



BGP EVPN VXLAN Configuration Guide, Cisco IOS XE Bengaluru 17.6.x (Catalyst 9500 Switches)

First Published: 2021-07-30

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.

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: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. 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)

© 2021 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

BGP EVPN VXLAN Overview 1

Restrictions for BGP EVPN VXLAN	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	3
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
BUM Traffic Rate Limiting	13
Flooding Suppression	13
How to Configure EVPN VXLAN Layer 2 Overlay Network	14
Configuring Layer 2 VPN EVPN on a VTEP	15
Configuring an EVPN Instance on the VLAN on a VTEP	18
Configuring the Access-Facing Interface in the VLAN on a VTEP	19
Configuring the Loopback Interface on a VTEP	19
Configuring the NVE Interface on a VTEP	20
Configuring BGP on a VTEP with EVPN Address Family	21
Verifying EVPN VXLAN Layer 2 Overlay Network	23
Configuration Examples for EVPN VXLAN Layer 2 Overlay Network	25
Example: Configuring Layer 2 VNI with Back-to-Back Multicast Replication	25
Example: Configuring Layer 2 VNI with Back to Back Ingress Replication	33
Example: Configuring Layer 2 VNI with Spine Multicast Replication	39
Example: Configuring Layer 2 VNI with Spine Ingress Replication	57
Example: Configuring BUM Traffic Rate Limiting	69

CHAPTER 3

Configuring EVPN VXLAN Layer 3 Overlay Network	73
Information About EVPN VXLAN Layer 3 Overlay Network	73
How to Configure EVPN VXLAN Layer 3 Overlay Network	74
Configuring an IP VRF on a VTEP	74
Configuring the Core-facing VLAN on a VTEP	76
Configuring Access-facing VLAN on a VTEP	77
Configuring Switch Virtual Interface for the Core-facing VLAN	77
Configuring the Switch Virtual Interface for the Access-facing VLANs	78
Configuring the Loopback Interface on a VTEP	79
Configuring the NVE Interface on a VTEP	80
Configuring BGP with IPv4 or IPv6 or Both Address Families on VTEP	81
Configuration Examples for EVPN VXLAN Layer 3 Overlay Network	84
Verifying EVPN VXLAN Layer 3 Overlay Network	91

CHAPTER 4

Configuring EVPN VXLAN Integrated Routing and Bridging	93
Restrictions for EVPN VXLAN Integrated Routing and Bridging	93
Information About EVPN VXLAN Integrated Routing and Bridging	93
EVPN VXLAN Distributed Anycast Gateway	94
Manual MAC Address Configuration	96
MAC Aliasing	96
EVPN VXLAN Centralized Default Gateway	96
Default Gateway MAC Address Assignment	98
Route Type 2 to Route Type 5 Host Route Reorigination for Overlay Networks with a Centralized Gateway (Asymmetric IRB)	98
How to Configure EVPN VXLAN Integrated Routing and Bridging	99
Configuring EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway	99
Configuring Layer 2 VPN EVPN on a VTEP	99
Configuring IP VRF on VTEP	99
Configuring the Loopback Interface on a VTEP	99
Configuring Core-facing and Access-facing VLANs on a VTEP	99
Configuring Switch Virtual Interface for the Core-facing VLAN on a VTEP	101
Configuring Switch Virtual Interface for the Access-facing VLANs on a VTEP	102
Configuring the NVE Interface on a VTEP	102
Configuring BGP with EVPN and VRF Address Families on a VTEP	104
Configuring EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway	107
Verifying EVPN VXLAN Integrated Routing and Bridging	108
Configuration Examples for EVPN VXLAN Integrated Routing and Bridging	108
Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Distributed Anycast Gateway	108
Example: Enabling EVPN VXLAN Integrated Routing and Bridging using Centralized Default Gateway	123

CHAPTER 5

Configuring Spine Switches in a BGP EVPN VXLAN Fabric	141
Information About Spine Switches in a BGP EVPN VXLAN Fabric	141
Deployment Scenarios for Spine Switches and Leaf Switches in a BGP EVPN VXLAN Fabric	141
Configuration Examples for Spine Switches in a BGP EVPN VXLAN Network	142

Configuration Example for Spine Switches Using iBGP when the Spine Switches and Leaf Switches are in the Same Autonomous System 142

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 159

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 178

CHAPTER 6

Configuring DHCP Relay in a BGP EVPN VXLAN Fabric 199

Restrictions for DHCP Relay in a BGP EVPN VXLAN Fabric 199

Information About DHCP Relay in a BGP EVPN VXLAN Fabric 199

DHCP Relay on VTEPs 200

How to Configure DHCP Relay in a BGP EVPN VXLAN Fabric 201

Configuring DHCP Relay on a VTEP 201

Configuring DHCP Relay on the Access SVI of a VTEP 202

Configuring the Layer 3 or Routed Interface on the Border VTEP for DHCP Server Reachability 203

Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric 205

Example: DHCP Server is in the Layer 3 Default VRF and the DHCP Client is in the Tenant VRF 206

Example: DHCP Server and DHCP Client are in the Same Tenant VRF 211

Example: DHCP Client and DHCP Server are in Different Tenant VRFs 214

Example: DHCP Server is in a non-Default, non-VXLAN VRF and DHCP Client is in the Tenant VRF 219

Additional References for DHCP Relay in a BGP EVPN VXLAN Fabric 223

CHAPTER 7

Configuring VXLAN-Aware Flexible Netflow 225

Information About VXLAN-Aware Flexible NetFlow 225

How to Configure VXLAN-Aware Flexible NetFlow 225

Configuring a Flow Record 226

Configuring Flow Exporter 227

Configuring a Flow Monitor 228

Configuring Flexible NetFlow on an NVE Interface 229

Configuration Examples for VXLAN-Aware Flexible NetFlow 230

CHAPTER 8

Configuring Tenant Routed Multicast 237

Restrictions for Tenant Routed Multicast 237

Information about Tenant Routed Multicast	238
TRM in PIM Sparse Mode	240
RP Placement	240
PIM Sparse Mode with Anycast RP	243
PIM Sparse Mode with RP Inside the BGP EVPN VXLAN Fabric	244
PIM Sparse Mode with RP Outside the BGP EVPN VXLAN Fabric	245
TRM in PIM Source Specific Mode	247
Data MDT	248
How to Configure Tenant Routed Multicast	250
Configuring TRM with PIM Sparse Mode	250
Configuring the TRM Multicast Distribution Tree in the VRF	250
Configuring Multicast Routing on the Overlay VRF	252
Configuring Multicast on Switch Virtual Interfaces for the Core-facing and Access-facing VLANs	252
Configuring BGP with MVPN Address Family on VTEP	253
Configuring RP for Underlay Network	254
Configuring RP for Overlay Network	255
Configuring TRM with PIM Source Specific Mode	256
Configuring SSM for Overlay Network	257
Verifying Tenant Routed Multicast	257
Troubleshooting Tenant Routed Multicast	258
Configuration Examples for Tenant Routed Multicast	258
Example: Configuring TRM in PIM Sparse Mode with Anycast RP	258
Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric	300
Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric	332
Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric	371
Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric	402
CHAPTER 9	Configuring EVPN VXLAN External Connectivity 443
	Restrictions for EVPN VXLAN External Connectivity 443
	Information About EVPN VXLAN External Connectivity 443

Implementation of Border Nodes for EVPN VXLAN External Connectivity	443
External Connectivity with Layer 3 Networks	445
External Connectivity with Layer 2 Networks	446
How to Configure EVPN VXLAN External Connectivity	447
Enabling Layer 3 External Connectivity with VRF-Lite	447
Configuring the VRF on the Border VTEP Interface that Faces the External Router	448
Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN	449
Configuring BGP on a Border VTEP for External Connectivity with MPLS Layer 3 VPN	450
Enabling EVPN VXLAN Layer 3 TRM Interworking with MVPN Networks	454
Enabling Layer 2 External Connectivity with IEEE 802.1Q Networks	455
Enabling Layer 2 External Connectivity with a VPLS Network Through an Access VFI	456
Defining an Access VFI on a Border VTEP	457
Adding an Access VFI and an EVPN Instance as Members of the VLAN of a Border VTEP	458
Configuring VPLS on a Border VTEP	459
Configuration Examples for EVPN VXLAN External Connectivity	459
Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through iBGP	459
Example: Enabling Layer 3 External Connectivity with MPLS Layer 3 VPN through eBGP	469
<hr/>	
CHAPTER 10	Configuring Multi-Homing in a BGP EVPN VXLAN Fabric
	481
Restrictions for Multi-Homing in a BGP EVPN VXLAN Fabric	481
Information About Multi-Homing in a BGP EVPN VXLAN Fabric	481
Single-Homing	482
Multi-Homing	483
Single-Active Redundancy Mode	484
Migration Between Single-Homed and Multi-Homed Network Topologies	487
How to Configure Multi-Homing in a BGP EVPN VXLAN Fabric	487
Configuring Dual-Homing with Single Active Redundancy in a BGP EVPN VXLAN Fabric	487
Configuring Redundancy on an Ethernet Segment	487
Associating an Ethernet Segment with an Interface on a VTEP	488
Migrating from a Single-Homed Topology to a Single-Active Dual-Homed Topology	489
Migrating from a Single-Active Dual-Homed Topology to a Single-Homed Topology	491
Configuration Examples for Multi-Homing in a BGP EVPN VXLAN Fabric	492
Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric	492

CHAPTER 11	Configuring Private VLANs in a BGP EVPN VXLAN Fabric	515
	Restrictions for Private VLANs in a BGP EVPN VXLAN Fabric	515
	Information About Private VLANs in a BGP EVPN VXLAN Fabric	515
	Primary and Secondary VLANs	516
	Private VLAN Ports	516
	Extension of Private VLANs in a BGP EVPN VXLAN Fabric	516
	Traffic Forwarding for Private VLANs in a BGP EVPN VXLAN Fabric	517
	Known Unicast Traffic Forwarding	517
	Broadcast, Unknown Unicast, and Multicast Traffic Forwarding	519
	Routed Traffic Forwarding	521
	How to Configure Private VLANs in a BGP EVPN VXLAN Fabric	521
	Configuring the Primary and Secondary VLANs for a Private VLAN	522
	Configuring the Port of a Private VLAN	523
	Enabling EVPN in a Private VLAN	524
	Configuration Examples for Private VLANs in a BGP EVPN VXLAN Fabric	525
CHAPTER 12	Scale and Performance Capabilities for BGP EVPN VXLAN on Cisco Catalyst 9500 Series Switches	551
	Scale and Performance Capabilities for BGP EVPN VXLAN on Cisco Catalyst 9500 Series Switches	551
CHAPTER 13	Troubleshooting BGP EVPN VXLAN	557
	Troubleshooting Scenarios for BGP EVPN VXLAN	557
	Troubleshooting Broadcast, Unknown Unicast, Multicast Traffic Forwarding	558
	Troubleshooting Unicast Forwarding Between VTEPs in the Same VLAN Through a Layer 2 VNI	562
	Verifying the Provisioning of an EVPN VXLAN Layer 2 Overlay Network	563
	Verifying Intra-Subnet Traffic Movement in an EVPN VXLAN Layer 2 Overlay Network	567
	Troubleshooting Unicast Forwarding Between VTEPs in Different VLANs Through a Layer 3 VNI	574
	Verifying the Provisioning of an EVPN VXLAN Layer 3 Overlay Network	575
	Verifying Inter-Subnet Traffic Movement and Symmetric IRB in an EVPN VXLAN Layer 3 Overlay Network	580
	Troubleshooting Unicast Forwarding Between a VXLAN Network and an IP Network	587
	Verifying the Provisioning of an EVPN VXLAN Layer 3 Overlay Network	587
	Verifying Traffic from a VXLAN Fabric to an IP Network Through a Border Leaf Switch Using Route Type 5	587

Troubleshooting Tenant Routed Multicast 590

PART I

Integration with Cisco DNA Service for Bonjour 591

CHAPTER 14

Cisco DNA Service for Bonjour Solution Overview 593

About Cisco DNA Service for Bonjour Solution 593

Solution Components 594

Supported Platforms 595

Supported Network Design 596

Traditional Wired and Wireless Networks 596

Wired Networks 597

Wireless Networks 598

Cisco SD-Access Wired and Wireless Networks 600

Fabric-Enabled Wired and Wireless Networks 600

Fabric-Enabled Policy Extended Node 602

BGP EVPN Networks 603

CHAPTER 15

Configuring Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks 605

Prerequisites for Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks 605

Restrictions for Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks 606

Information About Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks 606

Understanding Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks 607

How to Configure Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks 609

Configuring Local Area Bonjour in Unicast Mode over EVPN VXLAN Layer 3 Overlay Networks 609

Configuring Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks 616

Verifying Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks 619

Verifying Local Area Bonjour over an EVPN VXLAN Layer 3 Overlay Network 619

Verifying Wired Service-Peer Configuration 620

Verifying Wired SDG Agent Configuration and Service-Routing Status 621

Verifying Wireless Service-Peer and Service Routing Status 623

Verifying Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks 624

Verifying Cisco DNA Center Configuration and Service-Routing Status 626

Additional References for Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks 626

CHAPTER 16	Configuring VRF-Aware Local Area Bonjour Services	627
	Prerequisites for VRF-Aware Local Area Bonjour Services	627
	Restrictions for VRF-Aware Local Area Bonjour Services	628
	Information about VRF-Aware Local Area Bonjour Services	628
	Gateway Modes for VRF-Aware Bonjour Services	629
	Understanding VRF-Aware Wide Area Bonjour Services	630
	Understanding VRF-Aware Service on Multilayered Wired and Wireless Networks	632
	How to configure Intra-Virtual Network Proxy Service on Local Area Bonjour Domain	633
	How to configure Inter-Virtual Network Proxy Service on Local Area Bonjour Domain	634
	Configuring Inter-Virtual Network Location-Filter	635
	Verifying VRF-Aware Local Area Bonjour Services	637

PART II	Feature History for BGP EVPN VXLAN	641
----------------	---	------------

CHAPTER 17	Feature History for BGP EVPN VXLAN	643
	Feature History for BGP EVPN VXLAN	643



CHAPTER 1

BGP EVPN VXLAN Overview

- [Restrictions for BGP EVPN VXLAN, on page 1](#)
- [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](#)

Restrictions for BGP EVPN VXLAN

BGP EVPN VXLAN does not support Generic Routing Encapsulation (GRE) in the underlay.

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 ratified IETF RFC specifications RFC [7432](#) and RFC [8365](#) supporting BGP EVPN control plane with RFC [7348](#) VXLAN data-plane. In addition multiple other BGP EVPN RFCs and Internet drafts submitted by the BGP Enabled ServicesS (bess¹) workgroup are supported. 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.

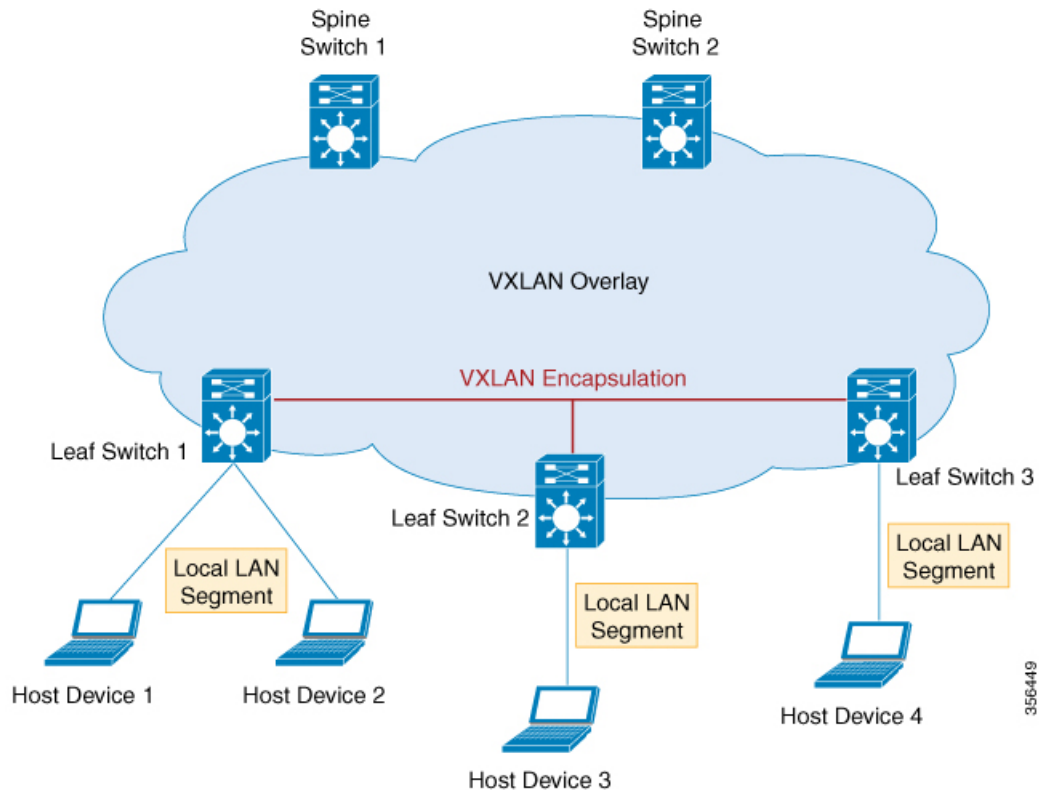
Virtual Tunnel End Points

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.

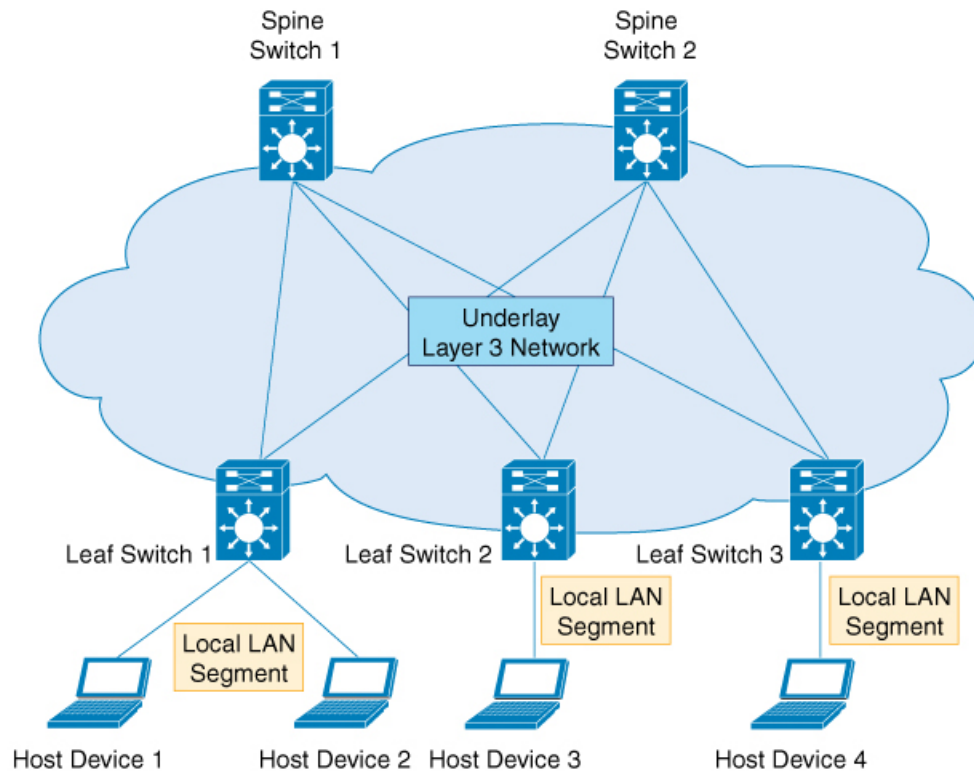
Without TRM, the multicast traffic is sent as part of the underlay network in the form of BUM traffic either using underlay multicast or ingress replication methods. This does not allow sources and receivers that are present across different subnets to communicate with each other. Using TRM, multicast 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.

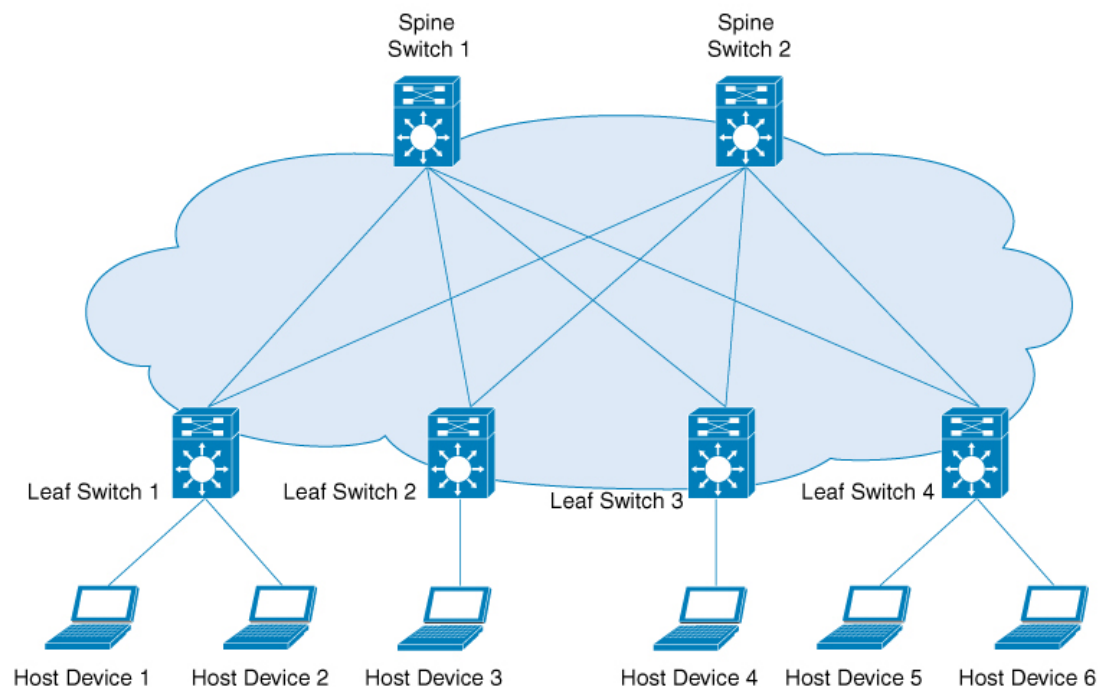
Spine Switch

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:

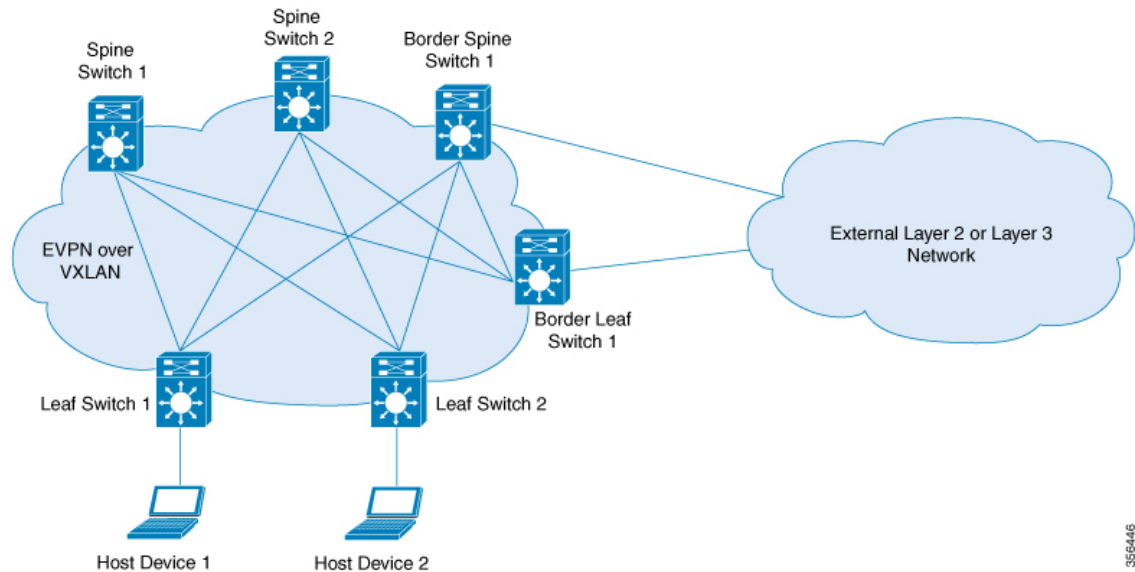


356448

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:



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 93](#) 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.

