPN VXLAN Topology



The following are the various troubleshooting scenarios that apply to BGP EVPN VXLAN for the topology illustrated in the [Figure 19: EVPN VXLAN Topology](#) above:

- **Scenario 1:** Troubleshooting Broadcast, Unknown Unicast, Multicast traffic Forwarding
- **Scenario 2:** Troubleshooting Unicast Forwarding Between VTEPs in the Same VLAN Through a Layer 2 VNI
- **Scenario 3:** Troubleshooting Unicast Forwarding Between VTEPs in Different VLANs Through a Layer 3 VNI
- **Scenario 4:** Troubleshooting Unicast Forwarding Between a VXLAN Network and an IP Network

## Troubleshooting Broadcast, Unknown Unicast, Multicast Traffic Forwarding

This scenario might occur when host device 2 attempts to learn the ARP for host device 3 in [Figure 19: EVPN VXLAN Topology, on page 228](#). Perform the checks listed in the following table before troubleshooting BUM traffic forwarding:

**Table 21: Scenario 1: Broadcast, Unknown Unicast, Multicast traffic Forwarding**

Check to be Performed	Steps to Follow
Is the packet of broadcast type?	Check if the packet is a broadcast packet, such as an ARP broadcast packet.
Are the hosts in the same subnet or in different subnets?	Perform any of the following steps: <ul style="list-style-type: none"> <li>• Check the host device.</li> <li>• Check the SVI configuration on the VTEP.</li> </ul>
Has the remote MAC address been learned for unknown unicast traffic?	Run the <b>show platform software fed switch active macTable vlan <i>vlan-id</i></b> command in privileged EXEC mode on the local VTEP and check if the MAC address of the remote host device is displayed in the output. If not, you have not yet learned the remote host device and it needs to be resolved.

BUM traffic is forwarded by a VTEP into the VXLAN Core using multicast routing. In order to follow the path of an ARP broadcast packet, you need to identify the multicast group that needs to be used to send this traffic into the core and to the other VTEPs. BUM traffic first arrives at the local Layer 2 interface. The traffic is encapsulated here and sent out using the multicast group that is sourced from the VXLAN Loopback interface.



**Note** Underlay multicast needs to be fully configured before troubleshooting BUM traffic forwarding for EVPN VXLAN.

To troubleshoot EVPN VXLAN BUM traffic forwarding, follow these steps:

1. [Determine the MAC Address of the Local Host Device and the Multicast Group Used for ARP Tunneling, on page 229](#)
2. [Set Up Embedded Capture Towards the Core-Facing Interface, on page 230](#)
3. [Ping the Remote Host Device, on page 230](#)
4. [Verify that an ARP Request Has Been Received and a Multicast Route Has Been Built, on page 230](#)
5. [Confirm the Presence of ARP Request Replies in Embedded Capture, on page 231](#)
6. [Verify that the Encapsulated ARP Request is Leaving in a Multicast Group to a VXLAN UDP Destination Port, on page 231](#)
7. [Verify that the ARP Reply from Core Interface is Encapsulated in Unicast to a VXLAN UDP Destination Port, on page 232](#)

**Determine the MAC Address of the Local Host Device and the Multicast Group Used for ARP Tunneling**

The following examples show how to verify the MAC address of the local host device and the multicast group that is used for tunneling the ARP broadcast request:

```

VTEP-1# show mac address-table address 005f.8602.10c6
Mac Address Table
-----
Vlan Mac Address Type      Ports
----
10 005f.8602.10c6 DYNAMIC Tw1/0/1  <<- MAC address of 10.10.10.11 is learnt here

VTEP-1# show run int nve 1
interface nve1
 no ip address
 source-interface Loopback999
 host-reachability protocol bgp
 member vni 10001 mcast-group 239.10.10.10  <<- Group is mapped to the VNI under NVE

VTEP-1# show run | s vlan conf
vlan configuration 10
 member evpn-instance 10 vni 10001  <<- VNI mapped under VLAN 10

VTEP-1# show l2vpn evpn evi
EVI   VLAN  Ether Tag  L2 VNI   Multicast   Pseudoport
-----
10    10    0          10001   239.10.10.10 Tw1/0/1:10
      <<- EVPN instance 10 is mapped to VLAN 10 and VNI 10001
      (Using multicast group 239.10.10.10 for Broadcast ecap tunnel)
<...snip...>

```

### Set Up Embedded Capture Towards the Core-Facing Interface

The following example shows how to set up embedded capture towards the core-facing interface:



**Note** On a production network, use this command with a filter.

```

VTEP-1# show monitor capture 1 parameter
monitor capture 1 interface TwoGigabitEthernet1/0/2 BOTH
monitor capture 1 match any
monitor capture 1 buffer size 100
monitor capture 1 limit pps 1000

```

### Ping the Remote Host Device

The following example shows how to ping the remote host device:

```

VTEP-1-HOST# ping 10.10.10.12  <<- sourced from Host machine 10.10.10.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.12, timeout is 2 seconds:
..!!!

```

### Verify that an ARP Request Has Been Received and a Multicast Route Has Been Built

This step is to verify that there is multicast reachability between VTEPs using standard multicast validation. Underly multicast state is not permanent. If it is not in use, these S,G states will expire.

The following output confirms that an ARP request has been received and a multicast route has been built:

```

VTEP-1# show ip mroute 239.10.10.10 10.255.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report,
Z - Multicast Tunnel, z - MDT-data group sender,
Y - Joined MDT-data group, y - Sending to MDT-data group,
G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
Q - Received BGP S-A Route, q - Sent BGP S-A Route,
V - RD & Vector, v - Vector, p - PIM Joins on route,
x - VxLAN group, c - PFP-SA cache created entry
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(10.255.1.1, 239.10.10.10), 00:00:25/00:02:34, flags: FTx <<- x flag set for VxLAN group
Incoming interface: Loopback999, RPF nbr 0.0.0.0 <<- Broadcast being encapsulated
into VXLAN tunnel IP

Outgoing interface list:
TwoGigabitEthernet1/0/2, Forward/Sparse, 00:00:23/00:03:06
<<- Sending towards core to VTEP-2
(10.255.1.4, 239.10.10.10), 3d18h/00:02:25, flags: JTx <<- BUM traffic from VTEP-2 (if the
ARP request was from VTEP-2)
Incoming interface: TwoGigabitEthernet1/0/2, RPF nbr 10.1.1.6
Outgoing interface list:
Tunnel0, Forward/Sparse-Dense, 3d18h/00:00:14 <<- Tunnel 0 is the VXLAN tunnel
used for decapsulation
    
```

### Confirm the Presence of ARP Request Replies in Embedded Capture

The following output confirms that the ARP request replies are present in embedded capture:

```

VTEP-1# show monitor capture 1 buffer display-filter "arp"
Starting the packet display ..... Press Ctrl + Shift + 6 to exit

7 0.000018 00:5f:86:02:10:c6 -> ff:ff:ff:ff:ff:ff ARP 110 Who has 10.10.10.12? Tell
10.10.10.11
9 0.000022 28:52:61:bf:a9:46 -> 00:5f:86:02:10:c6 ARP 110 10.10.10.12 is at 28:52:61:bf:a9:46
    
```

### Verify that the Encapsulated ARP Request is Leaving in a Multicast Group to a VXLAN UDP Destination Port

The following image shows the ARP request leaving encapsulated in the multicast group 239.10.10.10, sourced from a VXLAN Loopback, to the VXLAN UDP destination port 4789 in the VNI 10001 and VLAN 10.

```

No.    Time    Source                Destination            Protocol    Length  Info
-----
1 0.000  00:5f:86:02:10:c6    ff:ff:ff:ff:ff:ff     ARP        110     Who has 10.10.10.12? Tell 10.10.10.11
2 0.000  28:52:61:bf:a9:46    00:5f:86:02:10:c6     ARP        110     10.10.10.12 is at 28:52:61:bf:a9:46

> Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
> Ethernet II, Src: 74:a2:e6:4f:c9:00, Dst: 01:00:5e:0a:0a:0a
  > Destination: 01:00:5e:0a:0a:0a
  > Source: 74:a2:e6:4f:c9:00
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.255.1.1, Dst: 239.10.10.10
> User Datagram Protocol, Src Port: 65419 (65419), Dst Port: 4789 (4789)
  Source Port: 65419
  Destination Port: 4789
  Length: 76
  > Checksum: 0x0000 (none)
  [Stream index: 0]
> Virtual eXtensible Local Area Network
  > Flags: 0x0000, VXLAN Network ID (VNI)
  Group Policy ID: 0
  VXLAN Network Identifier (VNI): 10001
  Reserved: 0
> Ethernet II, Src: 00:5f:86:02:10:c6, Dst: ff:ff:ff:ff:ff:ff
  > Destination: ff:ff:ff:ff:ff:ff
  > Source: 00:5f:86:02:10:c6
  Type: ARP (0x0806)
  Trailer: 0000000000000000000000000000000000000000000000000000000000000000
  Address Resolution Protocol (request)
  
```

**Verify that the ARP Reply from Core Interface is Encapsulated in Unicast to a VXLAN UDP Destination Port**

The following image shows the ARP reply from core interface that is encapsulated in unicast, between VXLAN Loopbacks, to the VXLAN UDP destination port 4789 in the VNI 10001 and VLAN 10.

```

No.    Time    Source                Destination            Protocol    Length  Info
-----
1 0.000  00:5f:86:02:10:c6    ff:ff:ff:ff:ff:ff     ARP        110     Who has 10.10.10.12? Tell 10.10.10.11
2 0.000  28:52:61:bf:a9:46    00:5f:86:02:10:c6     ARP        110     10.10.10.12 is at 28:52:61:bf:a9:46

> Frame 2: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
> Ethernet II, Src: 74:a2:e6:4f:c9:00, Dst: 70:35:09:56:7e:d6
  > Destination: 70:35:09:56:7e:d6
  > Source: 74:a2:e6:4f:c9:00
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.255.1.2, Dst: 10.255.1.1
> User Datagram Protocol, Src Port: 65350 (65350), Dst Port: 4789 (4789)
  Source Port: 65350
  Destination Port: 4789
  Length: 76
  > Checksum: 0x0000 (none)
  [Stream index: 1]
> Virtual eXtensible Local Area Network
  > Flags: 0x0000, VXLAN Network ID (VNI)
  Group Policy ID: 0
  VXLAN Network Identifier (VNI): 10001
  Reserved: 0
> Ethernet II, Src: 28:52:61:bf:a9:46, Dst: 00:5f:86:02:10:c6
  > Destination: 00:5f:86:02:10:c6
  > Source: 28:52:61:bf:a9:46
  Type: ARP (0x0806)
  Trailer: 0000000000000000000000000000000000000000000000000000000000000000
  Address Resolution Protocol (reply)
  
```

Once all of the above checks are verified, if there is still a problem with broadcast reachability, then repeat the checks on the remote VTEP.