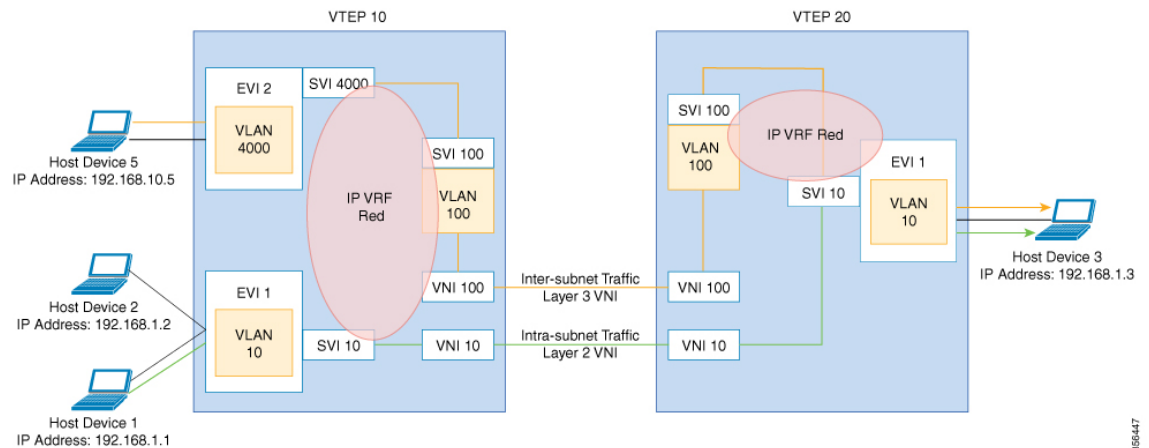
A VLAN that is locally defined on a VTEP can be mapped to a Layer 2 VNI. In order to allow host devices to connect to a Layer 2 VNI, the connected VLAN must be mapped to the Layer 2 VNI, and then the Layer 2 VNI is associated with the Network Virtualization Edge (NVE) logical interface on the VTEP.

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](#)
- [Verifying EVPN VXLAN Layer 2 Overlay Network, on page 23](#)
- [Configuration Examples for EVPN VXLAN Layer 2 Overlay Network, on page 25](#)

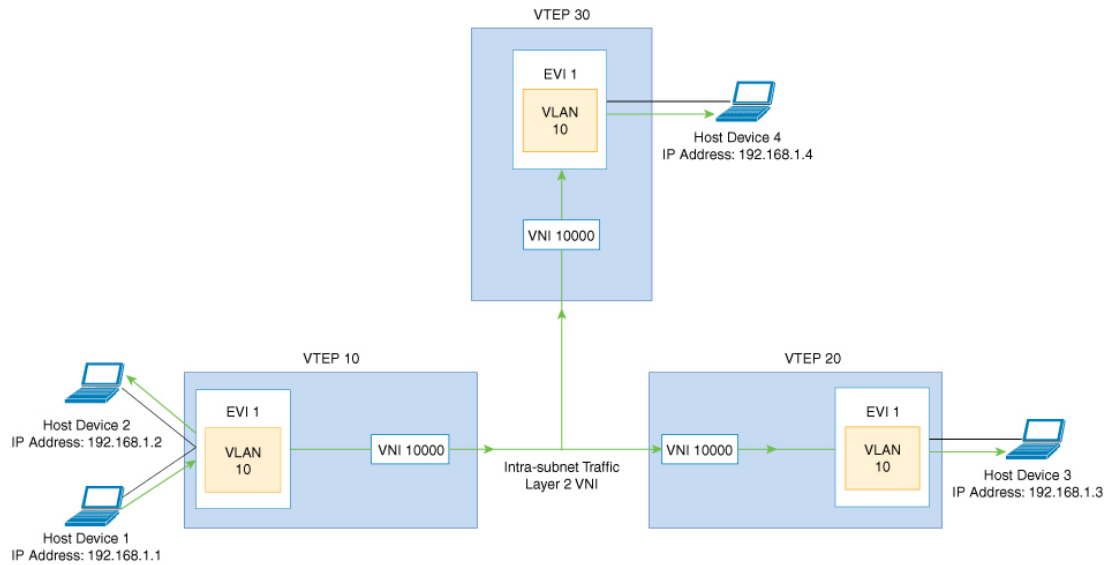
Information About EVPN VXLAN Layer 2 Overlay Network

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.

BUM Traffic Rate Limiting

You can use a policer to set the flood rate limit of the BUM traffic in the network to a predefined value. This prevents the flood rate from going beyond the limit and saves the network bandwidth.

To set the flood rate limit, configure a policy with a Layer 2 miss filter on the NVE interface of a VTEP. Ensure that the policy is applied on the NVE interface for egress traffic. All the Layer 2 member VNIs under this NVE share the same policy. Any new Layer 2 VNI that is added under the NVE shares this configured policy.

See [Example: Configuring BUM Traffic Rate Limiting, on page 69](#) for a sample topology and configuration example.

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.

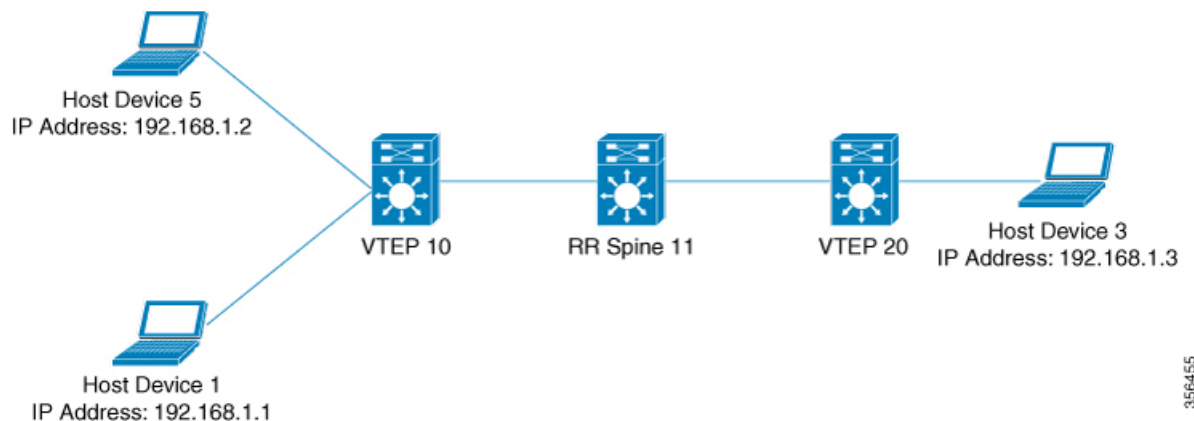
IPv6 Neighbor Discovery Flooding Suppression

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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	l2vpn evpn Example: Device (config)# l2vpn evpn	Enters EVPN configuration mode.
Step 4	replication-type {ingress static} Example: Device (config-evpn)# replication-type static	Configures the Layer 2 VPN EVPN replication type. Note 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.
Step 5	router-id loopback-interface-id Example: Device (config-evpn)# router-id loopback 0	Specifies the interface that will supply the IP addresses to be used in auto-generating route distinguishers.
Step 6	default-gateway advertise Example:	(Optional) Enables default gateway advertisement on the switch. To configure distributed anycast gateway in a VXLAN

	Command or Action	Purpose
	Device (config-evpn) # default-gateway advertise	<p>network using MAC aliasing, enable default gateway advertisement on all the leaf switches in the network.</p> <p>This command is applicable in integrated routing and bridging (IRB) scenarios where Layer 2 and Layer 3 VNIs coexist in a VRF. Refer to <i>Configuring EVPN VXLAN Integrated Routing and Bridging</i> module for more details.</p> <p>This command is mandatory only if the same MAC address is not manually configured on all the access SVIs.</p> <p>Note Use the default-gateway advertise {enable disable} command in EVPN instance configuration mode to override the global default gateway advertisement settings and enable or disable it for a specific EVPN instance.</p>
Step 7	logging peer state Example: Device (config-evpn) # logging peer state	(Optional) Displays syslog message when the first route is received or the last route is withdrawn from a given remote VTEP.
Step 8	mac duplication limit limit-number time time-limit Example: Device (config-evpn) # mac duplication limit 20 time 5	(Optional) Changes parameters for detecting duplicate MAC addresses.
Step 9	ip duplication limit limit-number time time-limit Example: Device (config-evpn) # ip duplication limit 20 time 5	(Optional) Changes parameters for detecting duplicate IP addresses.
Step 10	route-target auto vni Example: Device (config-evpn) # route-target auto vni	(Optional) Specifies to use VNI instead of EVPN instance number to auto-generate route target.
Step 11	exit Example: Device (config-evpn) # exit	Exits EVPN configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 12	l2vpn evpn instance <i>evpn-instance-number</i> vlan-based Example: <pre>Device(config)# l2vpn evpn instance 1 vlan-based</pre>	<p>Configures a VLAN based EVPN instance in Layer 2 VPN configuration mode.</p> <p>An EVPN instance needs to be explicitly configured only when something needs to be configured per EVPN instance such as a route target.</p>
Step 13	encapsulation vxlan Example: <pre>Device(config-evpn-evi)# encapsulation vxlan</pre>	<p>(Optional) Defines the encapsulation format as VXLAN.</p> <p>The encapsulation format is VXLAN by default.</p>
Step 14	replication-type {ingress static} Example: <pre>Device(config-evpn-evi)# replication-type ingress</pre>	<p>(Optional) Sets the replication type for the EVPN instance.</p> <p>In case a global replication type has already been configured, this overrides the global setting.</p>
Step 15	default-gateway advertise {enable disable} Example: <pre>Device(config-evpn-evi)# default-gateway advertise disable</pre>	<p>(Optional) Enables or disables the default gateway advertisement for the EVPN instance.</p> <p>In case default gateway advertisement has already been globally configured, this overrides the global setting.</p> <p>This command is mandatory only if the same MAC address is not manually configured on all the access SVIs.</p> <p>To configure distributed anycast gateway in a VXLAN network using MAC aliasing, enable default gateway advertisement on all the leaf switches in the network.</p>
Step 16	ip local-learning {enable disable} Example: <pre>Device(config-evpn-evi)# ip local-learning disable</pre>	<p>(Optional) Enables or disables local IP address learning for the specified EVPN instance.</p> <p>In case IP address learning has already been globally configured, this overrides the global setting.</p>
Step 17	re-originate route-type5 Example: <pre>Device(config-evpn-evi)# re-originate route-type5</pre>	<p>(Optional) Enables the centralized gateway (CGW) VTEP to re-originate the route-type 2 (RT 2) host routes from a Layer 2 VTEP as route-type 5 (RT 5) network routes into a Layer 3 overlay network.</p>
Step 18	no auto-route-target Example: <pre>Device(config-evpn-evi)# no auto-route-target</pre>	<p>(Optional) Disables auto generation of route targets.</p>

	Command or Action	Purpose
Step 19	rd <i>rd-value</i> Example: Device(config-evpn-evi) # rd 65000:100	(Optional) Configures a route distinguisher manually.
Step 20	route-target { import export both } <i>rt-value</i> Example: Device(config-evpn-evi) # route-target both 65000:100	(Optional) Configures route targets manually. Note Configure route targets manually if the auto-generated route target values (ASN:EVI or ASN:VNI) are different between the VTEPs.
Step 21	end Example: Device(config-evpn-evi) # end	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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vlan configuration <i>vlan-id</i> Example: Device(config)# vlan configuration 11	Enters VLAN feature configuration mode for the specified VLAN interface.
Step 4	member evpn-instance <i>evpn-instance-id vni l2-vni-number</i> Example: Device(config-vlan) # member evpn-instance 1 vni 10000	Adds EVPN instance as a member of the VLAN configuration. The VNI here is used as a Layer 2 VNI.
Step 5	end Example: Device(config-vlan) # end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-name</i> Example: Device(config)# interface GigabitEthernet1/0/1	Enters interface configuration mode for the specified interface.
Step 4	switchport access vlan <i>vlan-id</i> Example: Device(config-if)# switchport access vlan 11	Configures the interface as a static-access port of the specified VLAN. Interface can also be configured as a trunk interface, if required.
Step 5	end Example: Device(config-if)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>loopback-interface-id</i> Example: Device(config)# interface Loopback0	Enters interface configuration mode for the specified Loopback interface.
Step 4	ip address <i>ipv4-address</i> Example: Device(config-if)# ip address 10.12.11.11	Configures the IP address for the Loopback interface.
Step 5	ip pim sparse mode Example: Device(config-if)# ip pim sparse mode	Enables Protocol Independent Multicast (PIM) sparse mode on the Loopback interface.
Step 6	end Example: Device(config-vlan)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>nve-interface-id</i> Example: Device(config)# interface nve1	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	no ip address Example: Device(config-if)# no ip address	Disables IP processing on the interface by removing its IP address.
Step 5	source-interface <i>loopback-interface-id</i> Example: Device(config-if)# source-interface loopback0	Sets the IP address of the specified loopback interface as the source IP address.

	Command or Action	Purpose
Step 6	host-reachability protocol bgp Example: Device(config-if)# host-reachability protocol bgp	Configures BGP as the host-reachability protocol on the interface. Note You must configure the host-reachability protocol on the interface. If you do not execute this step, the VXLAN tunnel defaults to static VXLAN tunnel, which is currently not supported on the Cisco Catalyst 9000 Series switches.
Step 7	member vni layer2-vni-id {ingress-replication [local-routing] mcast-group multicast-group-address} Example: Device(config-if)# member vni 10000 mcast-group 227.0.0.1	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 mcast-group keyword for static replication and ingress-replication keyword for ingress replication. Use the local-routing keyword only when you need to configure route type 2 (RT 2) to route type 5 (RT 5) reorigination on the centralized gateway (CGW) VTEP.
Step 8	end Example: Device(config-if)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device (config) # router bgp 1	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
Step 4	bgp log-neighbor-changes Example: Device (config-router) # bgp log-neighbor-changes	(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	bgp update-delay <i>time-period</i> Example: Device (config-router) # bgp update-delay 1	(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> .
Step 6	bgp graceful-restart Example: Device (config-router) # bgp graceful-restart	(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	no bgp default ipv4-unicast Example: Device (config-router) # no bgp default ipv4-unicast	(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	neighbor <i>ip-address remote-as number</i> Example: Device (config-router) # neighbor 10.11.11.11 remote-as 1	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	neighbor { <i>ip-address</i> <i>group-name</i> } update-source <i>interface</i> Example: Device (config-router) # neighbor 10.11.11.11 update-source 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.
Step 10	address-family <i>l2vpn evpn</i> Example: Device (config-router) # address-family l2vpn evpn	Specifies the L2VPN address family and enters address family configuration mode.

	Command or Action	Purpose
Step 11	neighbor ip-address activate Example: Device(config-router-af)# neighbor 10.11.11.11 activate	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 12	neighbor ip-address send-community [both extended standard] Example: Device(config-router-af)# neighbor 10.11.11.11 send-community both	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 13	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 14	end Example: Device(config-router)# end	Returns to privileged EXEC mode.

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 1: Commands to Verify EVPN VXLAN Layer 2 Overlay Network

Command	Purpose
show l2vpn evpn evi [detail]	Displays detailed information for a particular EVPN instance or all EVPN instances.
show l2vpn evpn mac [detail]	Displays the MAC address database for Layer 2 EVPN.
show l2vpn evpn mac ip [detail]	Displays the IP address database for Layer 2 EVPN.
show l2vpn evpn summary	Displays a summary of Layer 2 EVPN information.
show l2vpn evpn capabilities	Displays platform capability information for Layer 2 EVPN.
show l2vpn evpn peers	Displays Layer 2 EVPN peer route counts and up time.
show l2vpn evpn route-target	Displays Layer 2 EVPN import route targets.
show l2vpn evpn memory	Displays Layer 2 EVPN memory usage.

Command	Purpose
show l2route evpn summary	Displays a summary of EVPN routes.
show l2route evpn mac [detail]	Displays MAC address information learnt by the switch in the EVPN control plane.
show l2route evpn mac ip [detail]	Displays MAC and IP address information learnt by the switch in the EVPN control plane.
show l2route evpn imet detail	Displays the IMET route details for Layer 2 EVPN address family. This command shows details only about traffic forwarded using ingress replication.
show bgp l2vpn evpn	Displays BGP information for Layer 2 VPN EVPN address family.
show bgp l2vpn evpn route-type 2	Displays BGP information for route type 2 of L2VPN EVPN address family.
show bgp l2vpn evpn evi context	Displays context information for Layer 2 EVPN instances.
show bgp l2vpn evpn evi <i>evpn-instance-id</i> route-type 3	Displays route type 3 information for the specified Layer 2 EVPN instance. This command shows details only about traffic forwarded using ingress replication.
show l2fib bridge-domain <i>bridge-domain-number</i> detail	Displays detailed information for a Layer 2 forwarding information base bridge domain.
show l2fib bridge-domain <i>bridge-domain-number</i> address unicast	Displays unicast MAC address information for a Layer 2 forwarding information base bridge domain.
show nve vni	Displays information about VXLAN network identifier members associated with an NVE interface.
show nve vni <i>vni-id</i> detail	Displays detailed NVE interface state information for a VXLAN network identifier member.
show nve peers	Displays NVE interface state information for peer leaf switches.
show mac address-table vlan <i>vlan-id</i>	Displays MAC addresses for a VLAN.
show platform software fed switch active matm macTable vlan <i>vlan-id</i>	Displays MAC addresses for a VLAN from MAC address table manager database for Forwarding Engine Driver (FED).
show device-tracking database	Displays device tracking database.
show device-tracking database mac	Displays device tracking MAC address database.

Command	Purpose
<code>show ip mroute</code>	Displays multicast routing table information.
<code>show ip bgp l2vpn evpn detail l2vpn-evpn-route</code>	Displays detailed information about a specific route.
<code>show ip bgp l2vpn evpn detail [mac-address ip-address]</code>	Display routes containing an IP address only or both MAC address and IP address.
<code>show ip bgp l2vpn evpn route-type 2 ethernet-tag {mac-address }</code>	Displays other MAC address formats for route-type 2 EVPN routes.

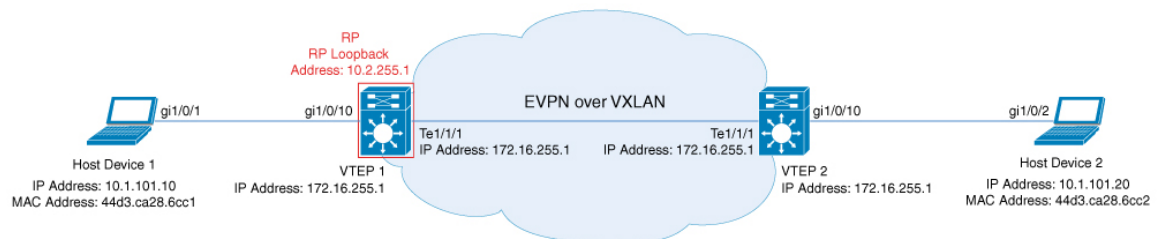
Configuration Examples for EVPN VXLAN Layer 2 Overlay Network

This sections provides configuration examples for EVPN VXLAN Layer 2 Overlay Network:

Example: Configuring Layer 2 VNI with Back-to-Back Multicast Replication

This example shows how to configure and verify a Layer 2 VNI with back-to-back multicast replication using the following topology:

Figure 1: EVPN VXLAN Network with a Layer 2 VNI with Multicast Replication



357016

The topology shows an EVPN VXLAN network with two VTEPs (VTEP 1 and VTEP 2) and no spine switches. Multicast replication is performed between the VTEPs to forward BUM traffic in the network. VTEP 1 acts as the rendezvous point (RP) for the multicast BUM traffic. The following table provides sample configurations for the devices in this topology:



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 2: Configuring VTEP 1 and VTEP 2 to Configure a Layer 2 VNI with Back-to-Back Multicast Replication

VTEP 1	VTEP 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access spanning-tree portfast ! interface TenGigabitEthernet1/1/1 no switchport ip address 172.16.12.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 ! </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! interface Loopback0 ip address 172.16.255.2 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback1 ip address 172.16.254.2 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access spanning-tree portfast ! interface TenGigabitEthernet1/1/1 no switchport ip address 172.16.12.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 ! </pre>

VTEP 1	VTEP 2
<pre> router ospf 1 router-id 172.16.255.1 ! router bgp 65001 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 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both exit-address-family ! ip pim rp-address 172.16.255.1 ! end Leaf-01# </pre>	<pre> router ospf 1 router-id 172.16.255.2 ! 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 172.16.255.1 ! end Leaf-02# </pre>

Verifying the Layer 2 VNI with Back-to-Back Multicast Replication

The following sections provide sample outputs for **show** commands to verify the Layer 2 VNI with back-to-back multicast replication on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 27](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 30](#)

Outputs to Verify the Configuration on VTEP 1

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

```

Leaf-01# show nve peers
Interface  VNI      Type Peer-IP           RMAC/Num_RTs  eVNI      state flags UP time
nve1      10101    L2CP 172.16.254.2      2             10101      UP    N/A  00:37:39

Leaf-01#
    
```

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

```

Leaf-01# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 7, main routing table version 7
6 network entries using 2304 bytes of memory
6 path entries using 1272 bytes of memory
2/2 BGP path/bestpath attribute entries using 576 bytes of memory
1 BGP extended community entries using 40 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 4192 total bytes of memory
BGP activity 6/0 prefixes, 6/0 paths, scan interval 60 secs
6 networks peaked at 10:04:33 Oct 26 2020 UTC (00:37:39.064 ago)
    
```

Example: Configuring Layer 2 VNI with Back-to-Back Multicast Replication

```
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2  4      65001    45     47       7     0     0 00:38:49      2
```

```
Leaf-01#
```

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

```
Leaf-01# show bgp l2vpn evpn
BGP table version is 7, local router ID is 172.16.255.1
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: 172.16.254.1:101
*> [2][172.16.254.1:101][0][48][44D3CA286CC1][0][*]/20
    :: 32768 ?
*> [2][172.16.254.1:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
    :: 32768 ?
*>i [2][172.16.254.1:101][0][48][44D3CA286CC2][0][*]/20
    172.16.254.2 0 100 0 ?
*>i [2][172.16.254.1:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
    172.16.254.2 0 100 0 ?
Route Distinguisher: 172.16.254.2:101
*>i [2][172.16.254.2:101][0][48][44D3CA286CC2][0][*]/20
    172.16.254.2 0 100 0 ?
*>i [2][172.16.254.2:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
    172.16.254.2 0 100 0 ?
```

```
Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn mac evi 101
MAC Address  EVI  VLAN  ESI  Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc1 101  101  0000.0000.0000.0000.0000 0  Gi1/0/10:101
44d3.ca28.6cc2 101  101  0000.0000.0000.0000.0000 0  172.16.254.2
```

```
Leaf-01#
```

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

```
Leaf-01# 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,
    e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:46:14/00:03:14, RP 172.16.255.1, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  TenGigabitEthernet1/1/1, Forward/Sparse, 00:43:31/00:03:14
  Loopback0, Forward/Sparse, 00:46:14/00:02:42

(*, 225.0.0.101), 00:46:14/stopped, RP 172.16.255.1, flags: SJCFx
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  TenGigabitEthernet1/1/1, Forward/Sparse, 00:43:31/00:03:17
  Tunnel0, Forward/Sparse-Dense, 00:46:14/00:01:47

(172.16.254.1, 225.0.0.101), 00:00:00/00:02:59, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
  TenGigabitEthernet1/1/1, Forward/Sparse, 00:00:00/00:03:29

(172.16.254.2, 225.0.0.101), 00:00:03/00:02:56, flags: x
Incoming interface: TenGigabitEthernet1/1/1, RPF nbr 172.16.12.2
Outgoing interface list:
  Tunnel0, Forward/Sparse-Dense, 00:00:03/00:02:56

Leaf-01#

```

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

```

Leaf-01# 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,
               e - Encap helper tunnel flag.
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
Tunnel2 Flags: A
TenGigabitEthernet1/1/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW

```

Example: Configuring Layer 2 VNI with Back-to-Back Multicast Replication

```

SW Forwarding: 2/0/96/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnel2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
TenGigabitEthernet1/1/1 Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
(172.16.254.1,225.0.0.101) Flags: HW
SW Forwarding: 1/0/96/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Null0 Flags: A
TenGigabitEthernet1/1/1 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.2,225.0.0.101) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnel2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
TenGigabitEthernet1/1/1 Flags: NS

```

```
Leaf-01#
```

Return to [Verifying the Layer 2 VNI with Back-to-Back Multicast Replication, on page 27](#).

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1      10101    L2CP 172.16.254.1      2          10101     UP    N/A  00:38:32

Leaf-02#

```

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

```

Leaf-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 7, main routing table version 7
6 network entries using 2304 bytes of memory
6 path entries using 1272 bytes of memory
2/2 BGP path/bestpath attribute entries using 576 bytes of memory
1 BGP extended community entries using 40 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 4192 total bytes of memory
BGP activity 6/0 prefixes, 6/0 paths, scan interval 60 secs
6 networks peaked at 10:02:19 Oct 26 2020 UTC (00:38:32.591 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   48     46       7     0     0 00:39:42      2

Leaf-02#

```

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

```

Leaf-02# show bgp l2vpn evpn
BGP table version is 7, local router ID is 172.16.255.2
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: 172.16.254.1:101
*>i [2][172.16.254.1:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.1          0      100      0 ?
*>i [2][172.16.254.1:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.1          0      100      0 ?
Route Distinguisher: 172.16.254.2:101
*>i [2][172.16.254.2:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.1          0      100      0 ?
*>i [2][172.16.254.2:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.1          0      100      0 ?
*> [2][172.16.254.2:101][0][48][44D3CA286CC2][0][*]/20
      ::                      32768 ?
*> [2][172.16.254.2:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      ::                      32768 ?

```

Leaf-02#

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

```

Leaf-02# show l2vpn evpn mac evi 101
MAC Address      EVI    VLAN  ESI                      Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc1  101    101   0000.0000.0000.0000.0000  0          172.16.254.1
44d3.ca28.6cc2  101    101   0000.0000.0000.0000.0000  0          Gi1/0/10:101

```

Leaf-02#

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

```

Leaf-02# 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,
e - encaps-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:43:49/00:02:09, RP 172.16.255.1, flags: SJCL
Incoming interface: TenGigabitEthernet1/1/1, RPF nbr 172.16.12.1
Outgoing interface list:
Loopback0, Forward/Sparse, 00:43:49/00:02:09

```

Example: Configuring Layer 2 VNI with Back-to-Back Multicast Replication

```
(*, 225.0.0.101), 00:43:49/stopped, RP 172.16.255.1, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/1/1, RPF nbr 172.16.12.1
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:43:49/00:01:11

(172.16.254.1, 225.0.0.101), 00:00:17/00:02:42, flags: JTx
  Incoming interface: TenGigabitEthernet1/1/1, RPF nbr 172.16.12.1
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:00:17/00:02:42

(172.16.254.2, 225.0.0.101), 00:00:20/00:02:39, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    TenGigabitEthernet1/1/1, Forward/Sparse, 00:00:20/00:03:09
```

Leaf-02#

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

```
Leaf-02# 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,
                  e - Encap helper tunnel flag.
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
  TenGigabitEthernet1/1/1 Flags: A NS
  Loopback0 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2/0/141/0, Other: 0/0/0
  TenGigabitEthernet1/1/1 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.1,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/96/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  TenGigabitEthernet1/1/1 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(172.16.254.2,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/96/0, Other: 0/0/0
  HW Forwarding: 1/0/114/0, Other: 0/0/0
  Null0 Flags: A
  TenGigabitEthernet1/1/1 Flags: F NS
```

```

Pkts: 0/0/0    Rate: 0 pps
Tunnel1 Flags: F
Pkts: 0/0/1    Rate: 0 pps

```

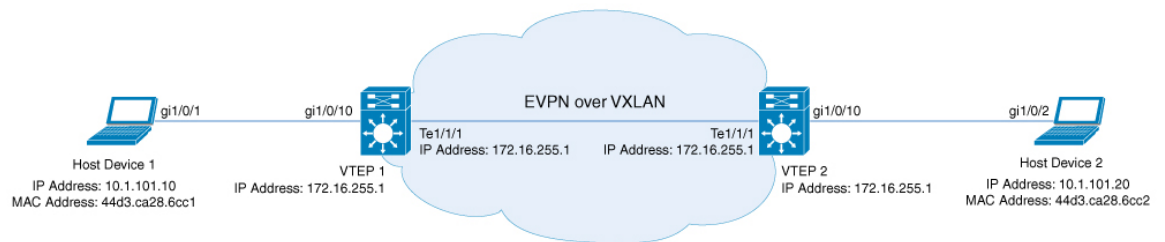
Leaf-02#

Return to [Verifying the Layer 2 VNI with Back-to-Back Multicast Replication](#), on page 27.

Example: Configuring Layer 2 VNI with Back to Back Ingress Replication

This example shows how to configure and verify a Layer 2 VNI with back-to-back ingress replication using the following topology:

Figure 2: EVPN VXLAN Network with a Layer 2 VNI with Ingress Replication



957017

The topology shows an EVPN VXLAN network with two VTEPs (VTEP 1 and VTEP 2) and no spine switches. Ingress replication is performed between the VTEPs to forward BUM traffic in the network. The following table provides sample configurations for the devices in this topology:



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: Configuring VTEP 1 and VTEP 2 to Configure a Layer 2 VNI with Back-to-Back Ingress Replication

VTEP 1	VTEP 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! ip routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type ingress ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! 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 GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access spanning-tree portfast ! interface TenGigabitEthernet1/1/1 no switchport ip address 172.16.12.1 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 ingress-replication ! </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! ip routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type ingress ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! 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 GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access spanning-tree portfast ! interface TenGigabitEthernet1/1/1 no switchport ip address 172.16.12.2 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 ingress-replication ! </pre>

VTEP 1	VTEP 2
<pre> router ospf 1 router-id 172.16.255.1 ! router bgp 65001 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 ! address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both exit-address-family ! end Leaf-01# </pre>	<pre> router ospf 1 router-id 172.16.255.2 ! 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 ! end Leaf-02# </pre>

Verifying the Layer 2 VNI with Back-to-Back Ingress Replication

The following sections provide sample outputs for **show** commands to verify the Layer 2 VNI with back-to-back ingress replication on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 27](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 30](#)

Outputs to Verify the Configuration on VTEP 1

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

```

Leaf-01# show nve peers
Interface  VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1      10101    L2CP 172.16.254.2     3             10101      UP    N/A  00:34:36
Leaf-01#
    
```

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

```

Leaf-01# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 34, main routing table version 34
9 network entries using 3456 bytes of memory
9 path entries using 1908 bytes of memory
4/4 BGP path/bestpath attribute entries using 1152 bytes of memory
1 BGP extended community entries using 40 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 6556 total bytes of memory
BGP activity 13/4 prefixes, 23/14 paths, scan interval 60 secs
9 networks peaked at 12:35:03 Oct 26 2020 UTC (00:34:37.010 ago)

Neighbor          V          AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
    
```

Example: Configuring Layer 2 VNI with Back to Back Ingress Replication

```

172.16.255.2      4          65001      213      215      34      0      0 03:06:17      3

Leaf-01#

```

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

```

Leaf-01# show bgp l2vpn evpn
BGP table version is 34, local router ID is 172.16.255.1
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: 172.16.254.1:101
*> [2][172.16.254.1:101][0][48][44D3CA286CC1][0][*]/20
      ::                                32768 ?
*> [2][172.16.254.1:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      ::                                32768 ?
*>i [2][172.16.254.1:101][0][48][44D3CA286CC2][0][*]/20
      172.16.254.2          0      100      0 ?
*>i [2][172.16.254.1:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      172.16.254.2          0      100      0 ?
Route Distinguisher: 172.16.254.2:101
*>i [2][172.16.254.2:101][0][48][44D3CA286CC2][0][*]/20
      172.16.254.2          0      100      0 ?
*>i [2][172.16.254.2:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      172.16.254.2          0      100      0 ?
Route Distinguisher: 172.16.254.1:101
*> [3][172.16.254.1:101][0][32][172.16.254.1]/17
      ::                                32768 ?
*>i [3][172.16.254.1:101][0][32][172.16.254.2]/17
      172.16.254.2          0      100      0 ?
Route Distinguisher: 172.16.254.2:101
*>i [3][172.16.254.2:101][0][32][172.16.254.2]/17
      172.16.254.2          0      100      0 ?

Leaf-01#

```

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

```

Leaf-01# show l2vpn evpn mac evi 101
MAC Address      EVI    VLAN  ESI                                Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc1  101    101   0000.0000.0000.0000.0000  0          Gi1/0/10:101
44d3.ca28.6cc2  101    101   0000.0000.0000.0000.0000  0          172.16.254.2

Leaf-01#

```

The following example shows the output for the **show l2fib bridge-domain evpn-instance detail** command on VTEP 1:

```

Leaf-01# show l2fib bridge-domain 101 detail
Bridge Domain : 101
Reference Count : 10
Replication ports count : 2
Unicast Address table size : 1
IP Multicast Prefix table size : 3

```



```

Flood List Information :
  Olist: 1125, Ports: 2

Port Information :
  BD_PORT   Gi1/0/10:101
  VXLAN_REP PL:25(1) T:VXLAN_REP [IR]10101:172.16.254.2

Unicast Address table information :
  44d3.ca28.6cc2  VXLAN_UC  PL:24(1) T:VXLAN_UC [MAC]10101:172.16.254.2

IP Multicast Prefix table information :
  Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 1125, Ports: 2
  Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 1125, Ports: 2
  Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 1125, Ports: 2

```

Leaf-01#

Return to [Verifying the Layer 2 VNI with Back-to-Back Multicast Replication, on page 27](#).

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI    state flags UP time
nve1     10101    L2CP 172.16.254.1      3         10101    UP     N/A  00:35:22

Leaf-02#

```

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

```

Leaf-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 34, main routing table version 34
9 network entries using 3456 bytes of memory
9 path entries using 1908 bytes of memory
4/4 BGP path/bestpath attribute entries using 1152 bytes of memory
1 BGP extended community entries using 40 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 6556 total bytes of memory
BGP activity 13/4 prefixes, 23/14 paths, scan interval 60 secs
9 networks peaked at 12:32:49 Oct 26 2020 UTC (00:34:55.476 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.255.1  4      65001   215    213     34     0    0 03:06:35      3

Leaf-02#

```

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

```

Leaf-02# show bgp l2vpn evpn
BGP table version is 34, local router ID is 172.16.255.2
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

```

Example: Configuring Layer 2 VNI with Back to Back Ingress Replication

```

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 172.16.254.1:101
*>i [2][172.16.254.1:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.1          0      100      0 ?
*>i [2][172.16.254.1:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.1          0      100      0 ?
Route Distinguisher: 172.16.254.2:101
*>i [2][172.16.254.2:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.1          0      100      0 ?
*>i [2][172.16.254.2:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.1          0      100      0 ?
*> [2][172.16.254.2:101][0][48][44D3CA286CC2][0][*]/20
      ::                      32768 ?
*> [2][172.16.254.2:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      ::                      32768 ?
Route Distinguisher: 172.16.254.1:101
*>i [3][172.16.254.1:101][0][32][172.16.254.1]/17
      172.16.254.1          0      100      0 ?
Route Distinguisher: 172.16.254.2:101
*>i [3][172.16.254.2:101][0][32][172.16.254.1]/17
      172.16.254.1          0      100      0 ?
*> [3][172.16.254.2:101][0][32][172.16.254.2]/17
      ::                      32768 ?

Leaf-02#

```

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

```

Leaf-02# show l2vpn evpn mac evi 101
MAC Address      EVI    VLAN  ESI                                     Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc1  101    101   0000.0000.0000.0000.0000  0          172.16.254.1
44d3.ca28.6cc2  101    101   0000.0000.0000.0000.0000  0          Gi1/0/10:101

```

```
Leaf-02#
```

The following example shows the output for the **show l2fib bridge-domain evpn-instance detail** command on VTEP 2:

```

Leaf-02# show l2fib bridge-domain 101 detail
Bridge Domain : 101
Reference Count : 10
Replication ports count : 2
Unicast Address table size : 1
IP Multicast Prefix table size : 3

Flood List Information :
Olist: 1125, Ports: 2

Port Information :
BD_PORT    Gi1/0/10:101
VXLAN_REP PL:16(1) T:VXLAN_REP [IR]10101:172.16.254.1

Unicast Address table information :
44d3.ca28.6cc1  VXLAN_UC  PL:15(1) T:VXLAN_UC [MAC]10101:172.16.254.1

IP Multicast Prefix table information :
Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 1125, Ports: 2
Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 1125, Ports: 2
Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 1125, Ports: 2

```

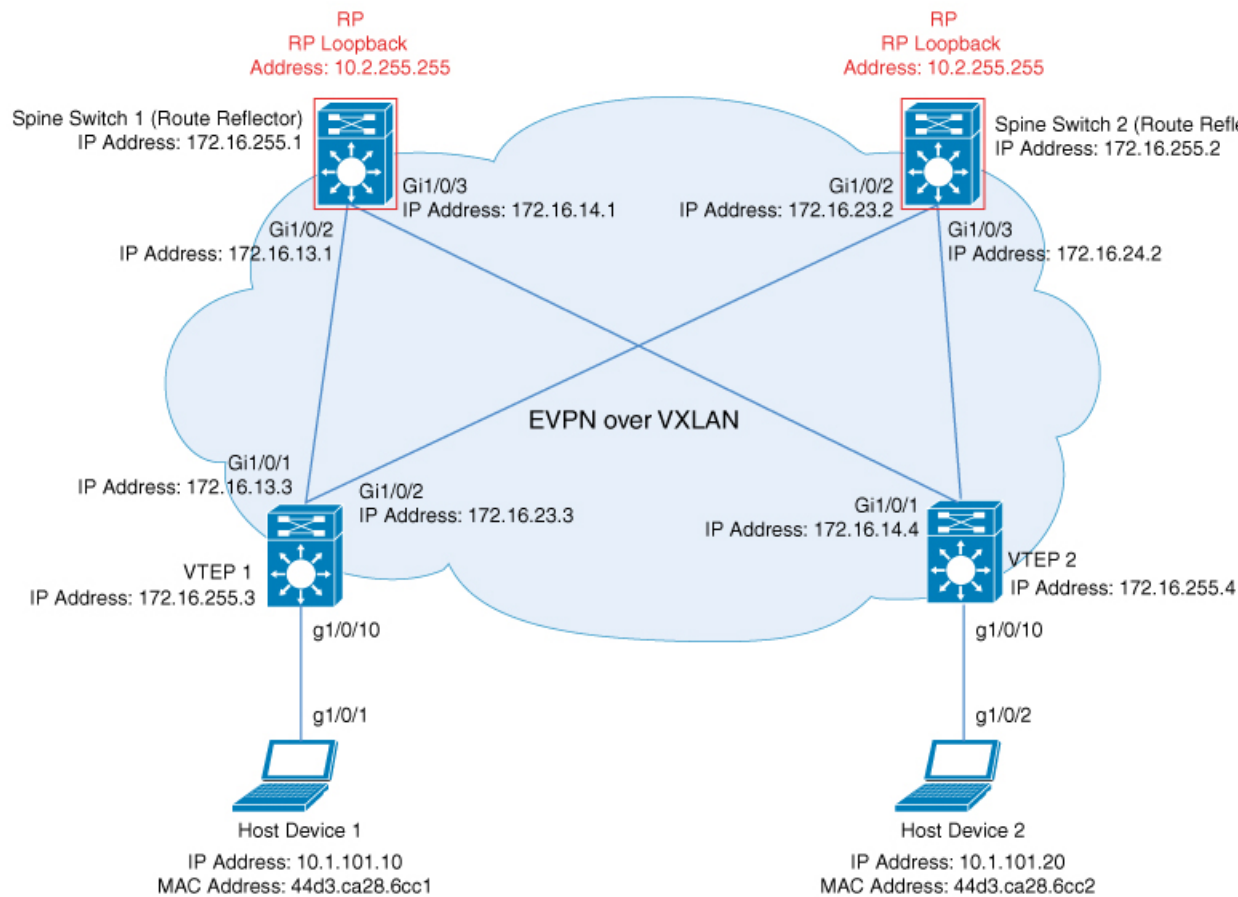
Leaf-02#

Return to [Verifying the Layer 2 VNI with Back-to-Back Multicast Replication](#), on page 27.

Example: Configuring Layer 2 VNI with Spine Multicast Replication

This example shows how to configure and verify a Layer 2 VNI with spine multicast replication using the following topology:

Figure 3: EVPN VXLAN Network with a Layer 2 VNI with Multicast Replication



The topology shows an EVPN VXLAN network with two spine switches (Spine Switch 1 and Spine Switch 2) and two VTEPs (VTEP 1 and VTEP 2). Multicast replication is performed between the VTEPs to forward BUM traffic in the network. Spine Switch 1 and Spine Switch 2 act as route reflectors and also as the RPs for the multicast BUM traffic in the network. The following tables provide sample configurations for the devices in this topology:

Table 4: Configuring VTEP 1 and VTEP 2 to Configure a Layer 2 VNI with Spine Multicast Replication

VTEP 1	VTEP 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access spanning-tree portfast ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 ! router ospf 1 router-id 172.16.255.3 ! </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! ip routing ! ip multicast-routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access spanning-tree portfast ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 mcast-group 225.0.0.101 ! router ospf 1 router-id 172.16.255.4 ! </pre>

VTEP 1	VTEP 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 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 ! ip pim rp-address 172.16.255.255 ! end Leaf-01#</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 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 ! ip pim rp-address 172.16.255.255 ! end Leaf-02#</pre>

Table 5: Configuring VTEP 1 and VTEP 2 to Configure a Layer 2 VNI with Spine Multicast Replication

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 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 GigabitEthernet1/0/2 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 GigabitEthernet1/0/3 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 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 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 ! address-family ipv4 exit-address-family ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 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 GigabitEthernet1/0/2 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 GigabitEthernet1/0/3 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 ! 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 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 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 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 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>

Verifying the Layer 2 VNI with Spine Multicast Replication

The following sections provide sample outputs for **show** commands to verify the Layer 2 VNI with spine multicast replication on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 43](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 47](#)
- [Outputs to Verify the Configuration on Spine Switch 1 \(RP inside the Network\), on page 50](#)
- [Outputs to Verify the Configuration on Spine Switch 2 \(RP inside the Network\), on page 54](#)

Outputs to Verify the Configuration on VTEP 1

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

```

Leaf-01# show nve peers
Interface  VNI      Type Peer-IP           RMAC/Num_RTs  eVNI      state flags UP time
nve1      10101  L2CP 172.16.254.4        2              10101      UP    N/A  00:00:56

Leaf-01#
    
```

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

```

Leaf-01# show ip route
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
       & - replicated local route overrides by connected
    
```

Example: Configuring Layer 2 VNI with Spine Multicast Replication

```

Gateway of last resort is not set

    172.16.0.0/16 is variably subnetted, 15 subnets, 2 masks
C       172.16.13.0/24 is directly connected, GigabitEthernet1/0/1
L       172.16.13.3/32 is directly connected, GigabitEthernet1/0/1
O       172.16.14.0/24
        [110/2] via 172.16.13.1, 01:43:35, GigabitEthernet1/0/1
C       172.16.23.0/24 is directly connected, GigabitEthernet1/0/2
L       172.16.23.3/32 is directly connected, GigabitEthernet1/0/2
O       172.16.24.0/24
        [110/2] via 172.16.23.2, 01:43:35, GigabitEthernet1/0/2
O       172.16.254.1/32
        [110/2] via 172.16.13.1, 00:09:33, GigabitEthernet1/0/1
O       172.16.254.2/32
        [110/2] via 172.16.23.2, 00:08:17, GigabitEthernet1/0/2
C       172.16.254.3/32 is directly connected, Loopback1
O       172.16.254.4/32
        [110/3] via 172.16.23.2, 01:43:35, GigabitEthernet1/0/2
        [110/3] via 172.16.13.1, 01:43:35, GigabitEthernet1/0/1
O       172.16.255.1/32
        [110/2] via 172.16.13.1, 01:43:35, GigabitEthernet1/0/1
O       172.16.255.2/32
        [110/2] via 172.16.23.2, 01:43:35, GigabitEthernet1/0/2
C       172.16.255.3/32 is directly connected, Loopback0
O       172.16.255.4/32
        [110/3] via 172.16.23.2, 01:43:35, GigabitEthernet1/0/2
        [110/3] via 172.16.13.1, 01:43:35, GigabitEthernet1/0/1
O       172.16.255.255/32
        [110/2] via 172.16.23.2, 00:08:17, GigabitEthernet1/0/2
        [110/2] via 172.16.13.1, 00:09:33, GigabitEthernet1/0/1

```

```
Leaf-01#
```

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

```

Leaf-01# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 54, main routing table version 54
6 network entries using 2304 bytes of memory
8 path entries using 1696 bytes of memory
2/2 BGP path/bestpath attribute entries using 576 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 4696 total bytes of memory
BGP activity 15/9 prefixes, 33/25 paths, scan interval 60 secs
9 networks peaked at 16:10:51 Oct 26 2020 UTC (01:42:36.958 ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	133	120	54	0	0	01:43:34	2
172.16.255.2	4	65001	134	123	54	0	0	01:43:34	2

```
Leaf-01#
```

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

```

Leaf-01# show bgp l2vpn evpn
BGP table version is 54, local router ID is 172.16.255.3
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: 172.16.254.3:101
*> [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
    ::                                32768 ?
*> [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
    ::                                32768 ?
*>i [2][172.16.254.3:101][0][48][44D3CA286CC2][0][*]/20
    172.16.254.4                      0 100 0 ?
*>i [2][172.16.254.3:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
    172.16.254.4                      0 100 0 ?
Route Distinguisher: 172.16.254.4:101
*>i [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
    172.16.254.4                      0 100 0 ?
* i 172.16.254.4                      0 100 0 ?
*>i [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
    172.16.254.4                      0 100 0 ?
* i 172.16.254.4                      0 100 0 ?

```

Leaf-01#

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

```

Leaf-01# show l2vpn evpn mac evi 101
MAC Address      EVI    VLAN  ESI                                Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc1  101    101   0000.0000.0000.0000.0000  0          Gi1/0/10:101
44d3.ca28.6cc2  101    101   0000.0000.0000.0000.0000  0          172.16.254.4

```

Leaf-01#

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

```

Leaf-01# 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,
       e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:05:22/00:02:42, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:

```

Example: Configuring Layer 2 VNI with Spine Multicast Replication