# Troubleshooting Unicast Forwarding Between VTEPs in the Same VLAN Through a Layer 2 VNI

This scenario might occur when host device 2 in VLAN 10 attempts to ping host device 3 that is also in VLAN 10. Perform the checks listed in the following table before troubleshooting unicast forwarding between VTEPs in the same VLAN through a Layer 2 VNI:



**Table 22: Scenario 2: Troubleshooting Unicast Forwarding Between VTEPs in the Same VLAN Through a Layer 2 VNI**

Check to be Performed	Steps to Follow
Has ARP been resolved on the local host for the Layer 2 adjacent remote host?	Run the <b>arp -a</b> command in privileged EXEC mode on the host device.
Do the hosts have the same subnet masks?	Perform any of the following steps: <ul style="list-style-type: none"> <li>• Check the host device.</li> <li>• Check the SVI configuration on the VTEP.</li> </ul>
Do you have the EVPN instance configured on your local VTEP?	Run the following commands in privileged EXEC mode on the VTEP: <ul style="list-style-type: none"> <li>• <b>show run   section l2vpn</b></li> <li>• <b>show run   section vlan config</b></li> <li>• <b>show run interface nve interface-number</b></li> </ul>
Has the remote MAC address been learned in platform MATM in the same VLAN as the local host?	Run the <b>show platform software fed switch active matm macTable vlan vlan-id</b> command in privileged EXEC mode on the VTEP to check for the remote MAC addresses in the same VLAN.

To troubleshoot unicast forwarding between two VTEPs in the same VLAN using a Layer 2 VNI, follow these steps:

- Verify the provisioning of the EVPN VXLAN Layer 2 overlay network.
- Verify intra-subnet traffic movement in the EVPN VXLAN Layer 2 overlay network.

## Verifying the Provisioning of an EVPN VXLAN Layer 2 Overlay Network

To verify the provisioning of an EVPN VXLAN Layer 2 overlay network, perform these checks:

1. [Verify the Provisioning of the EVPN Instance in EVPN Manager, on page 233](#)
2. [Ensure that an NVE Peer is Present for the Layer 2 VNI, on page 235](#)
3. [Verify the Provisioning of the Layer 2 VNI in NVE Component, on page 235](#)
4. [Verify That the Layer 2 VNI VXLAN Tunnel Pseudoport is added to the Access VLAN in Layer 2 Forwarding Information Base \(FIB\), on page 236](#)

### Verify the Provisioning of the EVPN Instance in EVPN Manager

The following examples show how to verify that the EVPN instance is provisioned in the EVPN manager:

```
VTEP-1# show run | section l2vpn
l2vpn evpn instance 10 vlan-based
encapsulation vxlan
```

```

route-target export 10:1      <<- Import or export right route-targets

route-target import 10:2     <<- Import or export right route-targets

VTEP-1# show run | section vlan config
vlan configuration 10
member evpn-instance 10 vni 10001  <<- EVPN instance & VNI mapped to the VLAN

VTEP-1# show run interface nve1
interface nve1
source-interface Loopback999
host-reachability protocol bgp
member vni10001 mcast-group 239.10.10.10  <<- VNI added to NVE interface

VTEP-1# show run interface loopback 999
interface Loopback999
description VxLAN Loopback
ip address 10.255.1.1 255.255.255.255

```



**Note** Run the **show run** commands on VTEP 2 to verify its configuration, if required.

```

VTEP-1# show l2vpn evpn evi 10 detail <<- VLAN number and EVPN Instance number
                                     are not always the same, confirm which
                                     EVPN Instance maps to your VLAN
                                     with the show l2vpn evpn evi command

EVPN instance: 10 (VLAN Based) <<- EVPN Instance number does map to the VLAN.
RD: 10.1.1.1:10 (auto)
Import-RTs: 10:2 <<- Importing VTEP-2 (if you are not seeing the prefix,
                 check configuration for the right import/export statement
                 under the l2vpn evpn instance)

Export-RTs: 10:1
Per-EVI Label: none
State: Established
Encapsulation: vxlan
Vlan: 10 <<- Layer 2 VLAN
Ethernet-Tag: 0
State: Established <<- If State is not "Established", there
                 could be a misconfiguration

Core If: Vlan99
Access If: Vlan10
NVE If: nve1
RMAC: 7035.0956.7edd
Core Vlan: 99
L2 VNI: 10001 <<- Layer 2 VNI
L3 VNI: 99999
VTEP IP: 10.255.1.1
MCAST IP: 239.10.10.10 <<- BUM Group for flooded traffic (Layer 2 learning, etc)

VRF: vxlan
IPv4 IRB: Enabled
IPv6 IRB: Enabled
Pseudoports:
  TwoGigabitEthernet1/0/1 service instance 10
<<- Layer 2 Access pseudoport (combination of Layer 2 port and service instance)

```



**Note** If only a Layer 2 overlay network has been configured for bridging, then the `Core If`, `Access If`, `RMAC`, `Core BD`, `L3 VNI`, and `VRF` fields do not show any values as they are not set.

```
VTEP-2# show l2vpn evpn evi 10 detail
EVPN instance: 10 (VLAN Based)
RD: 10.2.2.2:10 (auto)
Import-RTs: 10:1 <<- Importing VTEP-1 route-target
Export-RTs: 10:2
Per-EVI Label: none
State: Established
Encapsulation: vxlan
Vlan: 10 <<- Layer 2 VLAN
  Ethernet-Tag: 0
  State: Established
  Core If: Vlan99
  Access If: Vlan10
  NVE If: nve1
  RMAC: 7486.0bc4.b75d
  Core Vlan: 99
  L2 VNI: 10001 <<- Layer 2 VNI
  L3 VNI: 99999
  VTEP IP: 10.255.2.1
  MCAST IP: 239.10.10.10
  VRF: vxlan
  IPv4 IRB: Enabled
  IPv6 IRB: Enabled
  Pseudoports:
    GigabitEthernet2/0/1 service instance 10
    <<- Layer 2 Access pseudoport (combination of Layer 2 port and service instance)
```

### Ensure that an NVE Peer is Present for the Layer 2 VNI

The following examples show how to check if an NVE peer is present for the Layer 2 VNI:

```
VTEP-1# show nve peers vni 10001 <<- This VNI is learned from "show l2vpn evpn evi"
Interface VNI Type Peer-IP RMAC/Num_RTs eVNI state flags UP time
nve1 10001 L2CP 10.255.2.1 2 10001 UP N/A 00:01:03
<<- Layer 2 Control Plane (L2CP) peer for the VNI is an indicator that this is
Layer 2 forwarding
<<- Interface NVE1, L2CP, egress VNI are shown, state is UP for a time of 00:01:03

VTEP-2# show nve peers vni 10001
Interface VNI Type Peer-IP RMAC/Num_RTs eVNI state flags UP time
nve1 10001 L2CP 10.255.1.1 3 10001 UP N/A 00:47:2
<<- Interface NVE1, L2CP, egress VNI are shown, state is UP for a time of 00:47:2
```

### Verify the Provisioning of the Layer 2 VNI in NVE Component

The following example shows how to verify that the Layer 2 VNI is provisioned in the NVE component:

```
VTEP-1# show nve vni 10001 detail <<- VNI 10001 is correlated to VLAN 10
from show l2vpn evpn evi
Interface VNI Multicast-group VNI state Mode VLAN cfg vrf
nve1 10001 239.10.10.10 Up L2CP 10 CLI vxlan
```

<<- state is UP, type is Layer 2 VNI (L2CP); VLAN 10 is mapped to VNI 10001

L2 VNI IPv6 IRB down reason:  
BDI or associated L3 BDI's IPv6 addr un-configured  
IPv6 topo\_id disabled

L2CP VNI local VTEP info: <<- Layer 2 VNI provisioning  
VLAN: 10 <<- Confirms that mapping is with VLAN 10  
SVI if handler: 0x4D  
Local VTEP IP: 10.255.1.1 <<- VxLAN Tunnel IP

Core IRB info: <<- Layer 3 VPN provisioning (not required for troubleshooting  
a scenario with pure Layer 2 VPN packet path

L3VNI: 99999  
VRF name: vxlan  
VLAN: 99  
V4TopoID: 0x2  
V6TopoID: 0xFFFF  
Local VTEP IP: 10.255.1.1  
SVI if handler: 0x50  
SVI MAC: 7035.0956.7EDD

VNI Detailed statistics:  
Pkts In Bytes In Pkts Out Bytes Out  
0 0 18158681548 27383291735556

### Verify That the Layer 2 VNI VXLAN Tunnel Pseudoport is added to the Access VLAN in Layer 2 Forwarding Information Base (FIB)

The following examples show how to verify that the Layer 2 VXLAN tunnel pseudoport is added to the access VLAN in Layer 2 FIB:

```
VTEP-1# show l2fib bridge-domain 10 detail <<- Bridge-domain will be same as VLAN number
Bridge Domain : 10
Reference Count : 14
Replication ports count : 2
Unicast Address table size : 3
IP Multicast Prefix table size : 3

Flood List Information :
Olist: 5109, Ports: 2

VxLAN Information :
VXLAN_DEC nv1:10001:239.10.10.10

Port Information :
BD_PORT Tw1/0/1:10 <<- Pseudoport has been added to bridge-domain:
          (physical port + the BD number for the VLAN)
VXLAN_REP nv1:10001:239.10.10.10 <<- VXLAN Replication group

Unicast Address table information :
008e.7391.1946 VXLAN_CP L:10001:10.255.1.1 R:10001:10.255.2.1

IP Multicast Prefix table information :
Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 5109, Ports: 2
Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 5109, Ports: 2
Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 5109, Ports: 2
```

```

VTEP-2# show l2fib bridge-domain 10 detail
Bridge Domain : 10
Reference Count : 15
Replication ports count : 2
Unicast Address table size : 4
IP Multicast Prefix table size : 3

Flood List Information :
Olist: 5109, Ports: 2

VxLAN Information :
VXLAN_DEC nvl:10001:239.10.10.10

Port Information :
BD_PORT Gi2/0/1:10 <<- Pseudoport has been added to bridge-domain:
          (physical port + the BD number for the VLAN)
VXLAN_REP nvl:10001:239.10.10.10 <<- VXLAN replication group

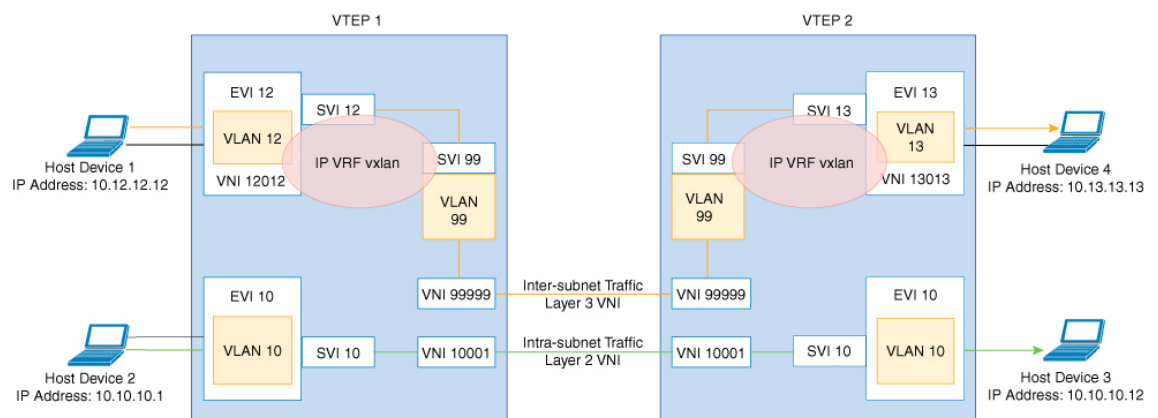
Unicast Address table information :
005f.8602.10c6 VXLAN_CP L:10001:10.255.2.1 R:10001:10.255.1.1

IP Multicast Prefix table information :
Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 5109, Ports: 2
Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 5109, Ports: 2
Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 5109, Ports: 2
    
```

## Verifying Intra-Subnet Traffic Movement in an EVPN VXLAN Layer 2 Overlay Network

The following figure illustrates the movement of traffic from host devices connected to VTEP 1 to host devices connected to VTEP 2:

**Figure 20: Movement of traffic in an EVPN VXLAN network Through Layer 2 and Layer 3 VNIs**



In the above figure, Layer 2 traffic moves from host device 2 to host device 3 through the Layer 2 VNI 10001. To verify the movement of intra-subnet traffic in the EVPN VXLAN Layer 2 overlay network, perform these checks:

1. [Verify that the Local MAC Addresses Have Been Learned in IOS-MATM, on page 238](#)
2. [Verify that Both Local and Remote MAC Addresses are Learned in FED-MATM, on page 238](#)

3. Confirm that the ICMP Echo Request Leaves VTEP 1 Encapsulated and Goes to a UDP Destination Port on VTEP 2, on page 239
4. Verify ARP for Local Host Devices, on page 239
5. Verify that the MAC Address Entries are Learned in SISF Device Tracking Table, on page 239
6. Verify that EVPN Manager Has Been Updated with the MAC Address Entries, on page 240
7. Verify that EVPN Manager Has Updated the MAC Routes into Layer 2 RIB, on page 241
8. Verify that Layer 2 RIB Has Updated BGP with the Local MAC Routes, and that BGP Has Updated Layer 2 RIB with the Remote MAC Routes, on page 241
9. Verify that the MAC Routes Learned from BGP and Updated to Layer 2 RIB are Also Updated to L2FIB, on page 243



**Note** Only MAC routes are considered while verifying the movement of intra-subnet traffic. MAC-IP routes are not applicable to bridged traffic.

### Verify that the Local MAC Addresses Have Been Learned in IOS-MATM

The following examples show how to verify that the local MAC addresses have been learned in IOS-MATM:

```
VTEP-1# show mac address-table interface tw 1/0/1 vlan 10
      Mac Address Table
-----
Vlan    Mac Address      Type    Ports
----    -
10      005f.8602.10c6   DYNAMIC Tw1/0/1    <<- IOS-MATM shows only
                                           local MAC addresses
```

```
VTEP-2# show mac address-table interface g 2/0/1 vlan 10
      Mac Address Table
-----
Vlan    Mac Address      Type    Ports
----    -
10      008e.7391.1946   DYNAMIC Gi2/0/1
```

### Verify that Both Local and Remote MAC Addresses are Learned in FED-MATM

The following examples show how to verify that both local and remote MAC addresses are learned in FED-MATM:

```
VTEP-1# show platform software fed switch active matm macTable vlan 10
VLAN  MAC              Type  Seq#  EC_Bi  Flags  machandle
siHandle      riHandle          diHandle      *a_time  *e_time  ports
-----
10      005f.8602.10c6    0x1    60    0      0      0x7efcc0d78fc8    0x7efcc0ca8b88
          0x0              0x7efcc06cf9c8    300      144    TwoGigabitEthernet1/0/1

<<- Local MAC address is displayed here
10      008e.7391.1946    0x1000001  0      0      64    0x7efcc0cafb38    0x7efcc0d7f628
```

```

0x7ffa48c850b8      0x7efcc038cc18      0      144  RLOC 10.255.2.1 adj_id
135
<<- Remote MAC address is displayed here

VTEP-2#sh platform software fed switch active matm macTable vlan 10
VLAN  MAC              Type  Seq#  EC_Bi  Flags  machandle      siHandle
      riHandle          diHandle      *a_time  *e_time  ports
-----
10    005f.8602.10c6      0x1000001    0      0      64    0x7fcec4e977d8  0x7fcec4e93ae8
      0x7fcec4e93308      0x7fcec430a3d8    0      0      RLOC 10.255.1.1 adj_id
64
<<- Remote MAC address is displayed here
10    008e.7391.1946      0x1         46     0      0      0x7fcec4c6a248  0x7fcec4c20698
      0x0                  0x7fcec4611438    300    126    GigabitEthernet2/0/1

<<- Local MAC address is displayed here

```

**Confirm that the ICMP Echo Request Leaves VTEP 1 Encapsulated and Goes to a UDP Destination Port on VTEP 2**

The following image confirms that the ICMP echo request leaves VTEP 1 encapsulated and goes to a UDP destination port on VTEP 2 through the loopback interface Lo999 and the Layer 2 VNI 10001:

Figure 21:

→	1	0.000	10.10.10.11	10.10.10.12	ICMP	164	Echo (ping) request
←	2	0.000	10.10.10.12	10.10.10.11	ICMP	164	Echo (ping) reply

```

▶ Frame 1: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits) on interface 0
▶ Ethernet II, Src: 00:00:00:00:00:00, Dst: 00:00:00:00:00:00
▶ Internet Protocol Version 4, Src: 10.255.1.1, Dst: 10.255.1.2 ← Lo999 VTEP loopbacks
▶ User Datagram Protocol, Src Port: 65419 (65419), Dst Port: 4789 (4789)
▼ Virtual eXtensible Local Area Network
  ▶ Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 10001 ← L2 VNI 10001 Vlan 10
    Reserved: 0
  ▶ Ethernet II, Src: 00:5f:86:02:10:c6, Dst: 28:52:61:bf:a9:46 ← Native Source/Dest IP/MAC
  ▶ Internet Protocol Version 4, Src: 10.10.10.11, Dst: 10.10.10.12 ← Native Source/Dest IP/MAC
  ▶ Internet Control Message Protocol

```

**Verify ARP for Local Host Devices**

The following examples show how to verify ARP for local host devices:

```

VTEP-1# show ip arp vrf vxlan 10.10.10.11
Protocol Address          Age (min)  Hardware Addr  Type   Interface
Internet 10.10.10.11          2          005f.8602.10c6 ARPA   Vlan10

VTEP-2# show ip arp vrf vxlan 10.10.10.12
Protocol Address          Age (min)  Hardware Addr  Type   Interface
Internet 10.10.10.12          4          008e.7391.1946 ARPA   Vlan10

```

**Verify that the MAC Address Entries are Learned in SISF Device Tracking Table**

The following examples show how to verify that the MAC addresses are learned in SISF device tracking table:

```
VTEP-1# show device-tracking database mac <<- Only Local MAC addresses are seen
                                         in SISF device tracking table
MAC          Interface      vlan prlvl  state      time left policy
005f.8602.10c6 Tw1/0/1    10 NO TRUST MAC-REACHABLE 347 s    evpn-sisf-policy
<<- MAC, REACH, and EVPN type SISF policy are displayed
```

```
VTEP-2# show device-tracking database mac <<- Only Local MAC addresses are seen
                                         in SISF device tracking table
MAC          Interface      vlan prlvl  state      time left policy
008e.7391.1946 Gi2/0/1    10 NO TRUST MAC-REACHABLE 164 s    evpn-sisf-policy
<<- MAC, REACH, and EVPN type SISF policy are displayed
```

### Verify that EVPN Manager Has Been Updated with the MAC Address Entries

EVPN manager learns local MAC addresses and adds them to Layer 2 RIB. EVPN Manager also learns the remote MAC addresses from Layer 2 RIB, but the entries are only used for processing MAC mobility.

The following examples show how to verify that EVPN manager has been updated with the MAC addresses:

```
VTEP-1# show 12vpn evpn mac evi 10
MAC Address  EVI  VLAN  ESI                                     Ether Tag  Next Hop
-----
005f.8602.10c6 10   10    0000.0000.0000.0000.0000 0           Tw1/0/1:10
<<- MAC Addresss learned by EVPN Manager. States look correct
008e.7391.1946 10   10    0000.0000.0000.0000.0000 0           10.255.2.1
```

```
VTEP-1#sh 12vpn evpn mac evi 10 detail
MAC Address:          005f.8602.10c6      <<- Local MAC address
EVPN Instance:       10          <<- EVPN Instance
Vlan:                10          <<- VLAN
Ethernet Segment:    0000.0000.0000.0000.0000
Ethernet Tag ID:     0
Next Hop(s):         TwoGigabitEthernet1/0/1 service instance 10<<- Local interface
                                                             or local instance
VNI:                 10001      <<- VNI Label
Sequence Number:     0
MAC only present:    Yes
MAC Duplication Detection: Timer not running
```

```
MAC Address:          008e.7391.1946      <<- Remote MAC Address
EVPN Instance:       10          <<- EVPN Instance
Vlan:                10          <<- VLAN
Ethernet Segment:    0000.0000.0000.0000.0000
Ethernet Tag ID:     0
Next Hop(s):         10.255.2.1      <<- Remote VTEP-2 Tunnel Loopback
Local Address:       10.255.1.1      <<- Local VTEP-1 Tunnel Loopback
VNI:                 10001      <<- VNI Label
Sequence Number:     0
MAC only present:    Yes
MAC Duplication Detection: Timer not running
```

```
VTEP-2# show 12vpn evpn mac evi 10
MAC Address  EVI  VLAN  ESI                                     Ether Tag  Next Hop
-----
005f.8602.10c6 10   10    0000.0000.0000.0000.0000 0           10.255.1.1
008e.7391.1946 10   10    0000.0000.0000.0000.0000 0           Gi2/0/1:10
```



```

VTEP-2#sh l2vpn evpn mac evi 10 detail
MAC Address:          005f.8602.10c6      <<- Remote MAC address
EVPN Instance:       10                  <<- EVPN Instance
Vlan:                10                  <<- VLAN
Ethernet Segment:    0000.0000.0000.0000
Ethernet Tag ID:     0
Next Hop(s):         10.255.1.1         <<- Remote VTEP-1 Tunnel Loopback
Local Address:        10.255.2.1         <<- Local VTEP-2 Tunnel Loopback
VNI:                 10001              <<- VNI Label
Sequence Number:     0
MAC only present:    Yes
MAC Duplication Detection: Timer not running

MAC Address:          008e.7391.1946      <<- Remote MAC address
EVPN Instance:       10                  <<- EVPN Instance
Vlan:                10                  <<- VLAN
Ethernet Segment:    0000.0000.0000.0000
Ethernet Tag ID:     0
Next Hop(s):         GigabitEthernet2/0/1 service instance 10 <<- Local interface
                                                             or local instance
VNI:                 10001              <<- VNI Label
Sequence Number:     0
MAC only present:    Yes
MAC Duplication Detection: Timer not running
    
```

### Verify that EVPN Manager Has Updated the MAC Routes into Layer 2 RIB

Layer 2 RIB learns local MAC addresses from EVPN manager and updates BGP and Layer 2 FIB with them. Layer 2 RIB also learns remote MAC addresses from BGP and updates EVPN manager and Layer 2 FIB with them. Layer 2 RIB needs both local and remote MAC addresses in order to update BGP and Layer 2 FIB.