```

Loopback1, Forward/Sparse, 00:05:20/00:02:42

(*, 225.0.0.101), 00:01:34/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, 00:01:34/00:01:27

(172.16.254.4, 225.0.0.101), 00:00:57/00:02:02, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
  Tunnel0, Forward/Sparse-Dense, 00:00:57/00:02:02

(172.16.254.3, 225.0.0.101), 00:01:32/00:01:27, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0, Registering
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 00:01:32/00:02:57

Leaf-01#

```

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

```

Leaf-01# 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,
             e - Encap helper tunnel flag.
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
  GigabitEthernet1/0/2 Flags: A NS
  Loopback1 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/114/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 1/1/0
  HW Forwarding: 148/0/155/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  Tunnel1 Flags: F
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.101) Flags: HW

```

```

SW Forwarding: 1/0/96/0, Other: 0/0/0
HW Forwarding: 2/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0/1 Rate: 0 pps

```

```
Leaf-01#
```

Return to [Verifying the Layer 2 VNI with Spine Multicast Replication](#), on page 43.

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     10101    L2CP 172.16.254.3    2           10101      UP    N/A  00:01:39

Leaf-02#

```

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

```

Leaf-02# show ip route
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
       & - replicated local route overrides by connected

```

```
Gateway of last resort is not set
```

```

      172.16.0.0/16 is variably subnetted, 15 subnets, 2 masks
O       172.16.13.0/24
        [110/2] via 172.16.14.1, 01:44:23, GigabitEthernet1/0/1
C       172.16.14.0/24 is directly connected, GigabitEthernet1/0/1
L       172.16.14.4/32 is directly connected, GigabitEthernet1/0/1
O       172.16.23.0/24
        [110/2] via 172.16.24.2, 01:44:23, GigabitEthernet1/0/2
C       172.16.24.0/24 is directly connected, GigabitEthernet1/0/2
L       172.16.24.4/32 is directly connected, GigabitEthernet1/0/2
O       172.16.254.1/32
        [110/2] via 172.16.14.1, 00:10:18, GigabitEthernet1/0/1
O       172.16.254.2/32
        [110/2] via 172.16.24.2, 00:09:02, GigabitEthernet1/0/2
O       172.16.254.3/32
        [110/3] via 172.16.24.2, 01:44:20, GigabitEthernet1/0/2
        [110/3] via 172.16.14.1, 01:44:15, GigabitEthernet1/0/1
C       172.16.254.4/32 is directly connected, Loopback1
O       172.16.255.1/32
        [110/2] via 172.16.14.1, 01:44:23, GigabitEthernet1/0/1
O       172.16.255.2/32
        [110/2] via 172.16.24.2, 01:44:23, GigabitEthernet1/0/2
O       172.16.255.3/32
        [110/3] via 172.16.24.2, 01:44:20, GigabitEthernet1/0/2
        [110/3] via 172.16.14.1, 01:44:15, GigabitEthernet1/0/1

```

Example: Configuring Layer 2 VNI with Spine Multicast Replication

```

C      172.16.255.4/32 is directly connected, Loopback0
O      172.16.255.255/32
       [110/2] via 172.16.24.2, 00:09:01, GigabitEthernet1/0/2
       [110/2] via 172.16.14.1, 00:10:18, GigabitEthernet1/0/1

```

```
Leaf-02#
```

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

```

Leaf-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 54, main routing table version 54
6 network entries using 2304 bytes of memory
8 path entries using 1696 bytes of memory
2/2 BGP path/bestpath attribute entries using 576 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 4696 total bytes of memory
BGP activity 15/9 prefixes, 34/26 paths, scan interval 60 secs
9 networks peaked at 16:08:37 Oct 26 2020 UTC (01:43:22.226 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   134    123     54    0    0 01:44:22      2
172.16.255.2  4      65001   134    123     54    0    0 01:44:15      2

```

```
Leaf-02#
```

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

```

Leaf-02# show bgp l2vpn evpn
BGP table version is 54, local router ID is 172.16.255.4
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: 172.16.254.3:101
* i [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
*>i [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3      0 100 0 ?
* i      172.16.254.3      0 100 0 ?
Route Distinguisher: 172.16.254.4:101
*>i [2][172.16.254.4:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3      0 100 0 ?
*>i [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3      0 100 0 ?
*> [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
      :: 32768 ?
*> [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      :: 32768 ?

```

```
Leaf-02#
```

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

```
Leaf-02# show l2vpn evpn mac evi 101
MAC Address      EVI      VLAN  ESI                               Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc1 101      101   0000.0000.0000.0000.0000 0          172.16.254.3
44d3.ca28.6cc2 101      101   0000.0000.0000.0000.0000 0          Gi1/0/10:101

Leaf-02#
```

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

```
Leaf-02# 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,
       e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:05:51/00:02:24, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Loopback1, Forward/Sparse, 00:05:49/00:02:09
    GigabitEthernet1/0/1, Forward/Sparse, 00:05:43/00:02:24

(*, 225.0.0.101), 00:02:46/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, 00:02:46/00:00:15

(172.16.254.4, 225.0.0.101), 00:01:43/00:01:16, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:01:43/00:02:45

(172.16.254.3, 225.0.0.101), 00:02:19/00:00:40, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:02:19/00:00:40

Leaf-02#
```

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

```
Leaf-02# 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
```

Example: Configuring Layer 2 VNI with Spine Multicast Replication

```

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,
e - Encap helper tunnel flag.
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
  GigabitEthernet1/0/2 Flags: A NS
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  Loopback1 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  1/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding:  146/0/167/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(172.16.254.4,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/96/0, Other: 1/1/0
  HW Forwarding:  4/0/145/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps

```

Leaf-02#

Return to [Verifying the Layer 2 VNI with Spine Multicast Replication, on page 43](#).

Outputs to Verify the Configuration on Spine Switch 1 (RP inside the Network)

The following example shows the output for the **show ip route** command on Spine Switch 1:

```

Spine-01# show ip route
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

    172.16.0.0/16 is variably subnetted, 15 subnets, 2 masks
C       172.16.13.0/24 is directly connected, GigabitEthernet1/0/2
L       172.16.13.1/32 is directly connected, GigabitEthernet1/0/2
C       172.16.14.0/24 is directly connected, GigabitEthernet1/0/3
L       172.16.14.1/32 is directly connected, GigabitEthernet1/0/3
O       172.16.23.0/24
        [110/2] via 172.16.13.3, 01:45:08, GigabitEthernet1/0/2
O       172.16.24.0/24
        [110/2] via 172.16.14.4, 01:45:12, GigabitEthernet1/0/3
C       172.16.254.1/32 is directly connected, Loopback1
O       172.16.254.2/32
        [110/3] via 172.16.14.4, 00:09:51, GigabitEthernet1/0/3
        [110/3] via 172.16.13.3, 00:09:51, GigabitEthernet1/0/2
O       172.16.254.3/32
        [110/2] via 172.16.13.3, 01:45:08, GigabitEthernet1/0/2
O       172.16.254.4/32
        [110/2] via 172.16.14.4, 01:45:12, GigabitEthernet1/0/3
C       172.16.255.1/32 is directly connected, Loopback0
O       172.16.255.2/32
        [110/3] via 172.16.14.4, 01:45:12, GigabitEthernet1/0/3
        [110/3] via 172.16.13.3, 01:45:08, GigabitEthernet1/0/2
O       172.16.255.3/32
        [110/2] via 172.16.13.3, 01:45:08, GigabitEthernet1/0/2
O       172.16.255.4/32
        [110/2] via 172.16.14.4, 01:45:12, GigabitEthernet1/0/3
C       172.16.255.255/32 is directly connected, Loopback2

Spine-01#

```

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 35, main routing table version 35
4 network entries using 1376 bytes of memory
8 path entries using 1664 bytes of memory
1/1 BGP path/bestpath attribute entries using 288 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 3448 total bytes of memory
BGP activity 12/8 prefixes, 28/20 paths, scan interval 60 secs
6 networks peaked at 16:08:39 Oct 26 2020 UTC (01:44:10.445 ago)

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2  4      65001   133     132      35    0    0 01:45:07      4
172.16.255.3  4      65001   122     135      35    0    0 01:45:07      2
172.16.255.4  4      65001   124     135      35    0    0 01:45:10      2

Spine-01#

```

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

```

Spine-01# show bgp l2vpn evpn
BGP table version is 35, local router ID is 172.16.255.1

```

Example: Configuring Layer 2 VNI with Spine Multicast Replication

```

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: 172.16.254.3:101
* i [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3          0      100      0 ?
*>i      172.16.254.3          0      100      0 ?
* i [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3          0      100      0 ?
*>i      172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
* i [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
      172.16.254.4          0      100      0 ?
*>i      172.16.254.4          0      100      0 ?
* i [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      172.16.254.4          0      100      0 ?
*>i      172.16.254.4          0      100      0 ?

```

```
Spine-01#
```

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
                  AS              Downtime Count Count
172.16.254.2      65001 Up        00:06:28 0      0      ?
Spine-01#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 1:

```

Spine-01# 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

(*, 224.0.1.40), 00:56:14/00:02:21, RP 172.16.255.255, flags: SPL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(*, 225.0.0.101), 00:00:12/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), 00:00:05/00:02:54, flags: PA
Incoming interface: GigabitEthernet1/0/3, RPF nbr 172.16.14.4
Outgoing interface list: Null

(172.16.254.3, 225.0.0.101), 00:00:12/00:02:47, flags: PA
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.13.3
Outgoing interface list: Null

Spine-01#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 1:

```

Spine-01# 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: 2/2/0
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
Tunnell Flags: A
GigabitEthernet1/0/3 Flags: IC
(*,225.0.0.101) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 1/0/1
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
(172.16.254.3,225.0.0.101) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/2 Flags: NS
(172.16.254.4,225.0.0.101) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/3 Flags: NS

Spine-01#

```

Return to [Verifying the Layer 2 VNI with Spine Multicast Replication](#), on page 43.

Outputs to Verify the Configuration on Spine Switch 2 (RP inside the Network)

The following example shows the output for the **show ip route** command on Spine Switch 2:

```
Spine-02# show ip route
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

    172.16.0.0/16 is variably subnetted, 15 subnets, 2 masks
O       172.16.13.0/24
        [110/2] via 172.16.23.3, 01:45:34, GigabitEthernet1/0/2
O       172.16.14.0/24
        [110/2] via 172.16.24.4, 01:45:38, GigabitEthernet1/0/3
C       172.16.23.0/24 is directly connected, GigabitEthernet1/0/2
L       172.16.23.2/32 is directly connected, GigabitEthernet1/0/2
C       172.16.24.0/24 is directly connected, GigabitEthernet1/0/3
L       172.16.24.2/32 is directly connected, GigabitEthernet1/0/3
O       172.16.254.1/32
        [110/3] via 172.16.24.4, 00:11:33, GigabitEthernet1/0/3
        [110/3] via 172.16.23.3, 00:11:33, GigabitEthernet1/0/2
C       172.16.254.2/32 is directly connected, Loopback1
O       172.16.254.3/32
        [110/2] via 172.16.23.3, 01:45:34, GigabitEthernet1/0/2
O       172.16.254.4/32
        [110/2] via 172.16.24.4, 01:45:38, GigabitEthernet1/0/3
O       172.16.255.1/32
        [110/3] via 172.16.24.4, 01:45:34, GigabitEthernet1/0/3
        [110/3] via 172.16.23.3, 01:45:30, GigabitEthernet1/0/2
C       172.16.255.2/32 is directly connected, Loopback0
O       172.16.255.3/32
        [110/2] via 172.16.23.3, 01:45:34, GigabitEthernet1/0/2
O       172.16.255.4/32
        [110/2] via 172.16.24.4, 01:45:38, GigabitEthernet1/0/3
C       172.16.255.255/32 is directly connected, Loopback2

Spine-02#
```

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 35, main routing table version 35
4 network entries using 1376 bytes of memory
8 path entries using 1664 bytes of memory
1/1 BGP path/bestpath attribute entries using 288 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 3448 total bytes of memory
BGP activity 10/6 prefixes, 28/20 paths, scan interval 60 secs
```

6 networks peaked at 16:09:46 Oct 26 2020 UTC (01:44:35.591 ago)

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	133	134	35	0	0	01:45:33	4
172.16.255.3	4	65001	125	137	35	0	0	01:45:33	2
172.16.255.4	4	65001	125	136	35	0	0	01:45:28	2

Spine-02#

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

```
Spine-02# show bgp l2vpn evpn
BGP table version is 35, local router ID is 172.16.255.2
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: 172.16.254.3:101
 * i [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3          0      100      0 ?
 *>i          172.16.254.3          0      100      0 ?
 * i [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3          0      100      0 ?
 *>i          172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
 * i [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
      172.16.254.4          0      100      0 ?
 *>i          172.16.254.4          0      100      0 ?
 * i [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      172.16.254.4          0      100      0 ?
 *>i          172.16.254.4          0      100      0 ?

```

Spine-02#

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         00:06:53 0      2      ?
```

Spine-02#

The following example shows the output for the **show ip mroute** command on Spine Switch 2:

```
Spine-02# 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,
```

Example: Configuring Layer 2 VNI with Spine Multicast Replication

```

    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

(*, 224.0.1.40), 00:56:18/00:03:26, RP 172.16.255.255, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 00:54:14/00:03:08
  GigabitEthernet1/0/3, Forward/Sparse, 00:56:18/00:03:26

(*, 225.0.0.101), 00:51:00/00:03:17, RP 172.16.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 00:50:34/00:03:17
  GigabitEthernet1/0/3, Forward/Sparse, 00:51:00/00:02:43

(172.16.254.4, 225.0.0.101), 00:00:17/00:02:42, flags: TA
Incoming interface: GigabitEthernet1/0/3, RPF nbr 172.16.24.4
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 00:00:17/00:03:17

(172.16.254.3, 225.0.0.101), 00:00:23/00:02:36, flags: TA
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.3
Outgoing interface list:
  GigabitEthernet1/0/3, Forward/Sparse, 00:00:23/00:03:06

Spine-02#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 2:

```

Spine-02# 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
Tunnell Flags: A
GigabitEthernet1/0/3 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS

```

```

Pkts: 0/0/0    Rate: 0 pps
(*,225.0.0.101) Flags: C HW
SW Forwarding: 2/0/150/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/3 Flags: F NS
Pkts: 0/0/2    Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
Pkts: 0/0/2    Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
GigabitEthernet1/0/3 Flags: F NS
Pkts: 0/0/0    Rate: 0 pps
(172.16.254.4,225.0.0.101) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
GigabitEthernet1/0/3 Flags: A
GigabitEthernet1/0/2 Flags: F NS
Pkts: 0/0/0    Rate: 0 pps
    
```

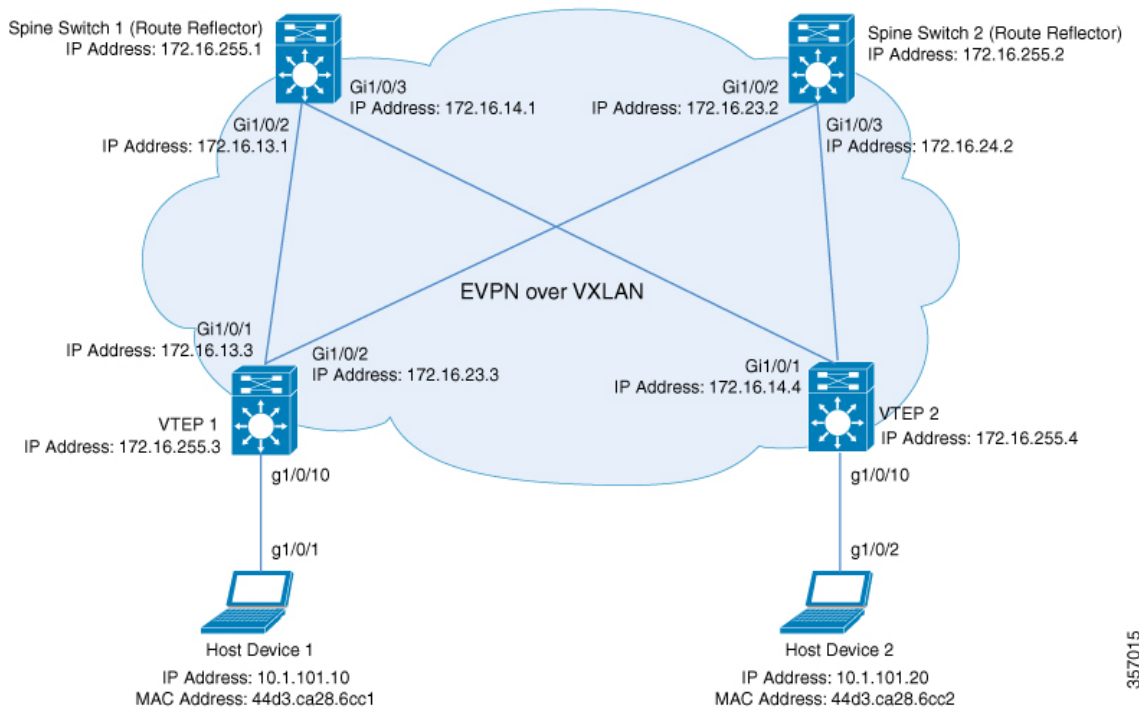
Spine-02#

Return to [Verifying the Layer 2 VNI with Spine Multicast Replication](#), on page 43.

Example: Configuring Layer 2 VNI with Spine Ingress Replication

This example shows how to configure and verify a Layer 2 VNI with spine ingress replication using the following topology:

Figure 4: EVPN VXLAN Network with a Layer 2 VNI with Ingress Replication



357015

The topology shows an EVPN VXLAN network with two spine switches (Spine Switch 1 and Spine Switch 2) and two VTEPs (VTEP 1 and VTEP 2). Ingress replication is performed between the VTEPs to forward BUM traffic in the network. Spine Switch 1 and Spine Switch 2 act as route reflectors in the network. The following tables provide sample configurations for the devices in this topology:

Table 6: Configuring VTEP 1 and VTEP 2 to Configure a Layer 2 VNI with Spine Ingress Replication

VTEP 1	VTEP 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! ip routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type ingress ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! 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 ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.3 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.23.3 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access spanning-tree portfast ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 ingress-replication ! router ospf 1 router-id 172.16.255.3 ! </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! ip routing ! l2vpn evpn replication-type static router-id Loopback1 ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type ingress ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! 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 ospf 1 area 0 ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.4 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access spanning-tree portfast ! interface nve1 no ip address source-interface Loopback1 host-reachability protocol bgp member vni 10101 ingress-replication ! router ospf 1 router-id 172.16.255.4 ! </pre>

VTEP 1	VTEP 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 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 ! end Leaf-01#</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 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 ! end Leaf-02#</pre>

Table 7: Configuring VTEP 1 and VTEP 2 to Configure a Layer 2 VNI with Spine Ingress Replication

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! system mtu 9198 ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 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 GigabitEthernet1/0/3 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 neighbor 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 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 ! address-family ipv4 exit-address-family ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! system mtu 9198 ! interface Loopback0 ip address 172.16.255.2 255.255.255.255 ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 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 GigabitEthernet1/0/3 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 ! address-family ipv4 exit-address-family ! </pre>
<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 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 exit-address-family ! 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 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 exit-address-family ! end Spine-02# </pre>

Verifying the Layer 2 VNI with Spine Ingress Replication

The following sections provide sample outputs for **show** commands to verify the Layer 2 VNI with spine ingress replication on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 61](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 63](#)
- [Outputs to Verify the Configuration on Spine Switch 1, on page 66](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 68](#)

Outputs to Verify the Configuration on VTEP 1

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

```
Leaf-01# show nve peers
Interface  VNI      Type Peer-IP          RMAC/Num_RTs  eVNI    state flags UP time
nve1      10101    L2CP 172.16.254.4      3           10101    UP    N/A  01:25:20

Leaf-01#
```

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

```
Leaf-01# show ip route
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
       & - replicated local route overrides by connected

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 12 subnets, 2 masks
C       172.16.13.0/24 is directly connected, GigabitEthernet1/0/1
L       172.16.13.3/32 is directly connected, GigabitEthernet1/0/1
O       172.16.14.0/24
        [110/2] via 172.16.13.1, 01:26:20, GigabitEthernet1/0/1
C       172.16.23.0/24 is directly connected, GigabitEthernet1/0/2
L       172.16.23.3/32 is directly connected, GigabitEthernet1/0/2
O       172.16.24.0/24
        [110/2] via 172.16.23.2, 01:26:20, GigabitEthernet1/0/2
C       172.16.254.3/32 is directly connected, Loopback1
O       172.16.254.4/32
        [110/3] via 172.16.23.2, 01:26:20, GigabitEthernet1/0/2
        [110/3] via 172.16.13.1, 01:26:20, GigabitEthernet1/0/1
O       172.16.255.1/32
        [110/2] via 172.16.13.1, 01:26:20, GigabitEthernet1/0/1
O       172.16.255.2/32
        [110/2] via 172.16.23.2, 01:26:20, GigabitEthernet1/0/2
C       172.16.255.3/32 is directly connected, Loopback0
O       172.16.255.4/32
        [110/3] via 172.16.23.2, 01:26:20, GigabitEthernet1/0/2
```

Example: Configuring Layer 2 VNI with Spine Ingress Replication

```
[110/3] via 172.16.13.1, 01:26:20, GigabitEthernet1/0/1
```

```
Leaf-01#
```

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

```
Leaf-01# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 13, main routing table version 13
9 network entries using 3456 bytes of memory
12 path entries using 2544 bytes of memory
4/4 BGP path/bestpath attribute entries using 1152 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 7272 total bytes of memory
BGP activity 9/0 prefixes, 15/3 paths, scan interval 60 secs
9 networks peaked at 16:10:51 Oct 26 2020 UTC (01:25:22.020 ago)

Neighbor      V          AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State/PfxRcd
172.16.255.1  4          65001    101    99      13    0    0 01:26:19      3
172.16.255.2  4          65001    102    100     13    0    0 01:26:19      3

Leaf-01#
```

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

```
Leaf-01# show bgp l2vpn evpn
BGP table version is 13, local router ID is 172.16.255.3
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: 172.16.254.3:101
*> [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
   ::                    32768 ?
*> [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
   ::                    32768 ?
*>i [2][172.16.254.3:101][0][48][44D3CA286CC2][0][*]/20
   172.16.254.4          0 100 0 ?
*>i [2][172.16.254.3:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
   172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.254.4:101
* i [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
   172.16.254.4          0 100 0 ?
*>i 172.16.254.4          0 100 0 ?
* i [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
   172.16.254.4          0 100 0 ?
*>i 172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.254.3:101
*> [3][172.16.254.3:101][0][32][172.16.254.3]/17
   ::                    32768 ?
*>i [3][172.16.254.3:101][0][32][172.16.254.4]/17
   172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.254.4:101
* i [3][172.16.254.4:101][0][32][172.16.254.4]/17
```

```

172.16.254.4          0    100    0 ?
*>i                 172.16.254.4      0    100    0 ?