The following examples show how to verify that EVPN manager has updated the MAC routes into Layer 2 RIB:

```

VTEP-1# show l2route evpn mac
  EVI      ETag  Prod  Mac Address                Next Hop(s)  Seq Number
-----
  10       0    L2VPN 005f.8602.10c6            Tw1/0/1:10   0
  <<- Local prefix was added by EVPN Manager (Layer 2 VPN) into Layer 2 RIB
  10       0    BGP   008e.7391.1946            V:10001 10.255.2.1  0
  <<- Remote prefix was added by BGP into Layer 2 RIB

VTEP-2# show l2route evpn mac
  EVI      ETag  Prod  Mac Address                Next Hop(s)  Seq Number
-----
  10       0    BGP   005f.8602.10c6            V:10001 10.255.1.1   0
  <<- Remote prefix was added by BGP into Layer 2 RIB
  10       0    L2VPN 008e.7391.1946            Gi2/0/1:10   0
  <<- Local prefix was added by EVPN Manager (Layer 2 VPN) into Layer 2 RIB
    
```

### Verify that Layer 2 RIB Has Updated BGP with the Local MAC Routes, and that BGP Has Updated Layer 2 RIB with the Remote MAC Routes

The following examples show how to verify that Layer 2 RIB has updated BGP with the local MAC routes and that BGP has updated Layer 2 RIB with the remote MAC routes:

```

VTEP-1# show bgp l2vpn evpn route-type 2 0 005f860210c6 *
    
```

```

<<- Route-type is 2, Ethernet tag = 0, Local MAC address is in
undelimited format, and * specifies to omit IP address
BGP routing table entry for [2][10.1.1.1:10][0][48][005F860210C6][0][*]/20, version 249
Paths: (1 available, best #1, table evi_10) <<- Added to BGP from EVPN Manager
provisioning in l2vpn evi context

Advertised to update-groups:
 2
Refresh Epoch 1
Local
  :: (via default) from 0.0.0.0 (10.1.1.1) <<- Locally Advertised by VTEP-1,
  (: indicates local)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  EVPN ESI: 00000000000000000000, Label1 10001 <<- VNI ID is 10001 for VLAN 10
  Extended Community: RT:10:1 ENCAP:8 <<- RT 10:1 (local RT), Encap type 8 is VXLAN
  Local irb vxlan vtep:
    vrf:vxlan, 13-vni:99999
    local router mac:7035.0956.7EDD
    core-irb interface:Vlan99
    vtep-ip:10.255.1.1
    rx pathid: 0, tx pathid: 0x0

```

```

VTEP-1# show bgp l2vpn evpn route-type 2 0 008e73911946 *
<<- Route-type is 2, Ethernet tag = 0, Remote MAC address is in
undelimited format, and * specifies to omit IP address
BGP routing table entry for [2][10.1.1.1:10][0][48][008e73911946][0][*]/20, version 253
Paths: (1 available, best #1, table evi_10) <<- EVPN instance BGP table for VLAN 10
Not advertised to any peer
Refresh Epoch 1
Local, imported path from [2][10.2.2.2:10][0][48][008e73911946][0][*]/20 (global)
<<- From VTEP-2, RD is 10.2.2.2:10, MAC length is 48, [*] indicates MAC only
10.255.2.1 (metric 2) (via default) from 10.2.2.2 (10.2.2.2)
<<- Next hop of VTEP-2 Lo999, learned from RR 10.2.2.2
Origin incomplete, metric 0, localpref 100, valid, internal, best
EVPN ESI: 00000000000000000000, Label1 10001 <<- VNI ID 10001 for VLAN 10
Extended Community: RT:10:2 ENCAP:8 <<- Layer 2 VPN Route-Target 10:2
Encap type 8 is VXLAN
Originator: 10.2.2.2, Cluster list: 10.2.2.2
rx pathid: 0, tx pathid: 0x0

BGP routing table entry for [2][10.2.2.2:10][0][48][008e73911946][0][*]/20, version 251
Paths: (1 available, best #1, table EVPN-BGP-Table)
Not advertised to any peer
Refresh Epoch 1
Local
  10.255.2.1 (metric 2) (via default) from 10.2.2.2 (10.2.2.2)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  EVPN ESI: 00000000000000000000, Label1 10001
  Extended Community: RT:10:2 ENCAP:8
  Originator: 10.2.2.2, Cluster list: 10.2.2.2
  rx pathid: 0, tx pathid: 0x0

```

```

VTEP-2# show bgp l2vpn evpn route-type 2 0 008e73911946 *
<<- Route-type is 2, Ethernet tag = 0, Local MAC address is in
undelimited format, and * specifies to omit IP address
BGP routing table entry for [2][10.2.2.2:10][0][48][008e73911946][0][*]/20, version 292
Paths: (1 available, best #1, table evi_10)
Advertised to update-groups:
 2
Refresh Epoch 1
Local
  :: (via default) from 0.0.0.0 (10.2.2.2) <<- Locally Advertised by VTEP-2,
  (: indicates local)

```

```
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
EVPN ESI: 00000000000000000000, Label1 10001 <<- VNI ID 10001 for VLAN 10
Extended Community: RT:10:2 ENCAP:8 <<- RT 10:2 (local RT), Encap type 8 is VXLAN
Local irb vxlan vtep:
  vrf:vxlan, l3-vni:99999
  local router mac:7486.0BC4.B75D
  core-irb interface:Vlan99
  vtep-ip:10.255.2.1
  rx pathid: 0, tx pathid: 0x0
```

```
VTEP-2# show bgp l2vpn evpn route-type 2 0 005f860210c6 *
<<- Route-type is 2, Ethernet tag = 0, Remote MAC address is in
  undelimited format, and * specifies to omit IP address
BGP routing table entry for [2][10.1.1.1:10][0][48][005F860210C6][0][*]/20, version 312
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 7
  Local
    10.255.1.1 (metric 2) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10001
      Extended Community: RT:10:1 ENCAP:8
      Originator: 10.1.1.1, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0
```

```
BGP routing table entry for [2][10.2.2.2:10][0][48][005F860210C6][0][*]/20, version 314
Paths: (1 available, best #1, table evi_10) <<- EVPN instance BGP table for VLAN 10
  Not advertised to any peer
  Refresh Epoch 7
  Local, imported path from [2][10.1.1.1:10][0][48][005F860210C6][0][*]/20 (global)
    <<- From VTEP-2, RD is 10.2.2.2:10, MAC length is 48, [*] indicates MAC only
    <<- From VTEP-1, RD is 10.1.1.1:10, MAC length is 48, [*] indicates MAC only
    10.255.1.1 (metric 2) (via default) from 10.2.2.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Label1 10001 <<- VNI ID 10001 for VLAN 10
      Extended Community: RT:10:1 ENCAP:8 <<- Layer 2 VPN Route-Target 10:1
      Encap type 8 is VXLAN
      Originator: 10.1.1.1, Cluster list: 10.2.2.2
      rx pathid: 0, tx pathid: 0x0
```

### Verify that the MAC Routes Learned from BGP and Updated to Layer 2 RIB are Also Updated to L2FIB

The following examples show how to verify that the MAC routes that are learned from BGP and updated to Layer 2 RIB are also updated to Layer 2 FIB:

```
VTEP-2# show l2fib bridge-domain 10 detail
Bridge Domain : 10
Reference Count : 15
Replication ports count : 2
Unicast Address table size : 4
IP Multicast Prefix table size : 3

Flood List Information :
  Olist: 5109, Ports: 2

VxLAN Information :
  VXLAN_DEC nv1:10001:239.10.10.10

Port Information :
  BD_PORT  Gi2/0/1:10
```

```

VXLAN_REP nv1:10001:239.10.10.10

Unicast Address table information :
 005f.8602.10c6 VXLAN_CP L:10001:10.255.2.1 R:10001:10.255.1.1
<<- Remote MAC address is learned (local MAC address is not expected to be present)

IP Multicast Prefix table information :
 Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 5109, Ports: 2
 Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 5109, Ports: 2
 Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 5109, Ports: 2

VTEP-1# show 12fib bridge-domain 10 detail
Bridge Domain : 10
Reference Count : 14
Replication ports count : 2
Unicast Address table size : 3
IP Multicast Prefix table size : 3

Flood List Information :
 Olist: 5109, Ports: 2

VxLAN Information :
 VXLAN_DEC nv1:10001:239.10.10.10

Port Information :
 BD_PORT Tw1/0/1:10
 VXLAN_REP nv1:10001:239.10.10.10

Unicast Address table information :
 008e.7391.1946 VXLAN_CP L:10001:10.255.1.1 R:10001:10.255.2.1
<<- Remote MAC address is learned (local MAC address is not expected to be present)

IP Multicast Prefix table information :
 Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 5109, Ports: 2
 Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 5109, Ports: 2
 Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 5109, Ports: 2
    
```



**Note** Only remote MAC routes are displayed in the output.

## Troubleshooting Unicast Forwarding Between VTEPS in Different VLANs Through a Layer 3 VNI

This scenario might occur when host device 1 in VLAN 12 attempts to ping host device 4 in VLAN 13. Perform the checks listed in the following table before troubleshooting unicast forwarding between VTEPs in different VLANs through a Layer 3 VNI:

**Table 23: Scenario 3: Troubleshooting Unicast Forwarding Between VTEPS in Different VLANs Through a Layer 3 VNI**

Check to be Performed	Steps to Follow
Are the source and destination host devices in different subnets?	Check the subnet of the local host device and compare it against the subnet of the remote host device.

Check to be Performed	Steps to Follow
Do you have an SVI interface configured for the remote subnet?	Run the <b>show ip interface brief   exclude unassigned</b> command in privileged EXEC mode on the VTEP.
Do you have the EVPN instance configured on your local VTEP?	Run the following commands in privileged EXEC mode on the VTEP: <ul style="list-style-type: none"> <li>• <b>show run   section l2vpn</b></li> <li>• <b>show run   section vlan config</b></li> <li>• <b>show run interface nve interface-number</b></li> </ul>

To troubleshoot unicast forwarding between two VTEPs in different VLANs using a Layer 3 VNI, follow these steps:

- Verify the provisioning of the EVPN VXLAN Layer 3 overlay network.
- Verify inter-subnet traffic movement and symmetric IRB in the EVPN VXLAN Layer 3 overlay network.