Leaf-01#

```

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

```

Leaf-01# show l2vpn evpn mac evi 101
MAC Address      EVI    VLAN  ESI                               Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc1  101    101   0000.0000.0000.0000.0000  0          Gi1/0/10:101
44d3.ca28.6cc2  101    101   0000.0000.0000.0000.0000  0          172.16.254.4

Leaf-01#

```

The following example shows the output for the **show l2fib bridge-domain evpn-instance detail** command on VTEP 1:

```

Leaf-01# show l2fib bridge-domain 101 detail
Bridge Domain : 101
  Reference Count : 10
  Replication ports count : 2
  Unicast Address table size : 1
  IP Multicast Prefix table size : 3

Flood List Information :
  Olist: 1125, Ports: 2

Port Information :
  BD_PORT   Gi1/0/10:101
  VXLAN_REP PL:2(1) T:VXLAN_REP [IR]10101:172.16.254.4

Unicast Address table information :
  44d3.ca28.6cc2  VXLAN_UC  PL:1(1) T:VXLAN_UC [MAC]10101:172.16.254.4

IP Multicast Prefix table information :
  Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 1125, Ports: 2
  Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 1125, Ports: 2
  Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 1125, Ports: 2

Leaf-01#

```

Return to [Verifying the Layer 2 VNI with Spine Ingress Replication, on page 61](#).

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface  VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1      10101    L2CP 172.16.254.3       3            10101      UP      N/A  01:27:15

Leaf-02#

```

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

Example: Configuring Layer 2 VNI with Spine Ingress Replication

```
Leaf-02# show ip route
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
       & - replicated local route overrides by connected
```

```
Gateway of last resort is not set
```

```
172.16.0.0/16 is variably subnetted, 12 subnets, 2 masks
O    172.16.13.0/24
     [110/2] via 172.16.14.1, 01:28:18, GigabitEthernet1/0/1
C    172.16.14.0/24 is directly connected, GigabitEthernet1/0/1
L    172.16.14.4/32 is directly connected, GigabitEthernet1/0/1
O    172.16.23.0/24
     [110/2] via 172.16.24.2, 01:28:18, GigabitEthernet1/0/2
C    172.16.24.0/24 is directly connected, GigabitEthernet1/0/2
L    172.16.24.4/32 is directly connected, GigabitEthernet1/0/2
O    172.16.254.3/32
     [110/3] via 172.16.24.2, 01:28:15, GigabitEthernet1/0/2
     [110/3] via 172.16.14.1, 01:28:10, GigabitEthernet1/0/1
C    172.16.254.4/32 is directly connected, Loopback1
O    172.16.255.1/32
     [110/2] via 172.16.14.1, 01:28:18, GigabitEthernet1/0/1
O    172.16.255.2/32
     [110/2] via 172.16.24.2, 01:28:18, GigabitEthernet1/0/2
O    172.16.255.3/32
     [110/3] via 172.16.24.2, 01:28:15, GigabitEthernet1/0/2
     [110/3] via 172.16.14.1, 01:28:10, GigabitEthernet1/0/1
C    172.16.255.4/32 is directly connected, Loopback0
```

```
Leaf-02#
```

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

```
Leaf-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 13, main routing table version 13
9 network entries using 3456 bytes of memory
12 path entries using 2544 bytes of memory
4/4 BGP path/bestpath attribute entries using 1152 bytes of memory
2 BGP rinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 7272 total bytes of memory
BGP activity 9/0 prefixes, 15/3 paths, scan interval 60 secs
9 networks peaked at 16:08:37 Oct 26 2020 UTC (01:27:15.987 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   103    101     13    0    0 01:28:16      3
172.16.255.2  4      65001   103    101     13    0    0 01:28:09      3

Leaf-02#
```

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

```
Leaf-02# show bgp l2vpn evpn
BGP table version is 13, local router ID is 172.16.255.4
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: 172.16.254.3:101
* i   [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3          0      100      0 ?
*>i   172.16.254.3          0      100      0 ?
* i   [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3          0      100      0 ?
*>i   172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
*>i   [2][172.16.254.4:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3          0      100      0 ?
*>i   [2][172.16.254.4:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3          0      100      0 ?
*>    [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
      ::                      32768 ?
*>    [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      ::                      32768 ?
Route Distinguisher: 172.16.254.3:101
* i   [3][172.16.254.3:101][0][32][172.16.254.3]/17
      172.16.254.3          0      100      0 ?
*>i   172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
*>i   [3][172.16.254.4:101][0][32][172.16.254.3]/17
      172.16.254.3          0      100      0 ?
*>    [3][172.16.254.4:101][0][32][172.16.254.4]/17
      ::                      32768 ?

Leaf-02#
```

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

```
Leaf-02# show l2vpn evpn mac evi 101
MAC Address   EVI   VLAN   ESI                               Ether Tag   Next Hop(s)
-----
44d3.ca28.6cc1 101   101   0000.0000.0000.0000.0000.0000  0           172.16.254.3
44d3.ca28.6cc2 101   101   0000.0000.0000.0000.0000.0000  0           Gi1/0/10:101

Leaf-02#
```

The following example shows the output for the **show l2fib bridge-domain evpn-instance detail** command on VTEP 2:

```
Leaf-02# show l2fib bridge-domain 101 detail
Bridge Domain : 101
Reference Count : 10
Replication ports count : 2
Unicast Address table size : 1
IP Multicast Prefix table size : 3
```

Example: Configuring Layer 2 VNI with Spine Ingress Replication

```

Flood List Information :
  Olist: 1125, Ports: 2

Port Information :
  BD_PORT    Gi1/0/10:101
  VXLAN_REP  PL:2(1) T:VXLAN_REP [IR]10101:172.16.254.3

Unicast Address table information :
  44d3.ca28.6cc1  VXLAN_UC  PL:1(1) T:VXLAN_UC [MAC]10101:172.16.254.3

IP Multicast Prefix table information :
  Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 1125, Ports: 2
  Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 1125, Ports: 2
  Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 1125, Ports: 2

```

```
Leaf-02#
```

Return to [Verifying the Layer 2 VNI with Spine Ingress Replication, on page 61](#).

Outputs to Verify the Configuration on Spine Switch 1

The following example shows the output for the **show ip route** command on Spine Switch 1:

```

Spine-01# show ip route
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

    172.16.0.0/16 is variably subnetted, 12 subnets, 2 masks
C       172.16.13.0/24 is directly connected, GigabitEthernet1/0/2
L       172.16.13.1/32 is directly connected, GigabitEthernet1/0/2
C       172.16.14.0/24 is directly connected, GigabitEthernet1/0/3
L       172.16.14.1/32 is directly connected, GigabitEthernet1/0/3
O       172.16.23.0/24
        [110/2] via 172.16.13.3, 01:29:42, GigabitEthernet1/0/2
O       172.16.24.0/24
        [110/2] via 172.16.14.4, 01:29:46, GigabitEthernet1/0/3
O       172.16.254.3/32
        [110/2] via 172.16.13.3, 01:29:42, GigabitEthernet1/0/2
O       172.16.254.4/32
        [110/2] via 172.16.14.4, 01:29:46, GigabitEthernet1/0/3
C       172.16.255.1/32 is directly connected, Loopback0
O       172.16.255.2/32
        [110/3] via 172.16.14.4, 01:29:46, GigabitEthernet1/0/3
        [110/3] via 172.16.13.3, 01:29:42, GigabitEthernet1/0/2
O       172.16.255.3/32
        [110/2] via 172.16.13.3, 01:29:42, GigabitEthernet1/0/2
O       172.16.255.4/32
        [110/2] via 172.16.14.4, 01:29:46, GigabitEthernet1/0/3

Spine-01#

```

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 7, main routing table version 7
6 network entries using 2064 bytes of memory
12 path entries using 2496 bytes of memory
3/3 BGP path/bestpath attribute entries using 864 bytes of memory
2 BGP rrinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 5544 total bytes of memory
BGP activity 6/0 prefixes, 12/0 paths, scan interval 60 secs
6 networks peaked at 16:08:39 Oct 26 2020 UTC (01:28:44.518 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2  4      65001   107    106     7     0     0 01:29:41      6
172.16.255.3  4      65001   102    105     7     0     0 01:29:41      3
172.16.255.4  4      65001   103    105     7     0     0 01:29:44      3

Spine-01#
```

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

```
Spine-01# show bgp l2vpn evpn
BGP table version is 7, local router ID is 172.16.255.1
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: 172.16.254.3:101
* i [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3      0      100      0 ?
*>i      172.16.254.3      0      100      0 ?
* i [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3      0      100      0 ?
*>i      172.16.254.3      0      100      0 ?
Route Distinguisher: 172.16.254.4:101
* i [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
      172.16.254.4      0      100      0 ?
*>i      172.16.254.4      0      100      0 ?
* i [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      172.16.254.4      0      100      0 ?
*>i      172.16.254.4      0      100      0 ?
Route Distinguisher: 172.16.254.3:101
* i [3][172.16.254.3:101][0][32][172.16.254.3]/17
      172.16.254.3      0      100      0 ?
*>i      172.16.254.3      0      100      0 ?
Route Distinguisher: 172.16.254.4:101
* i [3][172.16.254.4:101][0][32][172.16.254.4]/17
      172.16.254.4      0      100      0 ?
*>i      172.16.254.4      0      100      0 ?

Spine-01#
```

Return to [Verifying the Layer 2 VNI with Spine Ingress Replication](#), on page 61.

Outputs to Verify the Configuration on Spine Switch 2

The following example shows the output for the **show ip route** command on Spine Switch 2:

```
Spine-02# show ip route
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

    172.16.0.0/16 is variably subnetted, 12 subnets, 2 masks
O       172.16.13.0/24
         [110/2] via 172.16.23.3, 01:30:51, GigabitEthernet1/0/2
O       172.16.14.0/24
         [110/2] via 172.16.24.4, 01:30:55, GigabitEthernet1/0/3
C       172.16.23.0/24 is directly connected, GigabitEthernet1/0/2
L       172.16.23.2/32 is directly connected, GigabitEthernet1/0/2
C       172.16.24.0/24 is directly connected, GigabitEthernet1/0/3
L       172.16.24.2/32 is directly connected, GigabitEthernet1/0/3
O       172.16.254.3/32
         [110/2] via 172.16.23.3, 01:30:51, GigabitEthernet1/0/2
O       172.16.254.4/32
         [110/2] via 172.16.24.4, 01:30:55, GigabitEthernet1/0/3
O       172.16.255.1/32
         [110/3] via 172.16.24.4, 01:30:51, GigabitEthernet1/0/3
         [110/3] via 172.16.23.3, 01:30:47, GigabitEthernet1/0/2
C       172.16.255.2/32 is directly connected, Loopback0
O       172.16.255.3/32
         [110/2] via 172.16.23.3, 01:30:51, GigabitEthernet1/0/2
O       172.16.255.4/32
         [110/2] via 172.16.24.4, 01:30:55, GigabitEthernet1/0/3

Spine-02#
```

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 7, main routing table version 7
6 network entries using 2064 bytes of memory
12 path entries using 2496 bytes of memory
3/3 BGP path/bestpath attribute entries using 864 bytes of memory
2 BGP rinfo entries using 80 bytes of memory
1 BGP extended community entries using 40 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 5544 total bytes of memory
BGP activity 6/0 prefixes, 12/0 paths, scan interval 60 secs
6 networks peaked at 16:09:46 Oct 26 2020 UTC (01:29:52.664 ago)
```


Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	108	108	7	0	0	01:30:50	6
172.16.255.3	4	65001	105	107	7	0	0	01:30:50	3
172.16.255.4	4	65001	104	106	7	0	0	01:30:46	3

Spine-02#

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

```
Spine-02# show bgp l2vpn evpn
BGP table version is 7, local router ID is 172.16.255.2
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

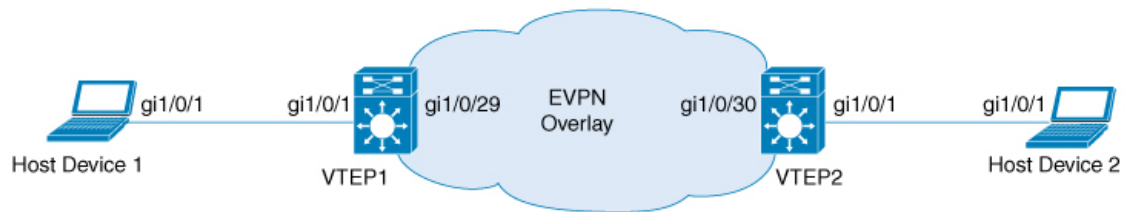
      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 172.16.254.3:101
  * i  [2][172.16.254.3:101][0][48][44D3CA286CC1][0][*]/20
      172.16.254.3          0      100      0 ?
  *>i  172.16.254.3          0      100      0 ?
  * i  [2][172.16.254.3:101][0][48][44D3CA286CC1][32][10.1.101.10]/24
      172.16.254.3          0      100      0 ?
  *>i  172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
  * i  [2][172.16.254.4:101][0][48][44D3CA286CC2][0][*]/20
      172.16.254.4          0      100      0 ?
  *>i  172.16.254.4          0      100      0 ?
  * i  [2][172.16.254.4:101][0][48][44D3CA286CC2][32][10.1.101.20]/24
      172.16.254.4          0      100      0 ?
  *>i  172.16.254.4          0      100      0 ?
Route Distinguisher: 172.16.254.3:101
  * i  [3][172.16.254.3:101][0][32][172.16.254.3]/17
      172.16.254.3          0      100      0 ?
  *>i  172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
  * i  [3][172.16.254.4:101][0][32][172.16.254.4]/17
      172.16.254.4          0      100      0 ?
  *>i  172.16.254.4          0      100      0 ?

Spine-02#
```

Return to [Verifying the Layer 2 VNI with Spine Ingress Replication](#), on page 61.

Example: Configuring BUM Traffic Rate Limiting

This example shows how to configure and verify BUM traffic rate limiting in a BGP EVPN VXLAN fabric using the following topology:



356465

The topology shows an EVPN VXLAN network with 2 VTEPs (VTEP 1 and VTEP 2) connected to perform bridging.

Configuring BUM Traffic Rate Limiting on a VTEP

The following example provides a sample configuration for BUM traffic rate limiting on VTEP 1:

```
Leaf-01# configure terminal
Leaf-01(config)# class-map match-all CL2Miss
Leaf-01(config-cmap)# match 12 dst-mac miss
Leaf-01(config-cmap)# exit
Leaf-01(config)# policy-map PL2Miss
Leaf-01(config-pmap)# class CL2Miss
Leaf-01(config-pmap-c)# police 100000
Leaf-01(config-pmap-c)# exit
Leaf-01(config)# interface nve1
Leaf-01(config-if)# service-policy output PL2Miss
Leaf-01(config-if)# exit
Leaf-01(config)# end
Leaf-01#
```

Verifying BUM Traffic Rate Limiting on a VTEP

The following example shows how to check the aggregated policy map and rate statistics on VTEP 1:

```
Leaf-01# show policy-map int nve1

nve1
  Service-policy output: PL2Miss
  Class-map: sam1 (match-all)
    0 packets
  Match: 12 dst-mac miss
  police:
    cir 100000 bps, bc 3125 bytes
    conformed 221238 bytes; actions:
      transmit
    exceeded 2647233234 bytes; actions:
      drop
    conformed 7000 bps, exceeded 69060000 bps
  Class-map: class-default (match-any)
    10022668 packets
  Match: any

Leaf-01#
```

The following example shows how to validate the member VNI policy under an NVE on VTEP 1:

```
Leaf-01# show platform software fed switch active qos policy target brief | begin PL2Miss
TCG summary for policy: PL2Miss
Loc Interface                IIF-ID                Dir tccg Child #m/p/q  State:(cfg,opr)
-----
L:255 nve1.VNI10000         0x00000000420012     OUT   2    0 0/1/0  VALID,SET_INHW
0x7f605dc9b258
L:255 nve1                  0x000000000000bb     OUT   2    0 0/1/0  VALID,INIT
0x7f605dc9c2f8

Leaf-01#
```

The following example shows how to validate the individual statistics on VTEP 1:

```
Leaf-01# show platform software fed switch active qos policer all_instances trail
All policer instances: With trail
*****
      List of AAL QoS Policer Instances on Targets
AAL Info:
=====
Handle      : 0x4
Target      : 0xdf0001b7(iif_id : 0x420012)
Asic num    : 0x0
Policer Type : Aggregate
le id       : 0x5db76438
le Type     : PORT
Ingress Block: 0x0
Egress Block : 0x25
Policer HW info:
  Ingress:(Total : 0)
    Policer Policer Policer
    Number  Type   offset
    -----
  Egress:(Total : 1)
    Policer Policer Policer
    Number  Type   offset
    -----
           0    1R2C    0
RAL handle  : 4294967295
RAL Info:(Base:Double)
=====
AFD handles : Ingress - Not allocated Egress - 0

AFD QIM Info:
=====
Policer Block Handle : 0
ASIC Num              : 0 (Physical:0, Core 0)
LE ID                 : 278
LE Type               : 1
Policer Base          : 126976
Size                  : 1
Start Index           : 0
End Index              : 0
Ingress Offset        : 1
Ingress Offsets       : 1R2C:0 (Total:0), 1R3C:0 (Total:0), 2R3C:0 (Total:0)
Egress Offsets        : 1R2C:0 (Total:1), 1R3C (Total:0):0, 2R3C:0 (Total:0)

Policer|Policer|Rate                |Exceed Rate                |Burst Size                |Exceed Burst
Size  |Drop or |Exceed Drop|Mark Tbl          |Class |Color|Offset |Type  |(bps) [RegVal]
| (bps) [RegVal] | (Bytes) [RegVal] | (Bytes) [RegVal]
|Markdown|orMarkdown|Exceed/Violate|Default|Aware|
=====
           0|Out1R2C|100057 [0x2f3b]          |29 [0x0000]                |3136 [49]                  |0 [0]
```

Example: Configuring BUM Traffic Rate Limiting

```

|DROP |N/A |0x0/0x0 |No |No |
-----|-----|-----|-----|-----|
Policer|DMA Stats In (Bytes) |DMA Stats Out (Bytes) |DMA Stats In (Frames)
|DMA Stats Out (Frames) |Offset |Green/Yellow |DMA Stats In (Frames)
|Green/Yellow |Green/Yellow/Red |Green/Yellow/Red
-----|-----|-----|-----|-----|
0| 2647454472/ | 0| 221238/ 2647233234/ | 0| 25955436/
0| 2169/ 25953267/ | 0|
-----|-----|-----|-----|-----|
***** END *****

Leaf-01#

```



CHAPTER 3

Configuring EVPN VXLAN Layer 3 Overlay Network

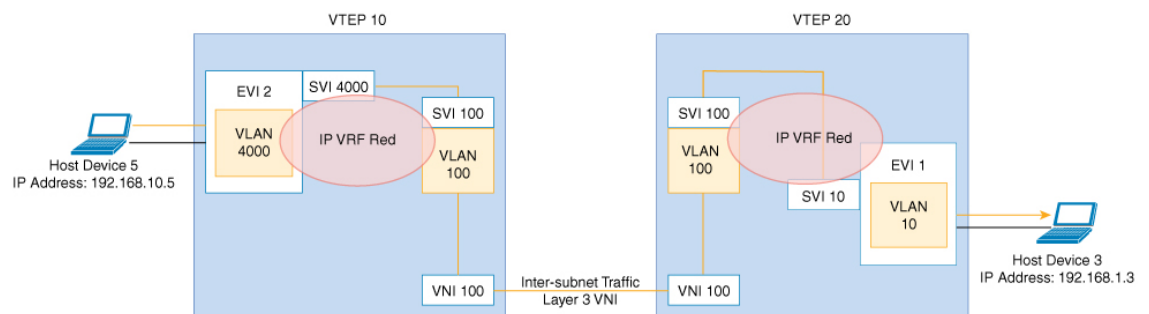
- [Information About EVPN VXLAN Layer 3 Overlay Network](#), on page 73
- [How to Configure EVPN VXLAN Layer 3 Overlay Network](#), on page 74
- [Configuration Examples for EVPN VXLAN Layer 3 Overlay Network](#), on page 84
- [Verifying EVPN VXLAN Layer 3 Overlay Network](#), on page 91

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.

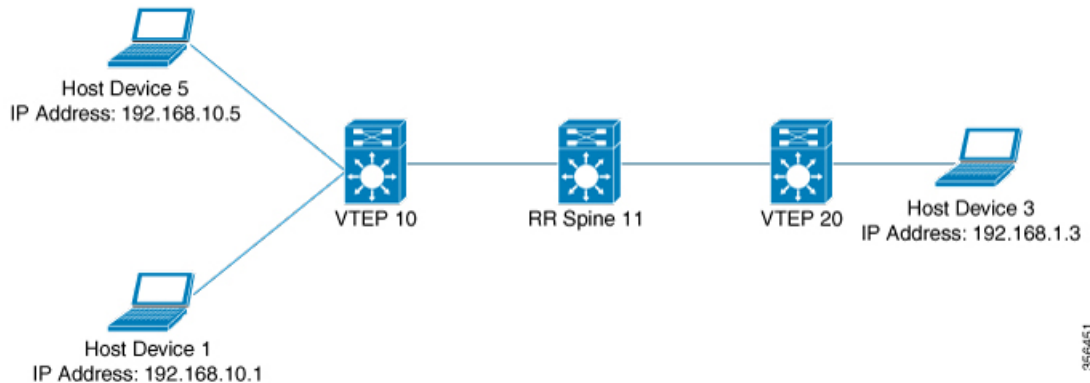
This module provides information only about how to configure a Layer 3 overlay network. You can also configure both Layer 2 and Layer 3 overlay networks together to enable integrated routing and bridging (IRB). For more information about IRB, see *Configuring EVPN VXLAN Integrated Routing and Bridging* 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
	Example: Device> enable	Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Device (config)# vrf definition Green	Enters the VRF configuration mode for the specified VRF instance.
Step 4	rd <i>vpn-route-distinguisher</i> Example: Device (config-vrf)# rd 100:1	Specifies the route distinguisher for the VRF instance.
Step 5	address-family ipv4 [multicast unicast] Example: Device (config-vrf)# address-family ipv4	Enters the IPv4 address family configuration mode.
Step 6	route-target { export import both } <i>route-target-ext-community</i> Example: Device (config-vrf-af)# route-target export 100:1 Example: Device (config-vrf-af)# route-target import 100:1	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).
Step 7	route-target { export import both } <i>route-target-ext-community</i> stitching Example: Device (config-vrf-af)# route-target export 100:1 stitching Example: Device (config-vrf-af)# route-target import 100:1 stitching	Configures importing, exporting, or both importing and exporting of EVPN route target communities for the VRF.
Step 8	exit-address-family Example: Device (config-vrf-af)# exit-address-family	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 9	address-family ipv6 [multicast unicast] Example: Device (config-vrf)# address-family ipv6	Enters the IPv6 address family configuration mode.

	Command or Action	Purpose
Step 10	route-target {export import both} <i>route-target-ext-community</i> Example: Device (config-vrf-af) # route-target export 100:1 Example: Device (config-vrf-af) # route-target import 100:1	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).
Step 11	route-target {export import both} <i>route-target-ext-community stitching</i> Example: Device (config-vrf-af) # route-target export 100:1 stitching Example: Device (config-vrf-af) # route-target import 100:1 stitching	Configures importing, exporting, or both importing and exporting of VXLAN route target communities for the VRF.
Step 12	exit-address-family Example: Device (config-vrf-af) # exit-address-family	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 13	end Example: Device (config-vrf) # end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vlan configuration <i>vlan-id</i> Example:	Enters VLAN feature configuration mode for the specified VLAN interface.

	Command or Action	Purpose
	Device(config)# vlan configuration 11	
Step 4	member vni l3-vni-number Example: Device(config-vlan)# member vni 5000	Adds EVPN instance as a member of the VLAN configuration. The VNI here is used as a Layer 3 VNI.
Step 5	end Example: Device(config-vlan)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-name Example: Device(config)# interface GigabitEthernet1/0/1	Enters interface configuration mode for the specified interface.
Step 4	switchport access vlan vlan-id Example: Device(config-if)# switchport access vlan 40	Configures the interface as a static-access port of the specified VLAN. Interface can also be configured as a trunk interface, if required.
Step 5	end Example: Device(config-if)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>vlan-id</i> Example: Device(config)# interface vlan 11	Enters interface configuration mode for the specified VLAN.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device(config-if)# vrf forwarding Green	Configures the SVI for the VLAN.
Step 5	ip unnumbered <i>Loopback-interface</i> Example: Device(config-if)# ip unnumbered Loopback0	Enables IP processing on the Loopback interface without assigning an explicit IP address to the interface.
Step 6	no autostate Example: Device(config-if)# no autostate	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 no autostate command must be configured under the SVI.
Step 7	end Example: Device(config-if)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>vlan-id</i> Example: Device(config)# interface vlan 40	Enters interface configuration mode for the specified VLAN.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device(config-if)# vrf forwarding Green	Configures the SVI for the VLAN.
Step 5	ip address <i>ip-address</i> Example: Device(config-if)# ip address 192.168.10.100 255.255.255.0	Configures the IP address of the SVI.
Step 6	mac-address <i>mac-address-value</i> Example: Device(config-if)# mac-address aabb.cc01.f100	(Optional) Manually sets the MAC address for the VLAN interface.
Step 7	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 8	end Example: Device(config-if)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>loopback-interface-id</i> Example: Device(config)# interface Loopback0	Enters interface configuration mode for the specified Loopback interface.
Step 4	ip address <i>ipv4-address</i> Example: Device(config-if)# ip address 10.12.11.11 255.255.255.255	Configures the IP address for the Loopback interface.
Step 5	ip pim sparse mode Example: Device(config-if)# ip pim sparse mode	(Optional) Enables Protocol Independent Multicast (PIM) sparse mode on the Loopback interface. Note 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.
Step 6	end Example: Device(config-vlan)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>nve-interface-id</i> Example: Device(config)# interface nve1	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	no ip address Example:	Disables IP processing on the interface by removing its IP address.

	Command or Action	Purpose
	<code>Device(config-if)# no ip address</code>	
Step 5	source-interface <i>loopback-interface-id</i> Example: <code>Device(config-if)# source-interface loopback0</code>	Sets the IP address of the specified loopback interface as the source IP address.
Step 6	host-reachability protocol bgp Example: <code>Device(config-if)# host-reachability protocol bgp</code>	Configures BGP as the host-reachability protocol on the interface. Note You must configure the host reachability protocol on the interface. If you do not execute this step, the VXLAN tunnel defaults to static VXLAN tunnel, which is currently not supported on the Cisco Catalyst 9000 Series switches.
Step 7	member vni <i>vni-id</i> vrf <i>vrf-name</i> Example: <code>Device(config-if)# member vni 5000 vrf Green</code>	Associates the Layer 3 VNI id with the NVE interface. Note The Layer 3 VNI id must match with the VNI id configured in the core VLAN on the VTEP.
Step 8	end Example: <code>Device(config-if)# end</code>	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
Step 1	enable Example: <code>Device> enable</code>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: <code>Device# configure terminal</code>	Enters global configuration mode.

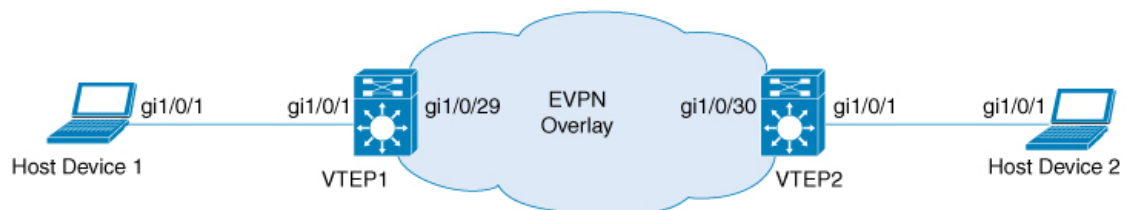
	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device (config) # router bgp 1	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
Step 4	bgp log-neighbor-changes Example: Device (config-router) # bgp log-neighbor-changes	(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	bgp update-delay <i>time-period</i> Example: Device (config-router) # bgp update-delay 1	(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	bgp graceful-restart Example: Device (config-router) # bgp graceful-restart	(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	no bgp default ipv4-unicast Example: Device (config-router) # no bgp default ipv4-unicast	(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	neighbor ip-address remote-as <i>number</i> Example: Device (config-router) # neighbor 10.11.11.11 remote-as 1	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	neighbor {ip-address group-name} update-source <i>interface</i> Example: Device (config-router) # neighbor 10.11.11.11 update-source 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.
Step 10	address-family l2vpn evpn Example: Device (config-router) # address-family l2vpn evpn	Specifies the L2VPN address family and enters address family configuration mode.
Step 11	neighbor ip-address activate Example: Device (config-router-af) # neighbor 10.11.11.11 activate	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.

	Command or Action	Purpose
Step 12	neighbor <i>ip-address</i> send-community [both extended standard] Example: Device(config-router-af)# neighbor 10.11.11.11 send-community both	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 13	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 14	address-family ipv4 vrf <i>vrf-name</i> Example: Device(config-router)# address-family ipv4 vrf Green	Specifies the IPv4 address family and enters address family configuration mode.
Step 15	advertise l2vpn evpn Example: Device(config-router-af)# advertise l2vpn evpn	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.
Step 16	redistribute connected Example: Device(config-router-af)# redistribute connected	(Optional) Redistributes connected routes to BGP.
Step 17	redistribute static Example: Device(config-router-af)# redistribute static	(Optional) Redistributes static routes to BGP.
Step 18	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 19	address-family ipv6 vrf <i>vrf-name</i> Example: Device(config-router)# address-family ipv6 vrf green	Specifies the IPv6 address family and enters address family configuration mode.
Step 20	advertise l2vpn evpn Example: Device(config-router-af)# advertise l2vpn evpn	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.

	Command or Action	Purpose
Step 21	redistribute connected Example: Device(config-router-af) # redistribute connected	(Optional) Redistributes connected routes to BGP.
Step 22	redistribute static Example: Device(config-router-af) # redistribute static	(Optional) Redistributes static routes to BGP.
Step 23	exit-address-family Example: Device(config-router-af) # exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 24	end Example: Device(config-router) # end	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 8: 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 88](#)
- [show bgp l2vpn evpn all, on page 88](#)
- [show ip route vrf, on page 89](#)
- [show platform software fed switch active matm mactable vlan, on page 90](#)

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   Vlan200
200   380e.4d9b.6a4a                         0x1000001     0   0       64 0x7f5d85117598      0x7f5d85110f78
      0x7f5d851b9648                       0x0           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                                0          0  vlan200

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 9: Commands to Verify EVPN VXLAN Layer 3 Overlay Network

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

Command	Purpose
show bgp l2vpn evpn all	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 93](#)
- [Information About EVPN VXLAN Integrated Routing and Bridging, on page 93](#)
- [How to Configure EVPN VXLAN Integrated Routing and Bridging, on page 99](#)
- [Verifying EVPN VXLAN Integrated Routing and Bridging, on page 108](#)
- [Configuration Examples for EVPN VXLAN Integrated Routing and Bridging, on page 108](#)

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. To achieve physical redundancy, configure Cisco Stackwise Virtual on the leaf switches. For more information, see *Configuring Cisco StackWise Virtual* module of the *High Availability Configuration Guide*.
- HSRP and VRRP are not supported for the EVPN VXLAN Layer 2 overlay networks when you use centralized gateway.
- Reorigination of route type 2 host routes to route type 5 routes is only supported starting from Cisco IOS XE Amsterdam 17.3.2a release.

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.

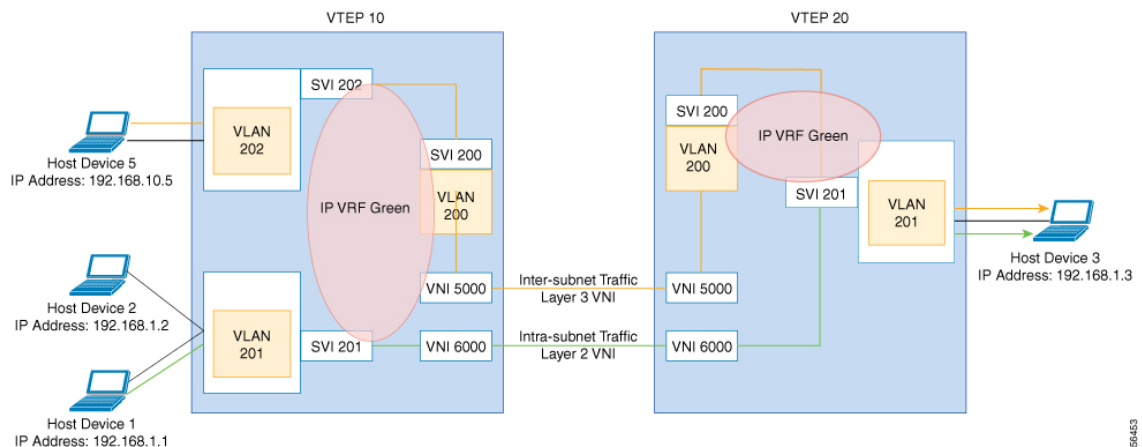
The return traffic during asymmetric IRB goes through a different virtual network instance (VNI) compared to the source traffic. Asymmetric IRB needs the source and destination VNIs to be associated with both the ingress and egress VTEPs.

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:



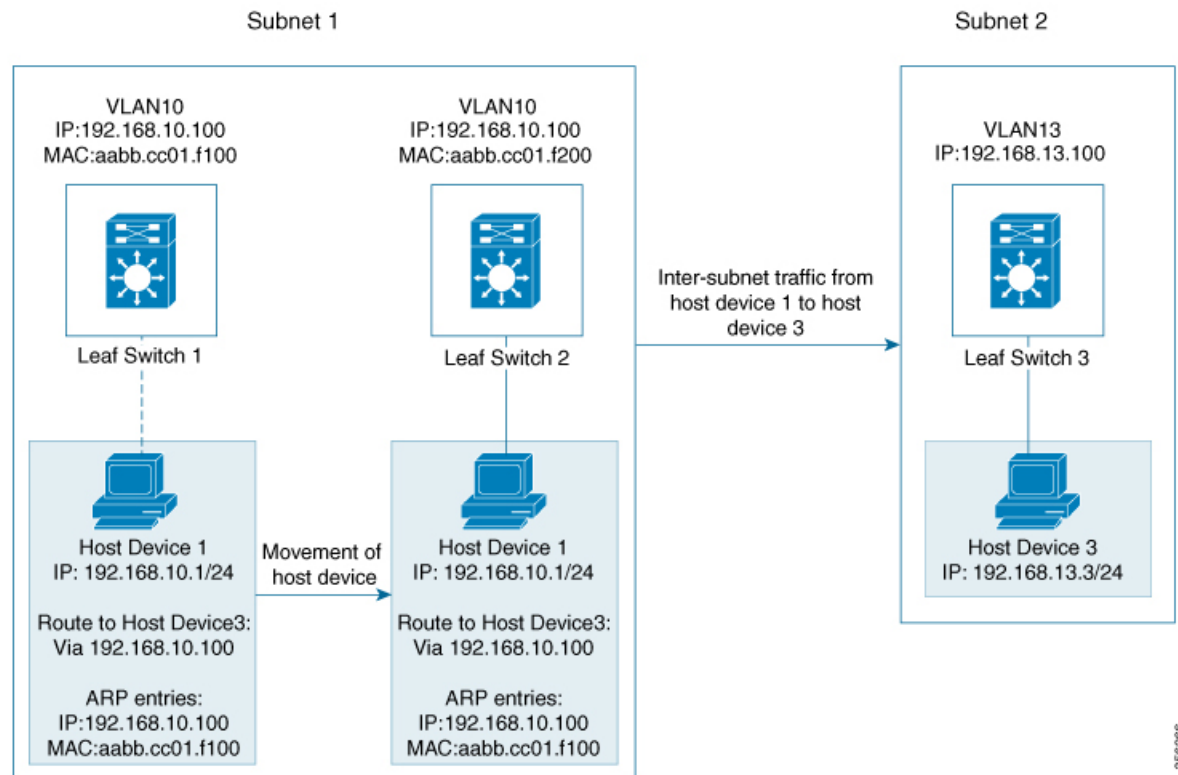
EVPN VXLAN Distributed Anycast Gateway

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 5: Distributed Gateway Topology



When multiple VETPs 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 5: Distributed Gateway Topology, on page 95](#) 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 5: Distributed Gateway Topology, on page 95](#) 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.



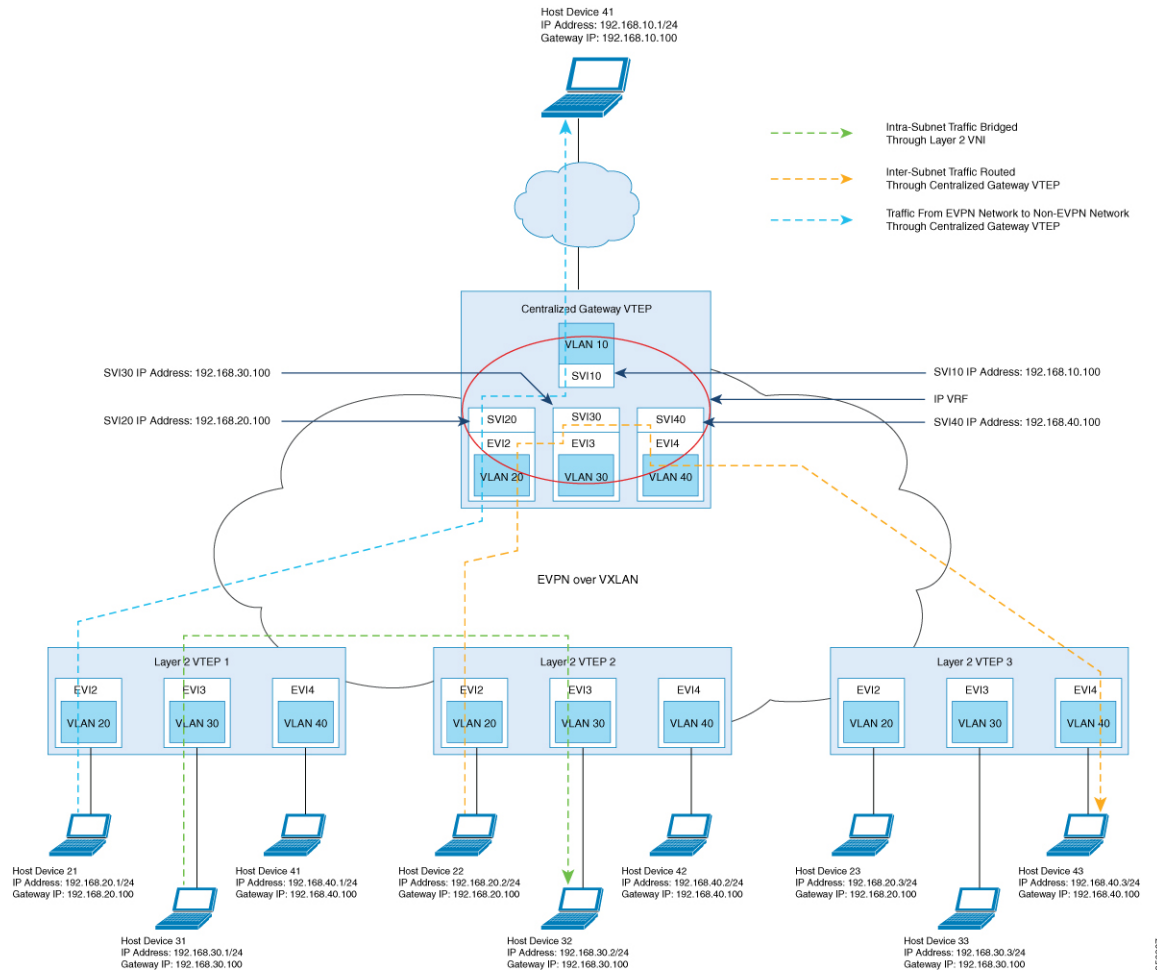
- Note** We recommend that you configure a centralized default gateway in an EVPN VXLAN network if:
- You require a boundary between Layer 2 and Layer 3 segments at the border of the BGP EVPN VXLAN fabric.
 - The inter-subnet traffic is subjected to a firewall inspection or any policy on a centralized 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 6: 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.

Route Type 2 to Route Type 5 Host Route Reorigination for Overlay Networks with a Centralized Gateway (Asymmetric IRB)

Route type 2 (RT 2) to Route type 5 (RT 5) reorigination allows a Layer 2 and Layer 3 VXLAN network to import IP addresses from another Layer 2-only VXLAN network. In this deployment model, a VXLAN gateway VTEP functions as both the CGW and the Layer 3 VXLAN gateway. You can use the gateway VTEP to connect a Layer 2-only VXLAN network with a Layer 2 and Layer 3 VXLAN network. The gateway VTEP uses the CGW and asymmetric IRB forwarding functionalities in the EVPN segment of the Layer 2-only network. The VTEP can reoriginate any RT 2 host routes, that it learns from the Layer 2 EVPN segment (or MAC VRF), as RT 5 routes in the Layer 3 EVPN segment (or IP VRF). The VTEP then forwards the reoriginated RT 5 routes to host devices in the Layer 2 and Layer 3 VXLAN network.



Note The Layer 3 EVPN segment can be one of the following:

- An IP VRF-only segment (see *Configuring EVPN VXLAN Layer 3 Overlay Network* module for more information)
- Distributed anycast gateway (DAG) overlay fabric

To enable RT 2 to RT 5 host route reorigination on the CGW VTEP, ensure that you run the **reoriginate route-type5** and **member vni layer2-vni-id ingress-replication local-routing** commands on that VTEP. For detailed steps, see [Configuring Layer 2 VPN EVPN on a VTEP, on page 99](#) and [Configuring the NVE Interface on a VTEP, on page 102](#) sections.



Note The CGW VTEP does not install the reoriginated RT 5 route on its local IP VRF.

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 15](#) for detailed steps.

Configuring IP VRF on VTEP

See [Configuring an IP VRF on a VTEP, on page 74](#) for detailed steps.

Configuring the Loopback Interface on a VTEP

See [Configuring the Loopback Interface on a VTEP, on page 79](#) 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
Step 1	enable Example:	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vlan configuration <i>vlan-id</i> Example: Device(config)# vlan configuration 201	Enters VLAN feature configuration mode for the specified VLAN interface.
Step 4	member evpn-instance <i>evpn-instance-id</i> vni <i>l2-vni-number</i> Example: Device(config-vlan)# member evpn-instance 1 vni 6000	Adds EVPN instance as a member of the VLAN configuration. The VNI here is used as a Layer 2 VNI.
Step 5	exit Example: Device(config-vlan)# exit	Returns to global configuration mode.
Step 6	vlan configuration <i>vlan-id</i> Example: Device(config)# vlan configuration 202	Enters VLAN feature configuration mode for the specified VLAN interface.
Step 7	member evpn-instance <i>evpn-instance-id</i> vni <i>l2-vni-number</i> Example: Device(config-vlan)# member evpn-instance 2 vni 7000	Adds EVPN instance as a member of the VLAN configuration. The VNI here is used as a Layer 2 VNI.
Step 8	exit Example: Device(config-vlan)# exit	Returns to global configuration mode.
Step 9	vlan configuration <i>vlan-id</i> Example: Device(config)# vlan configuration 200	Enters VLAN feature configuration mode for the specified VLAN interface.
Step 10	member vni <i>l3-vni-number</i> Example: Device(config-vlan)# member vni 5000	Adds EVPN instance as a member of the VLAN configuration. The VNI here is used as a Layer 3 VNI.
Step 11	exit Example: Device(config-vlan)# exit	Returns to global configuration mode.

	Command or Action	Purpose
Step 12	end Example: Device(config-vlan)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>vlan-id</i> Example: Device(config)# interface vlan 200	Enters interface configuration mode for the specified VLAN.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device(config-if)# vrf forwarding Green	Configures the SVI for the VLAN.
Step 5	ip unnumbered <i>Loopback-interface</i> Example: Device(config-if)# ip unnumbered Loopback0	Enables IP processing on the Loopback interface without assigning an explicit IP address to the interface.
Step 6	no autostate Example: Device(config-if)# no autostate	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 no autostate command must be configured under the SVI.
Step 7	end Example: Device(config-if)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>vlan-id</i> Example: Device(config)# interface vlan 202	Enters interface configuration mode for the specified VLAN.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device(config-if)# vrf forwarding Green	Configures the SVI for the VLAN.
Step 5	ip address <i>gateway-ip-address</i> Example: Device(config-if)# ip address 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.
Step 6	mac-address <i>mac-address-value</i> Example: Device(config-if)# mac-address aabb.cc01.f100	(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.
Step 7	end Example: Device(config-if)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>nve-interface-id</i> Example: Device(config)# interface nve1	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	no ip address Example: Device(config-if)# no ip address	Disables IP processing on the interface by removing its IP address.
Step 5	source-interface <i>loopback-interface-id</i> Example: Device(config-if)# source-interface loopback0	Sets the IP address of the specified loopback interface as the source IP address.
Step 6	host-reachability protocol bgp Example: Device(config-if)# host-reachability protocol bgp	Configures BGP as the host-reachability protocol on the interface. Note You must configure the host reachability protocol on the interface. If you do not execute this step, the VXLAN tunnel defaults to static VXLAN tunnel, which is currently not supported on the Cisco Catalyst 9000 Series switches.
Step 7	member vni <i>layer2-vni-id</i> { ingress-replication [local-routing] mcast-group <i>mcast-group-address</i> } Example: Device(config-if)# member vni 6000 mcast-group 227.0.0.1 Device(config-if)# member vni 7000 mcast-group 227.0.0.1	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 mcast-group keyword for static replication and ingress-replication keyword for ingress replication. Use the local-routing keyword only when you need to configure route type 2 (RT 2) to route type 5 (RT 5) reorigination on the centralized gateway (CGW) VTEP.

	Command or Action	Purpose
Step 8	member vni <i>layer3-vni-id</i> vrf <i>vrf-name</i> Example: Device(config-if) # member vni 5000 vrf Green	Associates the Layer 3 VNI member with the NVE.
Step 9	end Example: Device(config-if) # end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 1	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
Step 4	bgp log-neighbor-changes Example: Device(config-router) # bgp log-neighbor-changes	(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	bgp update-delay <i>time-period</i> Example: Device(config-router) # bgp update-delay 1	(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	bgp graceful-restart Example: Device(config-router) # bgp graceful-restart	(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> .

	Command or Action	Purpose
Step 7	no bgp default ipv4-unicast Example: <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	neighbor ip-address remote-as number Example: <pre>Device(config-router)# neighbor 10.11.11.11 remote-as 1</pre>	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	neighbor {ip-address group-name} update-source interface Example: <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	address-family l2vpn evpn Example: <pre>Device(config-router)# address-family l2vpn evpn</pre>	Specifies the L2VPN address family and enters address family configuration mode.
Step 11	neighbor ip-address activate Example: <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	neighbor ip-address send-community [both extended standard] Example: <pre>Device(config-router-af)# neighbor 10.11.11.11 send-community both</pre>	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 13	exit-address-family Example: <pre>Device(config-router-af)# exit-address-family</pre>	Exits address family configuration mode and returns to router configuration mode.
Step 14	address-family ipv4 vrf vrf-name Example: <pre>Device(config-router)# address-family ipv4 vrf green</pre>	Specifies the IPv4 address family and enters address family configuration mode.
Step 15	advertise l2vpn evpn Example:	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.