## Verifying the Provisioning of an EVPN VXLAN Layer 3 Overlay Network

To verify the provisioning of an EVPN VXLAN Layer 3 overlay network, perform these checks:

1. [Verify that the Access SVIs, Core SVIs, and NVE Interfaces are Up, on page 245](#)
2. [Verify that the IP VRF is Provisioned with the Correct SVIs, Stitching Route-Targets, and Route Distinguisher, on page 246](#)
3. [Verify that Both Layer 2 and Layer 3 VNIs are provisioned in the VRF and are UP, on page 247](#)
4. [Verify that EVPN Manager is Updated from the NVE with all the Layer 2 and IRB Attributes, on page 248](#)
5. [Verify that the Remote Layer 3 VNI Details are Learned on Each VTEP, on page 249](#)
6. [Verify that the Layer 3 VNI Tunnel Pseudopoint is Installed into Layer 2 FIB in the Core VLAN, on page 249](#)

### Verify that the Access SVIs, Core SVIs, and NVE Interfaces are Up

The following examples show how to verify that the access SVIs, core SVIs, and NVE interfaces are up:

```
VTEP-1# show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
Vlan10             10.10.10.1     YES NVRAM   up          up
Vlan12             10.12.12.1     YES NVRAM   up          up    <<< Access Interface
Vlan99             10.255.1.1     YES unset   up          up    <<< Core Interface
<<< If protocol status for the core interface is down, run the no autostate command
Loopback0          10.1.1.1       YES NVRAM   up          up
Loopback999        10.255.1.1     YES NVRAM   up          up
Tunnel0            10.255.1.1     YES unset   up          up
```

```

Tunnel1          10.1.1.5          YES unset up          up
nve1             unassigned       YES unset up          up

VTEP-2# show ip interface brief
Interface        IP-Address      OK? Method Status      Protocol
Vlan10           10.10.10.1     YES NVRAM  up          up
Vlan13           10.13.13.1     YES NVRAM  up          up    <<- Access Interface
Vlan99           10.255.2.1     YES unset  up          up    <<- Core Interface
    <<- If protocol status for the core interface is down, run the no autostate command
Loopback0        10.2.2.2       YES NVRAM  up          up
Loopback999     10.255.2.1     YES NVRAM  up          up
Tunnel0          10.255.2.1     YES unset  up          up
Tunnel1          10.1.1.10      YES unset  up          up

```

### Verify that the IP VRF is Provisioned with the Correct SVIs, Stitching Route-Targets, and Route Distinguisher

The following examples show how to verify that the IP VRF is provisioned with the correct SVIs, stitching route-targets, and route distinguisher:

```

VTEP-1# show run vrf vxlan    <<- vxlan is the name of the VRF
vrf definition vxlan
rd 10.255.1.1:1
!
address-family ipv4
  route-target export 10.255.1.1:1 stitching    <<- Exporting local route-target
  route-target import 10.255.2.1:1 stitching    <<- Importing VTEP-2 route-target

```

```

VTEP-1# show ip vrf vxlan    <<- vxlan is the name of the VRF
Name                        Default RD                Interfaces
vxlan                       10.255.1.1:1             V110
                             V112
                             V199

```

```

VTEP-1# show ip vrf detail vxlan    <<- vxlan is the name of the VRF
VRF vxlan (VRF Id = 2); default RD 10.255.1.1:1; default VPNID <not set>
New CLI format, supports multiple address-families
Flags: 0x180C
Interfaces:
V110 V112 V199
Address family ipv4 unicast (Table ID = 0x2):    <<- Table 2 maps to VRF vxlan,
                                                also found in BPG VPNv4 table

Flags: 0x0
No Export VPN route-target communities
No Import VPN route-target communities
Export VPN route-target stitching communities
    <<- VRF is using stitching route-targets. VTEPs must
    import each other's targets (same as Layer 3 VPN)
RT:10.255.1.1:1
Import VPN route-target stitching communities
RT:10.255.2.1:1
No import route-map
No global export route-map
No export route-map
VRF label distribution protocol: not configured
VRF label allocation mode: per-prefix

```

```

VTEP-2# show ip vrf vxlan    <<- vxlan is the name of the VRF
Name                        Default RD                Interfaces

```

```
vxlan                                10.255.2.1:1                V110
                                      V113
                                      V199
```

```
VTEP-2# show ip vrf detail vxlan    <<- vxlan is the name of the VRF
VRF vxlan (VRF Id = 2); default RD 10.255.2.1:1; default VPNID <not set>
New CLI format, supports multiple address-families
Flags: 0x180C
Interfaces:
V110 V113 V199
Address family ipv4 unicast (Table ID = 0x2):    <<- Table 2 maps to VRF vxlan,
                                                  also found in BPG VPNv4 table

Flags: 0x0
No Export VPN route-target communities
No Import VPN route-target communities
Export VPN route-target stitching communities
<<- VRF is using stitching route-targets. VTEPs must
import each other's targets (same as Layer 3 VPN)
RT:10.255.2.1:1
Import VPN route-target stitching communities
RT:10.255.1.1:1
No import route-map
No global export route-map
No export route-map
VRF label distribution protocol: not configured
VRF label allocation mode: per-prefix
```

### Verify that Both Layer 2 and Layer 3 VNIs are provisioned in the VRF and are UP

The following examples show how to verify that both Layer 2 and Layer 3 VNIs are provisioned in the VRF and are up:

```
VTEP-1# show run | section vlan config
vlan configuration 99    <<- VNI is a member of VRF vxlan, not of EVPN instance
member vni99999
```

```
VTEP-1# show run interface vlan 99
interface Vlan99
description connected to L3_VNI_99999
vrf forwarding vxlan
ip unnumbered Loopback999
```

```
VTEP-1# show run interface nve 1
no ip address
source-interface Loopback999
host-reachability protocol bgp
member vni 99999 vrf vxlan    <<- VNI tied to the VRF under NVE interface
member vni 12012 mcast-group 239.12.12.12 <<- VNI tied to the NVE
```

```
VTEP-1# show run | section l2vpn
l2vpn evpn instance 12 vlan-based
encapsulation vxlan
route-target export 12:1    <<- Remote VTEP is NOT importing this route target,
as it does not have the VLAN or VNI on its end
route-target import 12:1
no auto-route-target
```

```
VTEP-1# show run | section vlan config
```

```
vlan configuration 12
  member evpn-instance 12 vni 12012 <<- EVPN instance or VNI associated to the VLAN
```

```
VTEP-1# show nve vni
Interface VNI Multicast-group VNI state Mode VLAN cfg vrf
nve1 10001 239.10.10.10 Up L2CP 10 CLI vxlan
nve1 12012 239.12.12.12 Up L2CP 12 CLI vxlan <<- Layer 2 VNI
nve1 99999 N/A Up L3CP 99 CLI vxlan <<- Layer 3 VNI
```

```
VTEP-2# show nve vni
Interface VNI Multicast-group VNI state Mode VLAN cfg vrf
nve1 13013 239.13.13.13 Up L2CP 13 CLI vxlan <<- Layer 2 VNI
nve1 10001 239.10.10.10 Up L2CP 10 CLI vxlan
nve1 99999 N/A Up L3CP 99 CLI vxlan <<- Layer 3 VNI
```

### Verify that EVPN Manager is Updated from the NVE with all the Layer 2 and IRB Attributes

The following examples show how to verify that EVPN manager is updated from the NVE with all the Layer 2 and IRB attributes:

```
VTEP-1# show l2vpn evpn evi
EVI VLAN Ether Tag L2 VNI Multicast Pseudopoint
-----
12 12 0 12012 239.12.12.12 Tw1/0/1:12
<<- See which EVPN instance maps to the VLAN. The VLAN
or EVPN instance values are not always the same
<...snip...>
```

```
VTEP-1# show l2vpn evpn evi 12 detail
EVPN instance: 12 (VLAN Based)
RD: 10.1.1.1:12 (auto)
Import-RTs: 12:1
Export-RTs: 12:1
Per-EVI Label: none
State: Established
Encapsulation: vxlan
Vlan: 12 <<- VLAN Layer 2 VNI
Ethernet-Tag: 0
State: Established
Core If: Vlan99 <<- Interface handling IP VRF forwarding
Access If: Vlan12
NVE If: nve1
RMAC: 7035.0956.7edd <<- RMAC is the BIA of SVI 99 Core interface
Core Vlan: 99
L2 VNI: 12012
L3 VNI: 99999
VTEP IP: 10.255.1.1 <<- Local Tunnel endpoint IP address
MCAST IP: 239.12.12.12
VRF: vxlan <<- IP VRF for Layer 3 VPN
Pseudoports:
  TwoGigabitEthernet1/0/1 service instance 12
```

```
VTEP-2# show l2vpn evpn evi
EVI VLAN Ether Tag L2 VNI Multicast Pseudopoint
-----
13 13 0 13013 239.13.13.13 Gi2/0/1:13
<<- See which EVPN instance maps to the VLAN. The VLAN
or EVPN instance values are not always the same
```



```
VTEP-2# show l2vpn evpn evi 13 detail
EVPN instance: 13 (VLAN Based)
RD: 10.2.2.2:13 (auto)
Import-RTs: 13:2
Export-RTs: 13:2
Per-EVI Label: none
State: Established
Encapsulation: vxlan
Vlan: 13 <<- VLAN Layer 2 VNI
  Ethernet-Tag: 0
  State: Established
  Core If: Vlan99 <<- Interface handling IP VRF forwarding
  Access If: Vlan13
  NVE If: nve1
  RMAC: 7486.0bc4.b75d <<- RMAC is the BIA of SVI 99 Core interface
  Core Vlan: 99
  L2 VNI: 13013
  L3 VNI: 99999
  VTEP IP: 10.255.2.1 <<- Local Tunnel endpoint IP address
  MCAST IP: 239.13.13.13
  VRF: vxlan <<- IP VRF for Layer 3 VPN
Pseudoports:
  GigabitEthernet2/0/1 service instance 13
```

### Verify that the Remote Layer 3 VNI Details are Learned on Each VTEP

The following examples show how to verify that the remote Layer 3 VNI details are learned on each VTEP:

```
VTEP-1# show nve peers
Interface VNI Type Peer-IP RMAC/Num_RT eVNI state flags UP time
nve1 99999 L3CP 10.255.2.1 7486.0bc4.b75d 99999 UP A/M 1w1d
<<- Layer 3 Control Plane (L3CP), RMAC of Remote VTEP and Uptime of peer are displayed
```

```
VTEP-2# show nve peers
Interface VNI Type Peer-IP RMAC/Num_RT eVNI state flags UP time
nve1 99999 L3CP 10.255.1.1 7035.0956.7edd 99999 UP A/M 21:27:36
<<- Layer 3 Control Plane (L3CP), RMAC of Remote VTEP and Uptime of peer are displayed
```

### Verify that the Layer 3 VNI Tunnel Pseudoport is Installed into Layer 2 FIB in the Core VLAN

The following examples show how to verify that the Layer 3 VNI tunnel pseudoport is installed into Layer 2 FIB in the core VLAN:

```
VTEP-1# show l2fib bridge-domain 99 detail
<<- The Core VLAN can be obtained in the output of the
show l2vpn evpn evi <evpn-instance> detail command
Bridge Domain : 99
Reference Count : 8
Replication ports count : 0
Unicast Address table size : 1
IP Multicast Prefix table size : 3

Flood List Information :
Olist: 5112, Ports: 0
```

```

VxLAN Information :

Unicast Address table information :
 7486.0bc4.b75d VXLAN_CP L:99999:10.255.1.1 R:99999:10.255.2.1
<<- Encapsulation Information to reach remote VTEP-2

IP Multicast Prefix table information :
 Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 5112, Ports: 0
 Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 5112, Ports: 0
 Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 5112, Ports: 0

```

```

VTEP-2# show l2fib bridge-domain 99 detail
<<- The Core VLAN can be obtained in the output of the
show l2vpn evpn evi <evpn-instance> detail command

```

```

Bridge Domain : 99
Reference Count : 8
Replication ports count : 0
Unicast Address table size : 1
IP Multicast Prefix table size : 3

Flood List Information :
 Olist: 5111, Ports: 0

VxLAN Information :

Unicast Address table information :
 7035.0956.7edd VXLAN_CP L:99999:10.255.2.1 R:99999:10.255.1.1
<<- Encapsulation Information to reach remote VTEP-2

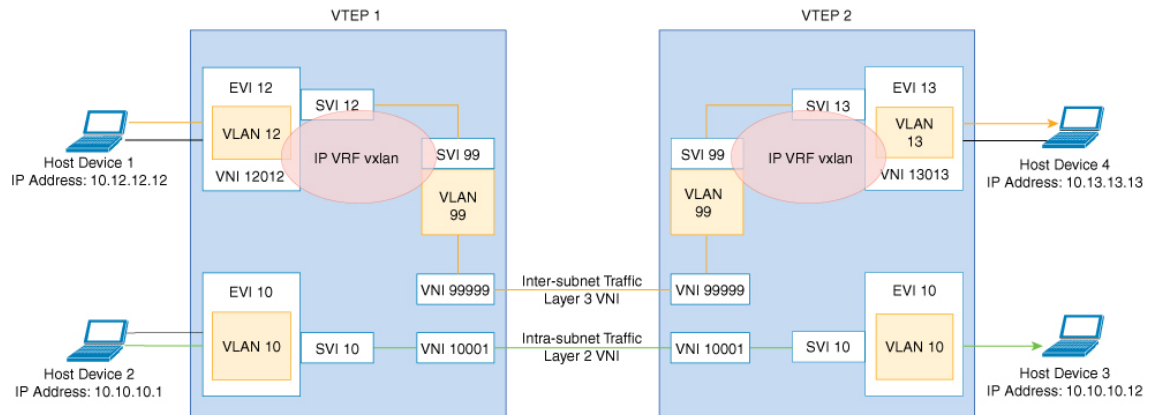
IP Multicast Prefix table information :
 Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 5111, Ports: 0
 Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 5111, Ports: 0
 Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 5111, Ports: 0

```

## Verifying Inter-Subnet Traffic Movement and Symmetric IRB in an EVPN VXLAN Layer 3 Overlay Network

The following figure illustrates the movement of traffic from host devices connected to VTEP 1 to host devices connected to VTEP 2:

Figure 22: Movement of traffic in an EVPN VXLAN network through Layer2 and Layer 3 VNIs



In the above figure, Layer 3 traffic moves from host device 1 to host device 4 through the Layer 3 VNI 99999. To verify the movement of inter-subnet traffic in the EVPN VXLAN Layer 3 overlay network, perform these checks:

1. Verify that Local MAC Address and IP Address Entries are Learned in SISF Device Tracking Table, on page 251
2. Verify that MAC Address and IP Address Entries are Learned in EVPN Manager, on page 252
3. Verify that MAC Address and IP Address Entries are Learned in Layer 2 RIB, on page 253
4. Verify that Local MAC Address and IP Address Entries are Learned in MAC VRF, on page 253
5. Verify that Remote MAC-IP Address Pair is Learend in the VRF, on page 254
6. Verify that IP Routes are Inserted in RIB, on page 255
7. Verify that the Adjacency Table Contains Entries for the VRF-Enabled Core VLAN Interface, on page 255
8. Confirm that Adjacency Exists to the VTEP Tunnel IP Address for a Host Device in IP VRF, on page 256
9. Confirm that Adjacency Exists to Reach Tunnel Destination, on page 256
10. Confirm that the ICMP Echo Request that Leaves Encapsulated from the Source VTEP Reaches the Loopback Tunnel Endpoint and UDP Destination Port on the Destination VTEP Through the Layer 3 VNI and IP VRF, on page 256

**Verify that Local MAC Address and IP Address Entries are Learned in SISF Device Tracking Table**

The following examples show how to verify that local MAC address and IP address entries are learned in SISF device tracking table:

```
VTEP-1# show device-tracking database vlanid 12
Binding Table has 4 entries, 2 dynamic (limit 100000)
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
```

```
Preflevel flags (prlvl):
0001:MAC and LLA match      0002:Orig trunk          0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access 0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated  0100:Statically assigned
```

```

Network Layer Address      Link Layer Address Interface      vlan prlvl  age
state      Time left
ARP 10.12.12.12          005f.8602.10e7  Tw1/0/1          12  0005  115s
REACHABLE  N/A
```

VTEP-2# **show device-tracking database vlanid 13**

```
vlanDB has 2 entries for vlan 13, 1 dynamic
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
```

```
Preflevel flags (prlvl):
0001:MAC and LLA match      0002:Orig trunk          0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access 0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated  0100:Statically assigned
```

```

Network Layer Address      Link Layer Address Interface      vlan prlvl  age
state      Time left
ARP 10.13.13.13          008e.7391.1977  Gi2/0/1          13  0005  155s
REACHABLE  N/A
```

### Verify that MAC Address and IP Address Entries are Learned in EVPN Manager

The following examples show how to verify that MAC address and IP address entries are learned in EVPN manager:

VTEP-1# **show l2vpn evpn mac ip evi 12**

```
IP Address      EVI    VLAN  MAC Address      Next Hop
-----
10.12.12.12    12     12    005f.8602.10e7  Tw1/0/1:12
```

VTEP-1#sh l2vpn evpn mac ip evi 12 detail

```
IP Address:      10.12.12.12
EVPN Instance:   12
Vlan:            12
MAC Address:     005f.8602.10e7
Ethernet Segment: 0000.0000.0000.0000.0000
Ethernet Tag ID: 0
Next Hop:        TwoGigabitEthernet1/0/1 service instance 12
VNI:             12012
Sequence Number: 0
IP Duplication Detection: Timer not running
```

VTEP-2# **show l2vpn evpn mac ip evi 13**

```
IP Address      EVI    VLAN  MAC Address      Next Hop
-----
10.13.13.13    13     13    008e.7391.1977  Gi2/0/1:13
```

VTEP-2#sh l2vpn evpn mac ip evi 13 detail

```
IP Address:      10.13.13.13
EVPN Instance:   13
Vlan:            13
MAC Address:     008e.7391.1977
Ethernet Segment: 0000.0000.0000.0000.0000
```

```
Ethernet Tag ID:          0
Next Hop:                 GigabitEthernet2/0/1 service instance 13
VNI:                     13013
Sequence Number:         0
IP Duplication Detection: Timer not running
```

### Verify that MAC Address and IP Address Entries are Learned in Layer 2 RIB

The following examples show how to verify that MAC address and IP address entries are learned in Layer 2 RIB:

```
VTEP-1# show l2route evpn mac ip
  EVI      ETag  Prod   Mac Address           Host IP           Next Hop(s)
-----
  12             0 L2VPN 005f.8602.10e7       10.12.12.12      Tw1/0/1:12

VTEP-2# show l2route evpn mac ip
  EVI      ETag  Prod   Mac Address           Host IP           Next Hop(s)
-----
  13             0 L2VPN 008e.7391.1977       10.13.13.13      Gi2/0/1:13
```

### Verify that Local MAC Address and IP Address Entries are Learned in MAC VRF

```
VTEP-1# show bgp l2vpn evpn evi 12 route-type 2 0 005F860210E7 10.12.12.12
BGP routing table entry for [2][10.1.1.1:12][0][48][005F860210E7][32][10.12.12.12]/24,
version 72
Paths: (1 available, best #1, table evi_12) <<- The Layer 2 VPN table number
                                                for EVPN instance 12

  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local <<- Indicates locally learned route
  :: (via default) from 0.0.0.0 (10.1.1.1)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  EVPN ESI: 00000000000000000000, Label1 12012, Label2 99999 <<- Displays both Layer 2
                                                                and VRF labels
  Extended Community: RT:12:1 RT:10.255.1.1:1 ENCAP:8 <<- Note the VRF stitching RT
                                                                as well as the Layer 2 RT

  Router MAC:7035.0956.7EDD
  Local irb vxlan vtep:
    vrf:vxlan, 13-vni:99999
    local router mac:7035.0956.7EDD <<- Local RMAC
    core-irb interface:Vlan99 <<- VRF Layer 3 VPN interface
    vtep-ip:10.255.1.1 <<- Loopback 999 tunnel endpoint
    rx pathid: 0, tx pathid: 0x0
```

The following examples show how to verify that local MAC address and IP address entries are learned in MAC VRF:

```
VTEP-2# show bgp l2vpn evpn evi 13 route-type 2 0 008E73911977 10.13.13.13
BGP routing table entry for [2][10.2.2.2:13][0][48][008E73911977][32][10.13.13.13]/24,
version 70
Paths: (1 available, best #1, table evi_13)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local <<- Indicates locally learned route
```

```

:: (via default) from 0.0.0.0 (10.2.2.2)
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
EVPN ESI: 00000000000000000000, Label1 13013, Label2 99999
Extended Community: RT:13:1 RT:10.255.2.1:1 ENCAP:8
Router MAC:7486.0BC4.B75D
Local irb vxlan vtep:
vrf:vxlan, l3-vni:99999
local router mac:7486.0BC4.B75D
core-irb interface:Vlan99
vtep-ip:10.255.2.1
rx pathid: 0, tx pathid: 0x0

```

### Verify that Remote MAC-IP Address Pair is Learned in the VRF

The following examples verify that remote MAC-IP address pair is learned in the VRF:

```

VTEP-1# show bgp vpnv4 unicast vrf vxlan 10.13.13.13
BGP routing table entry for 10.255.1.1:1:10.13.13.13/32, version 15
Paths: (1 available, best #1, table vxlan) <<- VPNv4 VRF BGP table
Not advertised to any peer
Refresh Epoch 2
Local, imported path from [2][10.2.2.2:13][0][48][008E73911977][32][10.13.13.13]/24
(global)
<<- EVPN type-2, l2vpn RD 10.2.2.2:13, MAC and IP addresses
10.255.2.1 (metric 3) (via default) from 10.2.2.2 (10.2.2.2)
<<- Next hop 10.255.2.1, learned from RR 10.2.2.2
Origin incomplete, metric 0, localpref 100, valid, internal, best
Extended Community: ENCAP:8 Router MAC:7486.0BC4.B75D
Originator: 10.2.2.2, Cluster list: 10.2.2.2
Local vxlan vtep:
vrf:vxlan, vni:99999
local router mac:7035.0956.7EDD
encap:8
vtep-ip:10.255.1.1
bdi:Vlan99
Remote VxLAN:
Topoid 0x2(vrf vxlan) <<- VRF vxlan (mapped to ID 2)
Remote Router MAC:7486.0BC4.B75D <<- VTEP-2 RMAC
Encap 8 <<- VXLAN encap (type 8)
Egress VNI 99999 <<- VRF VNI
RTEP 10.255.2.1 <<- VTEP-2 Remote Tunnel Endpoint
rx pathid: 0, tx pathid: 0x0

```

```

VTEP-2# show bgp vpnv4 unicast vrf vxlan 10.12.12.12
BGP routing table entry for 10.255.2.1:1:10.12.12.12/32, version 15
Paths: (1 available, best #1, table vxlan)
Not advertised to any peer
Refresh Epoch 2
Local, imported path from [2][10.1.1.1:12][0][48][005F860210E7][32][10.12.12.12]/24
(global)
<<- EVPN type-2, l2vpn RD 10.1.1.1:12, MAC and IP addresses
10.255.1.1 (metric 3) (via default) from 10.2.2.2 (10.2.2.2)
<<- Next hop 10.255.1.1, learned from RR 10.2.2.2
Origin incomplete, metric 0, localpref 100, valid, internal, best
Extended Community: ENCAP:8 Router MAC:7035.0956.7EDD
Originator: 10.1.1.1, Cluster list: 10.2.2.2
Local vxlan vtep:
vrf:vxlan, vni:99999
local router mac:7486.0BC4.B75D
encap:8
vtep-ip:10.255.2.1