	Command or Action	Purpose
	Device (config-router-af) # advertise l2vpn evpn	
Step 16	redistribute connected Example: Device (config-router-af) # redistribute connected	Redistributes connected routes to BGP.
Step 17	redistribute static Example: Device (config-router-af) # redistribute static	Redistributes static routes to BGP.
Step 18	exit-address-family Example: Device (config-router-af) # exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 19	address-family ipv6 vrf <i>vrf-name</i> Example: Device (config-router) # address-family ipv6 vrf green	Specifies the IPv6 address family and enters address family configuration mode.
Step 20	advertise l2vpn evpn Example: Device (config-router-af) # advertise l2vpn evpn	Advertises Layer 2 VPN EVPN routes within a tenant VRF in an EVPN VXLAN fabric.
Step 21	redistribute connected Example: Device (config-router-af) # redistribute connected	Redistributes connected routes to BGP.
Step 22	redistribute static Example: Device (config-router-af) # redistribute static	Redistributes static routes to BGP.
Step 23	exit-address-family Example: Device (config-router-af) # exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 24	end Example: Device (config-router) # end	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 99](#)
 - [Configuring the Loopback Interface on a VTEP, on page 99](#)
 - [Configuring Core-facing and Access-facing VLANs on a VTEP, on page 99](#)
 - [Configuring Switch Virtual Interface for the Core-facing VLAN on a VTEP, on page 101](#)
 - [Configuring Switch Virtual Interface for the Access-facing VLANs on a VTEP, on page 102](#)
 - [Configuring the NVE Interface on a VTEP, on page 102](#)
 - [Configuring BGP with EVPN and VRF Address Families on a VTEP, on page 104](#)



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. To achieve physical redundancy, configure Cisco Stackwise Virtual on the leaf switches. For more information, see *Configuring Cisco StackWise Virtual* module of the *High Availability Configuration Guide*.

- 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.

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 23](#) for the list of commands.

Verifying EVPN VXLAN Layer 3 Overlay Network

See [Verifying EVPN VXLAN Layer 3 Overlay Network, on page 91](#) 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 10: Commands to Verify Distributed Anycast Gateway

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

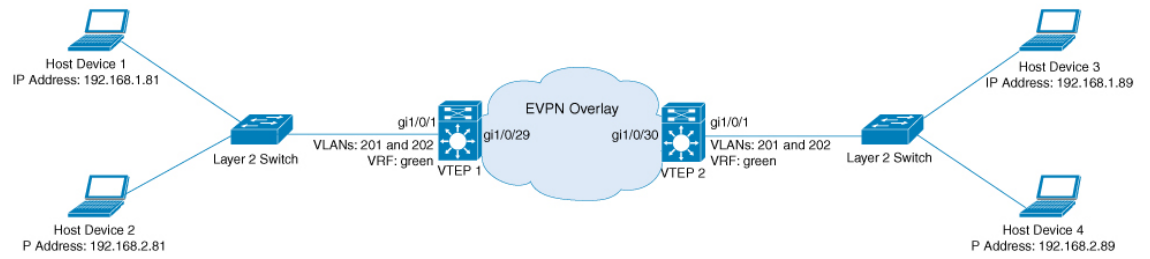
Configuration Examples for EVPN VXLAN Integrated Routing and Bridging

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 7: EVPN VXLAN Topology for IRB using Distributed Anycast Gateway



396597



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 11: 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
--------	--------

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>

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

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 113](#)
- [show l2vpn evpn peers vxlan, on page 114](#)
- [show l2vpn evpn evi evpn-instance detail, on page 114](#)
- [show l2vpn evpn default-gateway, on page 115](#)
- [show bgp l2vpn evpn all, on page 116](#)
- [show ip route vrf green, on page 119](#)
- [show platform software fed switch active matm mactable vlan, on page 120](#)

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.bccc2 V1202
Y  192.168.2.202                2    202    380e.4d9b.6a42 10.2.2.20
Y  2001:DB8:202::202            2    202    a0f8.4910.bccc2 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:

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

```
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.bcc2 10.1.1.10
Y 2001:DB8:202::202            2     202    380e.4d9b.6a42 V1202
Y 2001:DB8:202::202            2     202    a0f8.4910.bcc2 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

      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
```



```

::
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,

```

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

```

        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 ?

```

```

*>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

```

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

```

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

```

```

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  Vlan200
200   a0f8.4910.bce2                       0x1000001  0    0      64  0x7fa8859a3d38  0x7fa885947ba8
      0x7fa88598bfb8                       0x0          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  Vlan201
201   0059.dc50.ae01                       0x1000001  0    0      64  0x7fa8849e1be8  0x7fa88598da48
      0x7fa88598e1f8                       0x0          0        0  RLOC 10.1.1.10 adj_id 153
201   0059.dc50.ae4c                       0x1000001  0    0      64  0x7fa885993e68  0x7fa88598da48
      0x7fa88598e1f8                       0x0          0        0  RLOC 10.1.1.10 adj_id 153
201   a0f8.4910.bccc                       0x8002  0    0      64  0x7fa8859acc48  0x7fa885574e38
      0x0                                  0x5174          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  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        7  RLOC 10.1.1.10 adj_id 154
```

```

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

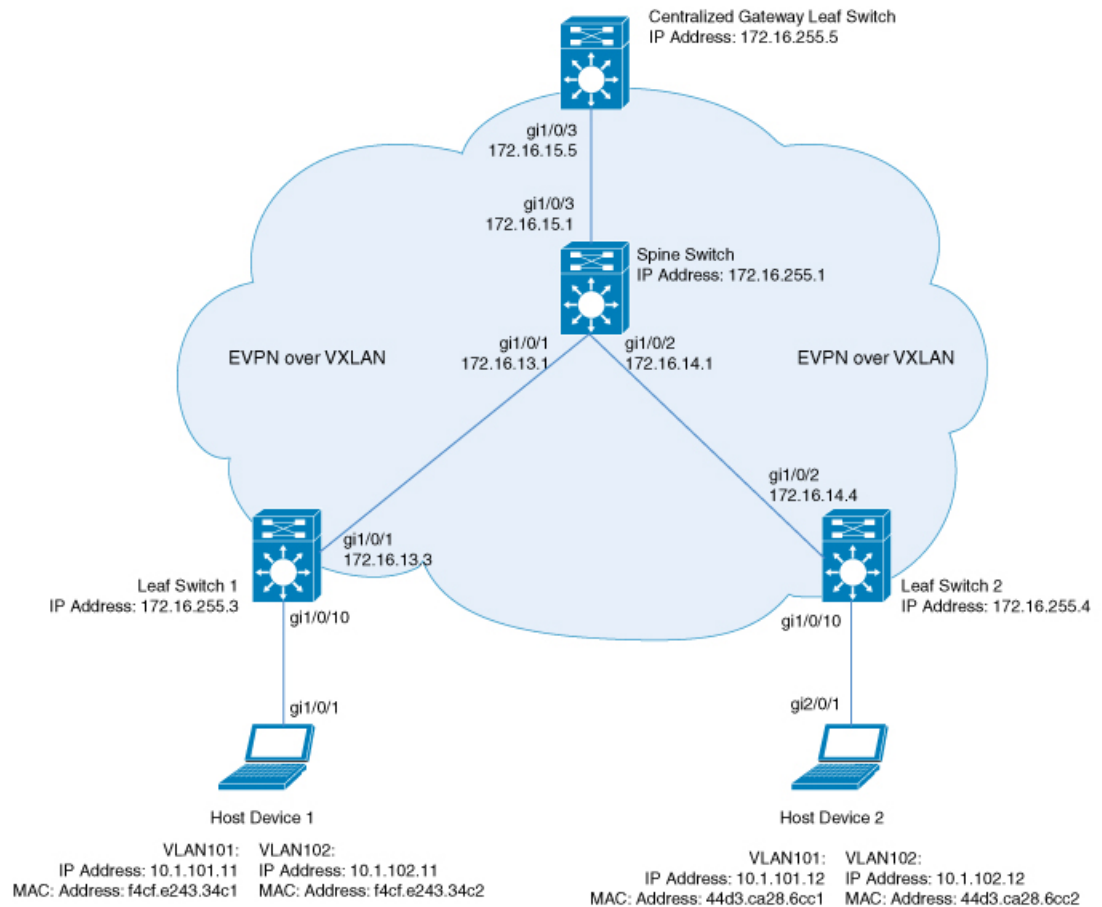
202    0018.736c.56c2    0x1         67    0    0    0x7fa8859d3488    0x7fa8859834f8
      0x0          0x7fa884f079d8    300      267    GigabitEthernet1/0/1

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 8: 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. To achieve physical redundancy, configure Cisco Stackwise Virtual on the leaf switches. For more information, see *Configuring Cisco StackWise Virtual* module of the *High Availability Configuration Guide*.

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

Table 12: 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>

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

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

en#S
lc#


```

enps
lcrv
sca
507
pi
mip
pi
fxo
1
aaa
0
!
en
on
pi
sca
107
pi
mip
pi
fxo
kcn
pi
fxo
1
aaa
0
!
en
on

```

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

```

en1$
lc16
  pi
  sc1
  137
  pi
  mip
  pi
  ipso
  hom
  pi
  ipso
  1
  aca
  0
  !
  eeh
  on
  pi
  sc1
  137
  pi
  mip
  pi
  ipso
  hom
  pi
  ipso
  1
  
```



```
enps  
lcw
```


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

```

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
      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

```

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

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()Dg1
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()Dg1
Next Hop(s):          V1102:0

```

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



CHAPTER 5

Configuring Spine Switches in a BGP EVPN VXLAN Fabric

- [Information About Spine Switches in a BGP EVPN VXLAN Fabric, on page 141](#)
- [Configuration Examples for Spine Switches in a BGP EVPN VXLAN Network, on page 142](#)

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 142](#) 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 159](#) 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 178](#) for a sample topology and configuration.

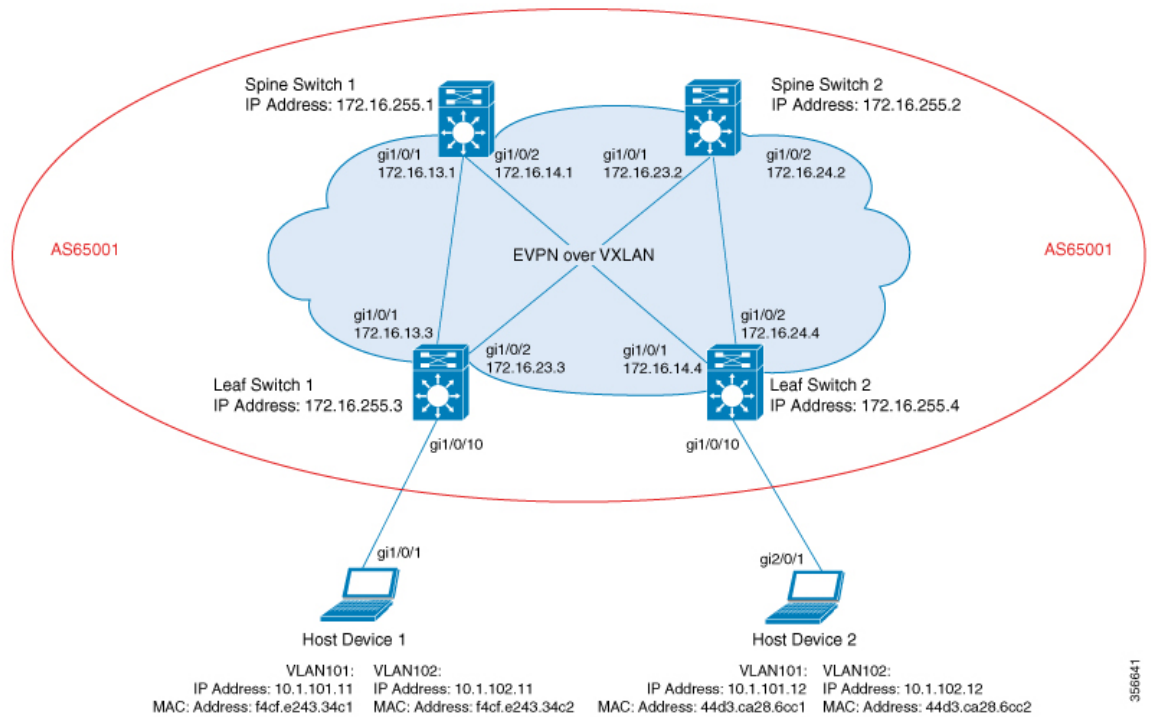
Configuration Examples for Spine Switches in a BGP EVPN VXLAN Network

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 9: 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 14: 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 15: 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 rrinfo 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

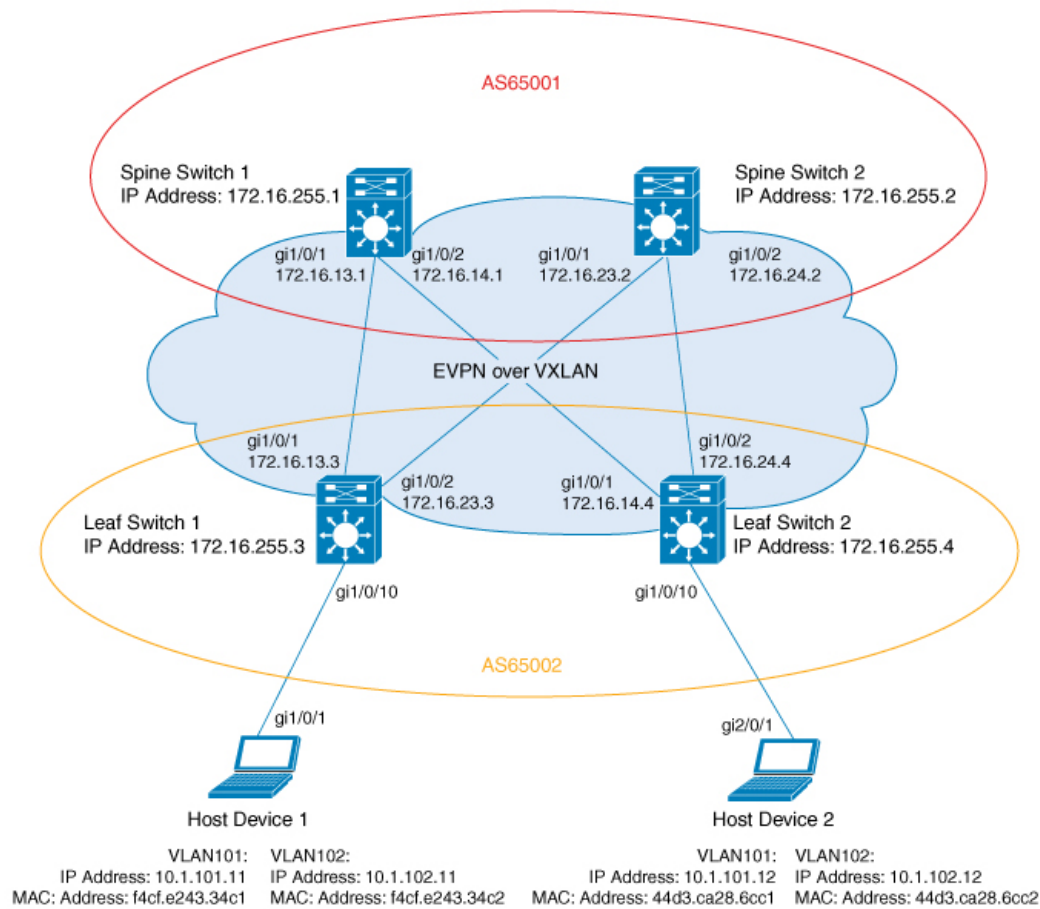
(172.16.254.4, 225.0.0.101), 00:00:09/00:02:50, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:00:09/00:03:20

(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 10: 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 allow-as-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 16: 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 17: 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
                  AS      State      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
                Down   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                                     Prio/Mode
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** 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

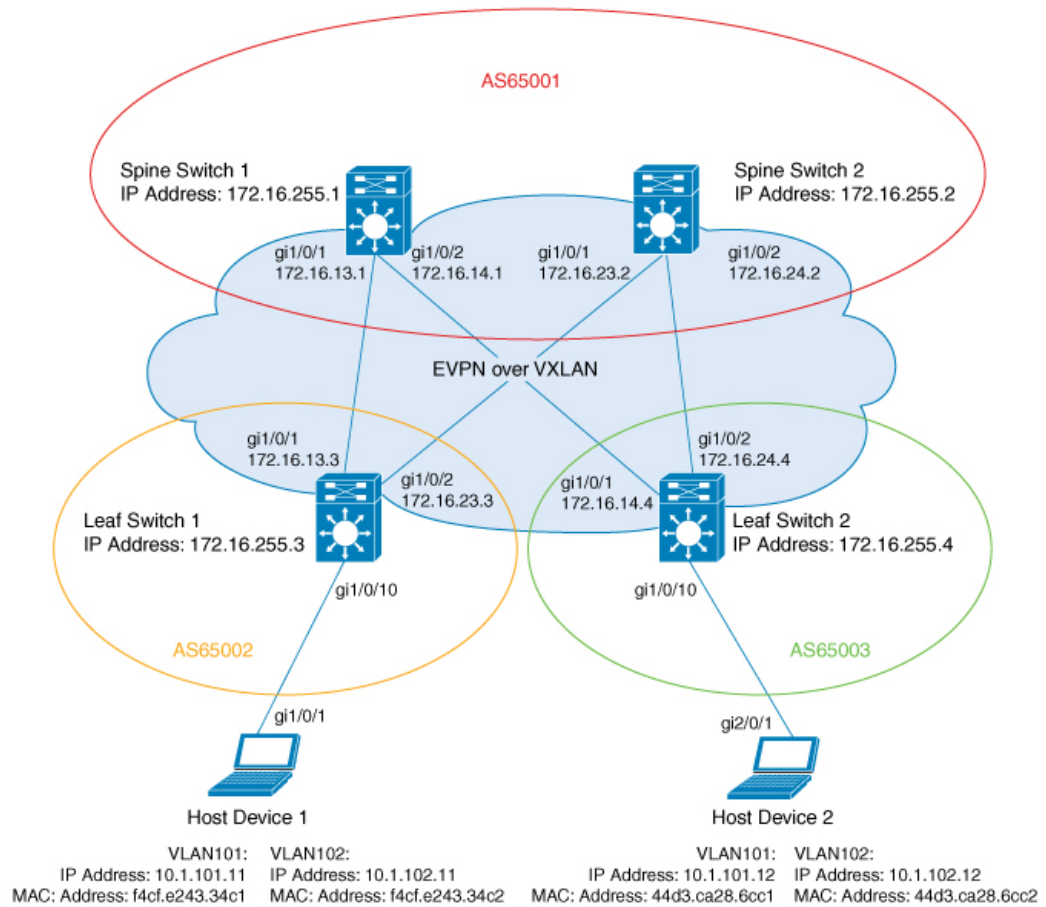
(172.16.254.3, 225.0.0.101), 00:02:09/00:00:50, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:02:09/00:00:50

```

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 11: 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 18: 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 19: 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
                AS      State    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 199](#)
- [Information About DHCP Relay in a BGP EVPN VXLAN Fabric, on page 199](#)
- [How to Configure DHCP Relay in a BGP EVPN VXLAN Fabric, on page 201](#)
- [Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric, on page 205](#)
- [Additional References for DHCP Relay in a BGP EVPN VXLAN Fabric, on page 223](#)

Restrictions for DHCP Relay in a BGP EVPN VXLAN Fabric

DHCPv6 prefix delegation 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 server is in the Layer 3 default VRF and DHCP client is in the tenant VRF.

See [Example: DHCP Server is in the Layer 3 Default VRF and the DHCP Client is in the Tenant VRF, on page 206](#) for a configuration example.

2. DHCP server and DHCP client are in the same tenant VRF.

See [Example: DHCP Server and DHCP Client are in the Same Tenant VRF, on page 211](#) for a configuration example.

3. DHCP server and DHCP client are in different tenant VRFs.

See [Example: DHCP Client and DHCP Server are in Different Tenant VRFs, on page 214](#) for a configuration example.

4. DHCP server is in a non-default non-VXLAN VRF and DHCP client is in the tenant VRF.

See [Example: DHCP Server is in a non-Default, non-VXLAN VRF and DHCP Client is in the Tenant VRF, on page 219](#) for a configuration example.

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 99](#) 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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip dhcp relay information option vpn Example: Device(config)# ip dhcp relay information option vpn	Adds option VPN suboption to DHCP option 82. 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.
Step 4	ip dhcp relay information option Example: Device(config)# ip dhcp relay information option	Enables DHCP option 82. Enables the system to insert a DHCP relay agent information option in the messages forwarded to the DHCP server.
Step 5	ip dhcp relay override gateway-ip-address link-selection Example: Device(config)# ip dhcp relay override giaddr link-selection	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
Step 6	ip dhcp compatibility suboption {link-selection server-override} standard Example: Device(config)# ip dhcp compatibility suboption link-selection standard 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. Use the link-selection standard keyword to switch to standard DHCP option 82[5]. Use the server-override standard keyword to switch to standard DHCP option 82[11].
Step 7	ip dhcp snooping vlan <i>vlan-id-list</i> Example: Device(config)# ip dhcp snooping vlan 201-202	Enables DHCP snooping on the specified list of VLANs.
Step 8	ip dhcp snooping Example: Device(config)# ip dhcp snooping	Enables DHCP snooping globally on the VTEP.
Step 9	end Example: 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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>vlan-id</i> Example: Device(config)# interface Vlan 201	Enters interface configuration mode for the specified VLAN interface. This VLAN interface acts as the GiAddr.
Step 4	vrf forwarding <i>vrf-name</i>	Associates the VRF with the interface.

	Command or Action	Purpose
	Example: Device(config-if)# <code>vrf forwarding green</code>	The interface must be associated with the same VRF for which the Layer 3 VNI has been configured for the EVPN VXLAN network.
Step 5	ip dhcp relay information option vpn-id Example: Device(config-if)# <code>ip dhcp relay information option vpn-id</code>	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.
Step 6	ip dhcp relay source-interface Loopback loopback-interface-id Example: Device(config-if)# <code>ip dhcp relay source-interface Loopback13</code>	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. Note The IP address configured on the Loopback interface must be unique per VTEP per VRF.
Step 7	ip address ip-address Example: Device(config-if)# <code>ip address 192.168.1.201 255.255.255.0</code>	Sets the IP address for the VLAN interface.
Step 8	ip helper-address [global vrf vrf-name] ip-address Example: Device(config-if)# <code>ip helper-address global 192.168.3.100</code> Device(config-if)# <code>ip helper-address vrf green 192.168.20.20</code>	Sets the DHCP IP helper address for the VLAN interface. Use the global keyword if the DHCP server is reachable over the global routing table (GRT). Use the vrf vrf-name keyword if the DHCP server is reachable over the tenant VRF.
Step 9	exit Example: Device(config-if)# <code>exit</code>	Exits interface configuration mode and returns to global configuration mode.
Step 10	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

Configuring the Layer 3 or Routed Interface on the Border VTEP for DHCP Server Reachability

DHCP server reachability can be achieved through a physical Layer 3 interface (or subinterface), a dot1Q interface, an SVI, or a Layer 3 Portchannel interface (or subinterface).



Note This task is optional if you implement plain IP address forwarding in the respective VRF.

To configure the Layer 3 or routed interface on the border VTEP for external connectivity, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>vlan-id</i> Example: Device (config)# interface vlan 203	Enters interface configuration mode for the specified VLAN interface.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device (config-if)# vrf forwarding green	Configures the SVI for the VLAN and associates the specified VRF with the interface.
Step 5	ip address <i>ip-address</i> Example: Device (config-if)# ip address 192.168.3.203 255.255.255.0	Configures the IP address for the VLAN.
Step 6	ipv6 address <i>ipv6-address</i> Example: Device (config-if)# ipv6 address 2001:203::203/64	Configures the IPv6 address for the VLAN.
Step 7	ipv6 enable Example: Device (config-if)# ipv6 enable	Enables IPv6 processing on the VLAN interface.
Step 8	exit Example: Device (config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 9	interface <i>interface-id</i> Example:	Enters interface configuration mode for the specified interface.

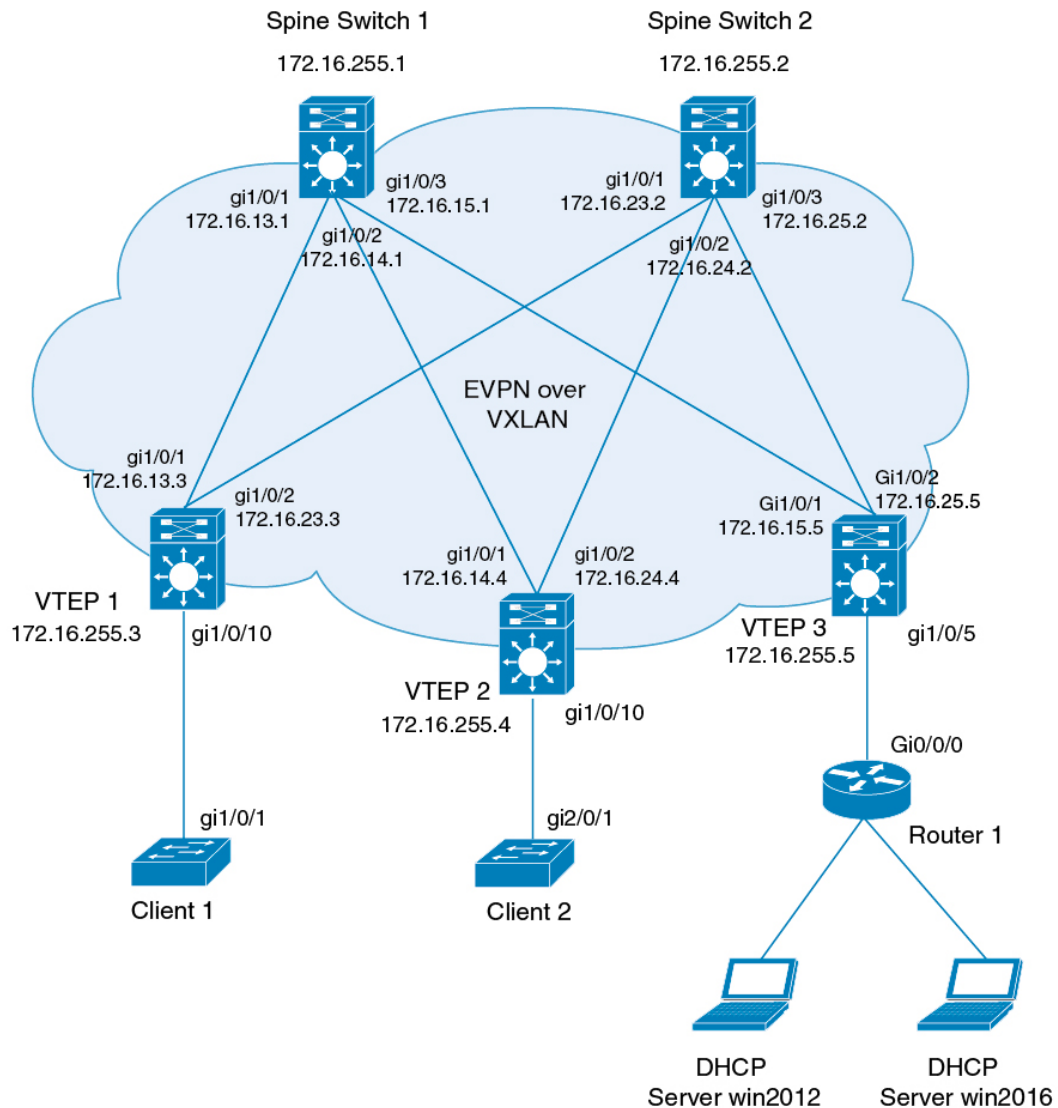
	Command or Action	Purpose
	Device(config)# interface GigabitEthernet1/0/30	
Step 10	switchport access vlan <i>vlan-id</i> Example: Device(config-if)# switchport access vlan 203	Specifies the VLAN to be used as access VLAN when the interface is in access mode.
Step 11	switchport mode access Example: Device(config-if)# switchport mode access	Configures the interface as an access interface.
Step 12	exit Example: Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 13	end Example: Device(config)# end	Returns to privileged EXEC mode.

Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric

This section provides configuration examples for DHCP relay in a BGP EVPN VXLAN fabric for the following scenarios using the topology in [Figure 12: DHCP Relay Deployment in a BGP EVPN VXLAN Fabric](#), on [page 206](#).

- Example: DHCP Server is in the Layer 3 Default VRF and the DHCP Client is in the Tenant VRF
- Example: DHCP Server and DHCP Client are in the Same Tenant VRF
- Example: DHCP Client and DHCP Server are in Different Tenant VRFs
- Example: DHCP Server is in a non-Default, non-VXLAN VRF and DHCP Client is in the Tenant VRF

Figure 12: DHCP Relay Deployment in a BGP EVPN VXLAN Fabric



The preceding figure shows an EVPN VXLAN network with two spine switches (Spine Switch 1 and Spine Switch 2) and three leaf switches (VTEP1, VTEP 2, and VTEP 3). VTEP 3 is connected to two DHCP servers. VTEP 1 and VTEP 2 are connected to a single DHCP client each.

Example: DHCP Server is in the Layer 3 Default VRF and the DHCP Client is in the Tenant VRF

This example shows how to configure DHCP relay deployment in a BGP EVPN VXLAN fabric for the topology in [Figure 12: DHCP Relay Deployment in a BGP EVPN VXLAN Fabric](#) when the DHCP server and the DHCP client are in the same tenant VRF. The DHCP server is reachable over global routing table (GRT).

The following tables provide sample configurations for the DHCP server and VTEP 1:

Table 20: Configuring DHCP when DHCP Server is in the Layer 3 Default VRF and DHCP Client is in the Tenant VRF

DHCP Configuration Snippet
<pre><snip: only the relevant configuration is shown> ip dhcp-relay source-interface Loopback0 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 101-102,201-202 ip dhcp snooping ! interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip ospf 1 area 0 ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip helper-address global 192.168.20.20 ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip helper-address global 192.168.20.20 ! interface Vlan201 vrf forwarding red ip address 10.2.201.1 255.255.255.0 ip helper-address global 192.168.20.20 ! <snip: only the relevant configuration is shown></pre>

Table 21: Configuring VTEP 1 when DHCP Server is in the Layer 3 Default VRF and DHCP Client is in the Tenant VRF

VTEP 1
<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 ! vrf definition red rd 2:2 ! address-family ipv4 route-target export 2:2 route-target import 2:2 route-target export 2:2 stitching route-target import 2:2 stitching exit-address-family ! ip routing ! ip multicast-routing ! ip dhcp-relay source-interface Loopback0 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 101-102,201-202 ip dhcp snooping ! 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 ! l2vpn evpn instance 201 vlan-based encapsulation vxlan ! l2vpn evpn instance 202 vlan-based encapsulation vxlan ! system mtu 9198 ! </pre>

VTEP 1