```

```

bdi:Vlan99
Remote VxLAN:
  Topoid 0x2(vrf vxlan)    <<- VRF vxlan (mapped to ID 2)
  Remote Router MAC:7035.0956.7EDD <<- VTEP-1 RMAC
  Encap 8                  <<- VXLAN encap (type 8)
  Egress VNI 99999        <<- VRF VNI
  RTEP 10.255.1.1         <<- VTEP-2 Remote Tunnel Endpoint
  rx pathid: 0, tx pathid: 0x0
  
```

### Verify that IP Routes are Inserted in RIB

The following examples show how to verify that IP routes are inserted in RIB:

```
VTEP-1# show ip route vrf vxlan 10.13.13.13
```

```

Routing Table: vxlan
Routing entry for 10.13.13.13/32
  Known via "bgp 69420", distance 200, metric 0, type internal
  Last update from 10.255.2.1 on Vlan99, 00:11:33 ago
  Routing Descriptor Blocks:
  * 10.255.2.1 (default), from 10.2.2.2, 00:11:33 ago, via Vlan99 <<- Next hop here is the
  Core VLAN interface

  Route metric is 0, traffic share count is 1
  AS Hops 0
  MPLS label: none
  
```

```
VTEP-2# show ip route vrf vxlan 10.12.12.12
```

```

Routing Table: vxlan
Routing entry for 10.12.12.12/32
  Known via "bgp 69420", distance 200, metric 0, type internal
  Last update from 10.255.1.1 on Vlan99, 00:04:06 ago
  Routing Descriptor Blocks:
  * 10.255.1.1 (default), from 10.2.2.2, 00:04:06 ago, via Vlan99 <<- Next hop here is the
  Core VLAN interface

  Route metric is 0, traffic share count is 1
  AS Hops 0
  MPLS label: none
  
```

### Verify that the Adjacency Table Contains Entries for the VRF-Enabled Core VLAN Interface

The following examples show how to verify that the adjacency table contains entries for the VRF-enabled core VLAN interface:

```
VTEP-1# show adjacency vlan 99 detail
```

```

Protocol Interface          Address
IP          Vlan99          10.255.2.1(9) <<- IP unnumbered from Loopback 999
                                0 packets, 0 bytes
                                epoch 0
                                sourced in sev-epoch 6
                                Encap length 14
                                74860BC4B75D703509567EDD0800
<<- Local RMAC is 74860BC4B75D, Remote RMAC is 703509567EDD, etype is 800
                                VXLAN Transport tunnel
<<- Tunnel Interface (RMAC, using VTEP Loopback IP address)
  
```

```
VTEP-2# show adjacency vlan 99 detail
```

```

Protocol Interface Address
IP Vlan99 10.255.1.1(9) <<- IP unnumbered from Loopback 999
0 packets, 0 bytes
epoch 0
sourced in sev-epoch 5
Encap length 14
703509567EDD74860BC4B75D0800
<<- Local RMAC is 703509567EDD, Remote RMAC is 74860BC4B75D, etype is 800
VXLAN Transport tunnel
<<- Tunnel Interface (RMAC, using VTEP Loopback IP address)
    
```

### Confirm that Adjacency Exists to the VTEP Tunnel IP Address for a Host Device in IP VRF

The following example shows how to confirm that adjacency exists to the VTEP Tunnel IP address for a host device in IP VRF:

```

VTEP-1# show ip cef vrf vxlan 10.13.13.13/32 <<- Remote host in VLAN 13 of VTEP-2
10.13.13.13/32
nexthop 10.255.2.1 Vlan99
    
```

### Confirm that Adjacency Exists to Reach Tunnel Destination

The following example shows how to confirm that adjacency exists to reach tunnel destination:

```

VTEP-1# show ip cef 10.255.1.11
10.255.2.1/32
nexthop 10.1.1.6 TwoGigabitEthernet1/0/2
    
```

### Confirm that the ICMP Echo Request that Leaves Encapsulated from the Source VTEP Reaches the Loopback Tunnel Endpoint and UDP Destination Port on the Destination VTEP Through the Layer 3 VNI and IP VRF

The following image confirms that the ICMP echo request that leaves encapsulated from source VTEP reaches the Loopback interface and UDP destination port on the destination VTEP through the Layer 3 VNI and IP VRF:

→	3	0.000	10.12.12.12	10.13.13.13	ICMP	164	Echo (ping) request
←	4	0.000	10.13.13.13	10.12.12.12	ICMP	164	Echo (ping) reply
	5	0.000	10.12.12.12	10.13.13.13	ICMP	164	Echo (ping) request
	6	0.000	10.13.13.13	10.12.12.12	ICMP	164	Echo (ping) reply

```

▶ Frame 3: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits) on interface 0
▶ Ethernet II, Src: 00:00:00:00:00:00, Dst: 00:00:00:00:00:00
▶ Internet Protocol Version 4, Src: 10.255.1.1, Dst: 10.255.2.1 ← Tunnel Endpoint IPs
▶ User Datagram Protocol, Src Port: 65478 (65478), Dst Port: 4789 (4789)
▼ Virtual eXtensible Local Area Network
  ▶ Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 9999 ← L3 VNI 9999 VRF vxlan
    Reserved: 0
  ▶ Ethernet II, Src: 00:01:00:01:00:00, Dst: 74:86:0b:c4:b7:5d ← VTEP-2 Dst: RMAC
  ▶ Internet Protocol Version 4, Src: 10.12.12.12, Dst: 10.13.13.13
  ▶ Internet Control Message Protocol
    
```



# Troubleshooting Unicast Forwarding Between a VXLAN Network and an IP Network

This scenario might occur when host device 1 attempts to ping an external IP address through a border leaf VTEP. Perform the checks listed in the following table before troubleshooting unicast forwarding between a VXLAN network and an external IP network.

**Table 24: Scenario 4: Troubleshooting Unicast Forwarding Between a VXLAN Network and an IP Network**

Check to be performed	Steps to follow
Is one IP address present in the VXLAN network and the other IP address coming from external IP network?	<p>Check the local subnets (or the SVI interfaces) if the remote subnet is present.</p> <p><b>Note</b> Local subnet has the remote subnet listed even in the case of scenario 3.</p>
Is the EVPN route type 5 being used to send traffic to remote destination?	<p>Run the <b>show bgp l2vpn evpn all</b> command in privileged EXEC mode on the VTEP. Look for remote prefix to be displayed as [ 5 ] for route type 5.</p>

To troubleshoot unicast forwarding between a VXLAN network and an external IP network, follow these steps:

- Verify the provisioning of the EVPN VXLAN Layer 3 overlay network.
- Verify traffic movement from the VXLAN network to the IP network through the border leaf switch using route type 5.

## Verifying the Provisioning of an EVPN VXLAN Layer 3 Overlay Network

See [Verifying the Provisioning of an EVPN VXLAN Layer 3 Overlay Network, on page 245](#) for detailed steps.

## Verifying Traffic from a VXLAN Fabric to an IP Network Through a Border Leaf Switch Using Route Type 5

To verify the movement of traffic from a VXLAN fabric to an external IP network through a border leaf switch, perform these checks:

1. [Check the Table Entries for BGP, EVPN, and VPNv4 Tables, on page 257](#)
2. [Check the Table Entries for BGP, EVPN, and VPNv4 Tables, on page 257](#)
3. [Confirm that Adjacency exists to Reach Tunnel Destination, on page 260](#)

### Check the Table Entries for BGP, EVPN, and VPNv4 Tables

The following examples show how to check the table entries for BGP, EVPN and VPNv4 tables:

```

VTEP-1# show bgp vpnv4 unicast vrf vxlan 10.9.9.9/32
<<- To a remote IP address outside the VXLAN fabric
BGP routing table entry for 10.255.1.1:1:10.9.9.9/32, version 150
Paths: (1 available, best #1, table vxlan) <<- VPNv4 VRF BGP table
Not advertised to any peer
Refresh Epoch 2
Local, imported path from [5][10.255.1.11:1][0][32][10.9.9.9]/17 (global)
<<- Learned from EVPN into VPNv4
  10.255.1.11 (metric 3) (via default) from 10.2.2.2 (10.2.2.2)
  Origin IGP, metric 0, localpref 100, valid, internal, best
  Extended Community: ENCAP:8 Router MAC:EC1D.8B55.F55D
  Originator: 10.255.1.11, Cluster list: 10.2.2.2
  Local vxlan vtep:
    vrf:vxlan, vni:99999
    local router mac:7035.0956.7EDD
    encap:8
    vtep-ip:10.255.1.1
    bdi:Vlan99
  Remote VxLAN:
    Topoid 0x2(vrf vxlan)
    Remote Router MAC:EC1D.8B55.F55D <<- Border_Leaf_VTEP RMAC
    Encap 8
    Egress VNI 99999 <<- VNI associated with VRF
    RTEP 10.255.1.11 <<- Tunnel IP address
    rx pathid: 0, tx pathid: 0x0

VTEP-1# show bgp l2vpn evpn all route-type 5 0 10.9.9.9 32
<<- This is sent as type 5 as there is no VNI at all for it to be mapped to
BGP routing table entry for [5][10.255.1.11:1][0][32][10.9.9.9]/17, version 650
Paths: (1 available, best #1, table EVPN-BGP-Table)
Not advertised to any peer
Refresh Epoch 2
Local
  10.255.1.11 (metric 3) (via default) from 10.2.2.2 (10.2.2.2)
  <<- Border_Leaf_VTEP Tunnel IP address
  Origin IGP, metric 0, localpref 100, valid, internal, best
  EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, VNI Label 99999, MPLS VPN
  Label 0
  <<- Using Layer 3 VNI 99999
  Extended Community: RT:10.255.1.11:1 ENCAP:8 Router MAC:EC1D.8B55.F55D
  <<- Route Target and RMAC of Border Leaf VTEP
  Originator: 10.255.1.11, Cluster list: 10.2.2.2
  rx pathid: 0, tx pathid: 0x0

Border_Leaf_VTEP# show bgp vpnv4 unicast vrf vxlan 10.12.12.12/32
<<- To VXLAN Fabric IP address on VTEP-1
BGP routing table entry for 10.255.1.11:1:10.12.12.12/32, version 3092
Paths: (1 available, best #1, table vxlan)
Not advertised to any peer
Refresh Epoch 4
Local, imported path from [2][10.1.1.1:12][0][48][005F860210E7][32][10.12.12.12]/24 (global)

<<- EVPN type-2 has been imported to VPNv4, from VTEP-1
  10.255.1.1 (metric 3) (via default) from 10.2.2.2 (10.2.2.2)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  Extended Community: RT:10.255.1.11:1 ENCAP:8 Router MAC:7035.0956.7EDD
  Originator: 10.1.1.1, Cluster list: 10.2.2.2
  Local vxlan vtep:
    vrf:vxlan, vni:99999
    local router mac:EC1D.8B55.F55D
    encap:8

```

```

vtep-ip:10.255.1.11
bdi:Vlan99
Remote VxLAN:
Topoid 0x2(vrf vxlan)
Remote Router MAC:7035.0956.7EDD <<- VTEP-1 RMAC
Encap 8
Egress VNI 99999
RTEP 10.255.1.1 <<- VTEP-1 Tunnel IP address
rx pathid: 0, tx pathid: 0x0
    
```

```

Border_Leaf_VTEP# show bgp l2vpn evpn all route-type 2 0 005F860210E7 10.12.12.12
<<- Border_Leaf_VTEP still knows the type-2. This is still exchanged between the VTEPs
even though the prefix has been imported to VPNv4
BGP routing table entry for [2][10.1.1.1:12][0][48][005F860210E7][32][10.12.12.12]/24,
version 3085
Paths: (1 available, best #1, table EVPN-BGP-Table)
Not advertised to any peer
Refresh Epoch 4
Local
10.255.1.1 (metric 3) (via default) from 10.2.2.2 (10.2.2.2)
Origin incomplete, metric 0, localpref 100, valid, internal, best
EVPN ESI: 00000000000000000000, Label1 12012, Label2 99999
<<- Both Layer 2 VNI and Layer 3 VNI labels are seen in type-2,
but only Layer 3 VNI 99999 is used, once imported to VPNv4
Extended Community: RT:12:1 RT:10.255.1.1:1 ENCAP:8
Router MAC:7035.0956.7EDD
Originator: 10.1.1.1, Cluster list: 10.2.2.2
rx pathid: 0, tx pathid: 0x0
    
```



**Note** To check if IP routes have been inserted into CEF table, run the **show ip route vrf vrf-name** command in privileged EXEC mode.

### Confirm that Adjacency Exists to the VTEP Tunnel IP Address for the Host Device in IP VRF

The following examples show how to confirm that adjacency exists to the VTEP Tunnel IP address for the host device in IP VRF:

```

VTEP-1# show ip cef vrf vxlan 10.9.9.9/32 platform
10.9.9.9/32
Platform adj-id: 0x1A, 0x0, tun_qos_dpidx:0 <<- Adjacency ID to remote IP address
    
```

```

VTEP-1# show platform software fed sw ac matm macTable vlan 99
VLAN  MAC                               Type Seq#  EC_Bi  Flags machandle          siHandle
   riHandle                               diHandle      *a_time *e_time  ports
-----
99    7035.0956.7edd                          0x8002      0      0      64  0x7ffa48d61be8          0x7ffa48d630b8
      0x0                                    0x5154      0      0      0
99    7486.0bc4.b75d                          0x1000001   0      0      64  0x7ffa48fb1bb8          0x7ffa48fac698
      0x7ffa48fab038                        0x7ffa4838cc18 0      0      0      RLOC 10.255.2.1 adj_id
103
99    ec1d.8b55.f55d                          0x1000001   0      0      64  0x7ffa48d065e8          0x7ffa48d01d08
      0x7ffa48c9a618                        0x7ffa4838cc18 0      0      0      RLOC 10.255.1.11 adj_id
47
    
```

### Confirm that Adjacency exists to Reach Tunnel Destination

The following example shows how to confirm that adjacency exists to reach Tunnel destination:

```
VTEP-1# show ip cef 10.255.1.11  
10.255.1.11/32  
  nexthop 10.1.1.6 TwoGigabitEthernet1/0/2
```



## CHAPTER 10

# Feature History for BGP EVPN VXLAN

---

- [Feature History for BGP EVPN VXLAN, on page 261](#)

## Feature History for BGP EVPN VXLAN

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.1	EVPN VXLAN Overlay Network for IPv4 Bridged Traffic	EVPN VXLAN overlay network for IPv4 bridged traffic is a Layer 2 overlay network that allows host devices within the same subnet to send IPv4 bridged traffic to each other using a Layer 2 virtual network instance (VNI).
	EVPN VXLAN Overlay Network for IPv4 Routed Traffic	EVPN VXLAN overlay network for IPv4 routed traffic is a Layer 3 overlay network that allows host devices in different Layer 2 networks to send IPv4 routed traffic to each other using a Layer 3 VNI and an IP VRF.
	Layer 2 Broadcast, Unknown Unicast, and Multicast (BUM) Traffic Forwarding using Underlay Multicast	Multi-destination Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic in an EVPN VXLAN network is replicated through a multicast group in the underlay network and forwarded to all the endpoints of the network.
	Leaf Functionality	A leaf switch sits on the edge of a BGP EVPN VXLAN fabric and is connected to the host or access devices. It functions as a virtual tunnel end point (VTEP) and performs encapsulation and decapsulation.
	EVPN VXLAN Integrated Routing and Bridging	EVPN VXLAN integrated and bridging (IRB) allows the VTEPs in a VXLAN network to forward both Layer 2 or bridged traffic and Layer 3 or routed traffic. It is implemented as symmetric and asymmetric IRB.
	EVPN VXLAN Distributed Anycast Gateway	EVPN VXLAN distributed anycast gateway is a default gateway addressing mechanism that enables the use of the same gateway IP address across all the leaf switches that are part of a VXLAN network.  Support was introduced with manual MAC address configuration on the Layer 2 VNI VLAN's switch virtual interface (SVI) on all VTEPs as the only method to enable the feature.
	DHCP Relay for IPv4 Traffic in BGP EVPN VXLAN Fabric	The VTEP in a BGP EVPN VXLAN fabric is configured as a DHCP relay agent to provide DHCP relay services for IPv4 traffic in a multi-tenant VXLAN environment.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	EVPN VXLAN Overlay Network for IPv6 Bridged Traffic	EVPN VXLAN overlay network for IPv6 bridged traffic is a Layer 2 overlay network that allows host devices within the same subnet to send IPv6 bridged traffic to each other using a Layer 2 VNI.
	EVPN VXLAN Overlay Network for IPv6 Routed Traffic	EVPN VXLAN overlay network for IPv6 routed traffic is a Layer 3 overlay network that allows host devices in different Layer 2 networks to send IPv6 routed traffic to each other using a Layer 3 VNI and an IP VRF.
	Layer 2 Broadcast, Unknown Unicast, and Multicast (BUM) Traffic Forwarding using Ingress Replication	Ingress replication is a unicast approach to handle multi-destination Layer 2 BUM traffic in an EVPN VXLAN network. It involves an ingress device replicating every incoming BUM packet and sending them as a separate unicast to the remote egress devices.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.12.1	MAC Aliasing for EVPN VXLAN Distributed Anycast Gateway	<p>MAC aliasing allows the leaf switches in an EVPN VXLAN network to advertise the MAC addresses of their Layer 2 VLAN's SVI as the gateway MAC address to all the other leaf switches in the network.</p> <p>MAC aliasing removes the need to explicitly configure the same MAC address on the Layer 2 VNI VLAN's SVI on all VTEPs in order to enable distributed anycast gateway.</p>
	EVPN VXLAN Multihoming in Single-active Redundancy Mode	<p>Multi-homing provides redundancy in the connection between a customer edge (CE) device and a VTEP by connecting the customer network with multiple VTEPs in an EVPN VXLAN network.</p> <p>In single-active redundancy mode, only one VTEP, among a group of VTEPs that are attached to the particular ethernet segment, is allowed to forward traffic to and from that ethernet segment.</p> <p>Multi-homing in single-active redundancy mode was introduced only in the form of dual-homing, allowing a CE device to be connected to two VTEPs.</p>
	Border Leaf Functionality	A border leaf switch is a leaf switch in a BGP EVPN VXLAN fabric that enables external connectivity with other Layer 2 and Layer 3 networks by acting as the connecting node between the two networks.
	Autonomous System Number Rewrite	The <b>rewrite-evpn-rt-asn</b> command was introduced to enable the rewrite of the autonomous system number (ASN) of the EVPN route target that originates from the current autonomous system with the ASN of the target eBGP EVPN peer.
	VRF-Lite Border Leaf Handoff	VRF-Lite border leaf handoff in a BGP EVPN VXLAN fabric allows Layer 3 external connectivity with a VRF-Lite network through a border leaf switch.
	MPLS Layer 3 VPN Border Leaf Handoff	MPLS Layer 3 VPN border leaf handoff in a BGP EVPN VXLAN fabric allows Layer 3 external connectivity with an MPLS Layer 3 VPN network through a border leaf switch.
	IEEE 802.1Q Border Leaf Handoff	IEEE 802.1Q border leaf handoff in a BGP EVPN VXLAN fabric allows Layer 2 external connectivity with an IEEE 802.1Q network through a border leaf switch.
	Access Border Leaf Handoff	Access border leaf handoff in a BGP EVPN VXLAN fabric allows Layer 2 external connectivity with an Access network through a border leaf switch.



Release	Feature	Feature Information
	VPLS over MPLS Border Leaf Handoff	VPLS over MPLS border leaf handoff in a BGP EVPN VXLAN fabric allows Layer 2 external connectivity with a VPLS over MPLS network through a border leaf switch.
Cisco IOS XE Amsterdam 17.1.1	Spine Functionality	A spine switch acts as the connecting node between all the leaf switches in a BGP EVPN VXLAN fabric, forwards the traffic between the leaf switches and provides redundancy to the network.
	Border Spine Functionality	A border spine switch in a BGP EVPN VXLAN fabric enables external connectivity with other Layer 2 and Layer 3 networks by acting as the connecting node between the two networks.
	Layer 3 Tenant Routed Multicast for IPv4 Traffic	Layer 3 tenant routed multicast (TRM) for IPv4 traffic enables multicast forwarding for IPv4 traffic in a BGP EVPN VXLAN fabric. It provides multi-tenancy-aware multicast forwarding between senders and receivers within the same subnet or different subnets, locally or across VTEPs.
	VRF-Lite Border Spine Handoff	VRF-Lite border spine handoff in a BGP EVPN VXLAN fabric allows Layer 3 external connectivity with a VRF-Lite network through a border spine switch.
	MPLS Layer 3 VPN Border Spine Handoff	MPLS Layer 3 VPN border spine handoff in a BGP EVPN VXLAN fabric allows Layer 3 external connectivity with an MPLS Layer 3 VPN network through a border spine switch.
	IEEE 802.1Q Border Spine Handoff	IEEE 802.1Q border spine handoff in a BGP EVPN VXLAN fabric allows Layer 2 external connectivity with an IEEE 802.1Q network through a border spine switch.
	Access Network Border Spine Handoff	Access border spine handoff in a BGP EVPN VXLAN fabric allows Layer 2 external connectivity with an Access network through a border spine switch.
	VPLS over MPLS Border Spine Handoff	VPLS over MPLS border spine handoff in a BGP EVPN VXLAN fabric allows Layer 2 external connectivity with a VPLS over MPLS network through a border spine switch.
	BGP EVPN VXLAN MIB support	Support was introduced for the MIB.
Cisco IOS XE Amsterdam 17.2.1	ARP and IPv6 Neighbor Discovery Flooding Suppression	Flooding suppression avoids the flooding of ARP and IPv6 neighbor discovery packets over the VXLAN network to the local and remote host or access devices.
	EVPN VXLAN Centralized Default Gateway	EVPN VXLAN Centralized Default Gateway allows a single VTEP in the EVPN VXLAN network to act as the Layer 3 gateway for all the Layer 2 VNIs in the network.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>.