```
vlan configuration 101
member evpn-instance 101 vni 10101
vlan configuration 102
member evpn-instance 102 vni 10102
vlan configuration 201
member evpn-instance 201 vni 10201
vlan configuration 202
member evpn-instance 202 vni 10202
vlan configuration 901
member vni 50901
vlan configuration 902
member vni 50902
!
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 GigabitEthernet1/0/2
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
ip helper-address global 192.168.20.20
!
interface Vlan102
vrf forwarding green
ip address 10.1.102.1 255.255.255.0
ip helper-address global 192.168.20.20
!
interface Vlan201
vrf forwarding red
ip address 10.2.201.1 255.255.255.0
ip helper-address global 192.168.20.20
!
interface Vlan202
vrf forwarding red
ip address 10.2.202.1 255.255.255.0
ip helper-address global 192.168.20.20
!
```

VTEP 1

```

interface Vlan901
vrf forwarding green
ip unnumbered Loopback0
no autostate
!
interface Vlan902
vrf forwarding red
ip unnumbered Loopback0
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 mcast-group 225.0.0.102
member vni 10201 mcast-group 225.0.0.201
member vni 10202 mcast-group 225.0.0.202
member vni 50901 vrf green
member vni 50902 vrf red
!
router ospf 1
router-id 172.16.255.3
!
router bgp 65001
bgp router-id interface Loopback0
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
!
address-family ipv4 vrf red
advertise l2vpn evpn
redistribute connected
redistribute static
exit-address-family
!
ip pim rp-address 172.16.255.255
!
end

Leaf-01#

```

Return to [Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric, on page 205](#).

Example: DHCP Server and DHCP Client are in the Same Tenant VRF

This example shows how to configure DHCP relay deployment in a BGP EVPN VXLAN fabric for the topology in [Figure 12: DHCP Relay Deployment in a BGP EVPN VXLAN Fabric](#) when the DHCP server and the DHCP client are in the same tenant VRF. The DHCP server is reachable over this common tenant VRF.

The following tables provide sample configurations for the DHCP server and VTEP 1:

Table 22: Configuring DHCP when DHCP Server and DHCP Client are in the Same Tenant VRF

DHCP Configuration Snippet
<pre><snip: only the relevant configuration is shown> 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 101-102,201-202 ip dhcp snooping ! interface Loopback101 vrf forwarding green ip address 10.1.251.1 255.255.255.255 ! interface Vlan101 vrf forwarding green ip dhcp relay source-interface Loopback101 ip address 10.1.101.1 255.255.255.0 ip helper-address 192.168.20.20 ! interface Vlan102 vrf forwarding green ip dhcp relay source-interface Loopback101 ip address 10.1.102.1 255.255.255.0 ip helper-address 192.168.20.20 <snip: only the relevant configuration is shown></pre>

Table 23: Configuring VTEP 1 when DHCP Server and DHCP Client are in the Same Tenant VRF

VTEP 1
<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 ! ip routing ! ip multicast-routing ! 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 101-102,201-202 ip dhcp snooping ! 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 ! l2vpn evpn instance 201 vlan-based encapsulation vxlan ! l2vpn evpn instance 202 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 201 member evpn-instance 201 vni 10201 vlan configuration 202 member evpn-instance 202 vni 10202 vlan configuration 901 member vni 50901 vlan configuration 902 member vni 50902 ! </pre>

VTEP 1

```
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 Loopback101
vrf forwarding green
ip address 10.1.251.1 255.255.255.255
!
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 GigabitEthernet1/0/2
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 dhcp relay source-interface Loopback101
ip address 10.1.101.1 255.255.255.0
ip helper-address 192.168.20.20
!
interface Vlan102
vrf forwarding green
ip dhcp relay source-interface Loopback101
ip address 10.1.102.1 255.255.255.0
ip helper-address 192.168.20.20
!
interface Vlan901
vrf forwarding green
ip unnumbered Loopback0
no autostate
!
```

VTEP 1

```

interface nve1
no ip address
source-interface Loopback1
host-reachability protocol bgp
member vni 10101 mcast-group 225.0.0.101
member vni 10102 mcast-group 225.0.0.102
member vni 50901 vrf green
!
router ospf 1
router-id 172.16.255.3
!
router bgp 65001
bgp router-id interface Loopback0
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
!
ip pim rp-address 172.16.255.255
!
end
!
Leaf-01#

```

Return to [Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric](#), on page 205.

Example: DHCP Client and DHCP Server are in Different Tenant VRFs

This example shows how to configure DHCP relay deployment in a BGP EVPN VXLAN fabric for the topology in [Figure 12: DHCP Relay Deployment in a BGP EVPN VXLAN Fabric](#) when the DHCP server and the DHCP client are in different tenant VRFs. The DHCP server is reachable over a VRF that is different from the client's VRF.

The following tables provide sample configurations for the DHCP server and VTEP 1:

Table 24: Configuring DHCP when DHCP Server and DHCP Client are in Different Tenant VRFs**DHCP Configuration Snippet**

```
<snip: only the relevant configuration is shown>

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 101-102,201-202
ip dhcp snooping
!
interface Loopback101
vrf forwarding green
ip address 10.1.251.1 255.255.255.255
!
interface Vlan201
vrf forwarding red
ip dhcp relay source-interface Loopback101
ip address 10.2.201.1 255.255.255.0
ip helper-address vrf green 192.168.20.20

<snip: only the relevant configuration is shown>
```

Table 25: Configuring VTEP 1 when DHCP Server and DHCP Client are in Different Tenant VRFs

VTEP 1
<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 ! vrf definition red rd 2:2 ! address-family ipv4 route-target export 2:2 route-target import 2:2 route-target export 2:2 stitching route-target import 2:2 stitching exit-address-family ! ip routing ! ip multicast-routing ! 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 101-102,201-202 ip dhcp snooping ! 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 ! l2vpn evpn instance 201 vlan-based encapsulation vxlan ! l2vpn evpn instance 202 vlan-based encapsulation vxlan ! system mtu 9198 ! </pre>

VTEP 1

```
vlan configuration 101
member evpn-instance 101 vni 10101
vlan configuration 102
member evpn-instance 102 vni 10102
vlan configuration 201
member evpn-instance 201 vni 10201
vlan configuration 202
member evpn-instance 202 vni 10202
vlan configuration 901
member vni 50901
vlan configuration 902
member vni 50902
!
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 Loopback101
vrf forwarding green
ip address 10.1.251.1 255.255.255.255
!
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 GigabitEthernet1/0/2
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 dhcp relay source-interface Loopback101
ip address 10.1.101.1 255.255.255.0
ip helper-address 192.168.20.20
!
interface Vlan102
vrf forwarding green
ip dhcp relay source-interface Loopback101
ip address 10.1.102.1 255.255.255.0
ip helper-address 192.168.20.20
```

VTEP 1

```

interface Vlan201
vrf forwarding red
ip dhcp relay source-interface Loopback101
ip address 10.2.201.1 255.255.255.0
ip helper-address vrf green 192.168.20.20
!
interface Vlan202
vrf forwarding red
ip dhcp relay source-interface Loopback101
ip address 10.2.202.1 255.255.255.0
ip helper-address vrf green 192.168.20.20
!
interface Vlan901
vrf forwarding green
ip unnumbered Loopback0
no autostate
!
interface Vlan902
vrf forwarding red
ip unnumbered Loopback0
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 mcast-group 225.0.0.102
member vni 10201 mcast-group 225.0.0.201
member vni 10202 mcast-group 225.0.0.202
member vni 50901 vrf green
member vni 50902 vrf red
!
router ospf 1
router-id 172.16.255.3
!
router bgp 65001
bgp router-id interface Loopback0
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
!

```

VTEP 1

```

address-family ipv4 vrf green
advertise l2vpn evpn
redistribute connected
redistribute static
exit-address-family
!
address-family ipv4 vrf red
advertise l2vpn evpn
redistribute connected
redistribute static
exit-address-family
!
ip pim rp-address 172.16.255.255
!
end
Leaf-01#

```

Return to [Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric, on page 205](#).

Example: DHCP Server is in a non-Default, non-VXLAN VRF and DHCP Client is in the Tenant VRF

This example shows how to configure DHCP relay deployment in a BGP EVPN VXLAN fabric for the topology in [Figure 12: DHCP Relay Deployment in a BGP EVPN VXLAN Fabric](#) when the DHCP server is in a non-default, non-VXLAN VRF and the DHCP client is in the tenant VRF. The DHCP server is reachable over a VRF that is different from the client's VRF.

The following tables provide sample configurations for the DHCP server and VTEP 1:

Table 26: Configuring DHCP when DHCP Server is in a non-Default, non-VXLAN VRF and DHCP Client is in the Tenant VRF

DHCP Configuration Snippet

<snip: only the relevant configuration is shown>

```

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 101-102,201-202
ip dhcp snooping
!
interface Loopback101
vrf forwarding green
ip address 10.1.251.1 255.255.255.255
!
interface Vlan201
vrf forwarding red
ip dhcp relay source-interface Loopback101
ip address 10.2.201.1 255.255.255.0
ip helper-address vrf green 192.168.20.20

```

<snip: only the relevant configuration is shown>

Table 27: Configuring VTEP 1 when the DHCP Server is in a non-Default, non-VXLAN VRF and the DHCP Client is in the Tenant VRF

VTEP 1
<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 ! vrf definition red rd 2:2 ! address-family ipv4 route-target export 2:2 route-target import 2:2 route-target export 2:2 stitching route-target import 2:2 stitching exit-address-family ! ip routing ! ip multicast-routing ! 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 101-102,201-202 ip dhcp snooping ! 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 ! l2vpn evpn instance 201 vlan-based encapsulation vxlan ! l2vpn evpn instance 202 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 </pre>

VTEP 1

```
vlan configuration 201
member evpn-instance 201 vni 10201
vlan configuration 202
member evpn-instance 202 vni 10202
vlan configuration 901
member vni 50901
vlan configuration 902
member vni 50902
!
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 Loopback101
vrf forwarding green
ip address 10.1.251.1 255.255.255.255
!
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 GigabitEthernet1/0/2
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 dhcp relay source-interface Loopback101
ip address 10.1.101.1 255.255.255.0
ip helper-address 192.168.20.20
!
interface Vlan102
vrf forwarding green
ip dhcp relay source-interface Loopback101
ip address 10.1.102.1 255.255.255.0
ip helper-address 192.168.20.20
```

VTEP 1

```

interface Vlan201
vrf forwarding red
ip dhcp relay source-interface Loopback101
ip address 10.2.201.1 255.255.255.0
ip helper-address vrf green 192.168.20.20
!
interface Vlan202
vrf forwarding red
ip dhcp relay source-interface Loopback101
ip address 10.2.202.1 255.255.255.0
ip helper-address vrf green 192.168.20.20
!
interface Vlan901
vrf forwarding green
ip unnumbered Loopback0
no autostate
!
interface Vlan902
vrf forwarding red
ip unnumbered Loopback0
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 mcast-group 225.0.0.102
member vni 10201 mcast-group 225.0.0.201
member vni 10202 mcast-group 225.0.0.202
member vni 50901 vrf green
member vni 50902 vrf red
!
router ospf 1
router-id 172.16.255.3
!
router bgp 65001
bgp router-id interface Loopback0
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
!

```

VTEP 1

```
address-family ipv4 vrf green
advertise l2vpn evpn
redistribute connected
redistribute static
exit-address-family
!
address-family ipv4 vrf red
advertise l2vpn evpn
redistribute connected
redistribute static
exit-address-family
!
ip pim rp-address 172.16.255.255
!
end
Leaf-01#
```

Return to [Configuration Examples for DHCP Relay in a BGP EVPN VXLAN Fabric](#), on page 205.

Additional References for DHCP Relay in a BGP EVPN VXLAN Fabric

Related Documents

Related Topic	Document Title
DHCP Server configuration in IOS XE EVPN VXLAN	Configure DHCP in IOS XE EVPN/VXLAN



CHAPTER 7

Configuring VXLAN-Aware Flexible Netflow

- [Information About VXLAN-Aware Flexible NetFlow, on page 225](#)
- [How to Configure VXLAN-Aware Flexible NetFlow, on page 225](#)
- [Configuration Examples for VXLAN-Aware Flexible NetFlow, on page 230](#)

Information About VXLAN-Aware Flexible NetFlow

Flexible NetFlow (FNF) uses flows to provide statistics for accounting, network monitoring, and network planning. VXLAN-aware FNF provides information about the VXLAN-encapsulated IPV4 and IPV6 packets in the network. VXLAN-aware FNF captures the VXLAN flow information for both bridged and routed traffic.

A flow is a unidirectional stream of packets that arrives on a source interface and has the same values for the keys. A key is an identified value for a field within the packet. You create a flow using a flow record to define the unique keys for your flow. FNF allows you to define an optimal flow record for a particular application by selecting the keys from a large collection of predefined fields. All key values must match for the packet to count in a given flow. Flows are stored in the FNF cache. You can export the data FNF gathers for your flow by using an exporter.

In a BGP EVPN VXLAN fabric, an FNF monitor is configured on the NVE interface on a VTEP and on the physical interface on a spine switch. For more information about FNF, see *Configuring Flexible NetFlow* module of the *Network Management Configuration Guide*.

How to Configure VXLAN-Aware Flexible NetFlow

To configure VXLAN-aware FNF, perform these steps:

1. Create a flow record by specifying key fields and non-key fields to the flow.
2. Create a flow exporter by specifying the export protocol and transport destination port, source, and other parameters.
3. Create a flow monitor based on the flow record and flow exporter.
4. Apply the flow monitor to the network virtualization edge (NVE) interface on the VTEPs.



Note The commands listed in this section are applicable only to VXLAN-aware FNF. For detailed steps to configure FNF, see *How to Configure Flexible Netflow* section in the *Configuring Flexible NetFlow* module of the *Network Management Configuration Guide*.

Configuring a Flow Record

To configure a flow record for VXLAN-aware FNF, perform the following steps:



Note All the **match** commands listed in this configuration task are mandatory.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	flow record <i>flow-record-name</i> Example: Device(config)# flow record vxlan_nf_record_input	Creates a flow record and enters flow record configuration mode. This command also allows you to modify an existing flow record. Note We recommend that you configure a unique flow record for each address family (IPv4 and IPv6) and also for each traffic direction (input and output). Ensure that the flow record for ingress traffic has the match commands configured with the input keyword. Ensure that the flow record for egress traffic has the match commands configured with the output keyword.
Step 4	match datalink vlan { input output } Example:	Configures the VLAN ID (for input or output traffic) as a key field for the FNF flow record.

	Command or Action	Purpose
	Device(config-flow-record)# match datalink vlan output	Note Ensure that you configure the vlan input and vlan output fields. These fields are required for VXLAN-aware FNF to work on EVPN input and output traffic flows.
Step 5	match routing vrf input Example: Device(config-flow-record)# match routing vrf input	Configures the VRF ID (for input or output traffic) as a key field for the FNF flow record. Note Ensure that you configure the vrf input field. This field is required for VXLAN-aware FNF to work on EVPN input and output traffic flows.
Step 6	match vxlan vtep {input output} Example: Device(config-flow-record)# match vxlan vtep output	Configures the VTEP ID as a key field for the FNF flow record. The input keyword shows the VTEP source IP address in the captured flow. The output keyword shows the VTEP destination IP address in the captured flow
Step 7	match vxlan vniid Example: Device(config-flow-record)# match vxlan vniid	Configures the VXLAN VNI ID as a key field for the FNF flow record.
Step 8	end Example: Device(config-flow-record)# end	Returns to privileged EXEC mode.

Configuring Flow Exporter

To configure flow exporter for VXLAN-aware FNF, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# configure terminal	
Step 3	flow exporter <i>flow-exporter-name</i> Example: Device(config)# flow exporter e1	Creates a flow exporter and enters flow exporter configuration mode.
Step 4	destination <i>ipv4-address</i> Example: Device(config-flow-exporter)# destination 172.16.103.2	Sets the IPv4 destination address or hostname for the exporter.
Step 5	source <i>interface-type interface-number</i> Example: Device(config-flow-exporter)# source TenGigabitEthernet1/5/0/3	Specifies the interface to use to reach the NetFlow collector at the configured destination. Ensure that the source IP address is unique per fabric. Note We recommend that you configure a unique Loopback on each VTEP. Note Flow exporter does not support unnumbered IP interface as source interface.
Step 6	ttl <i>seconds</i> Example: Device(config-flow-exporter)# ttl 4	Configures the time-to-live (TTL) value for datagrams sent by the exporter. The range is from 1 to 255 seconds. The default is 255.
Step 7	transport udp <i>port-number</i> Example: Device(config-flow-exporter)# transport udp 2055	Specifies the UDP port to use to reach the NetFlow collector.
Step 8	export-protocol { ipfix netflow-v9 } Example: Device(config-flow-exporter)# export-protocol ipfix	Specifies the version of the NetFlow export protocol used by the exporter.
Step 9	end Example: Device(config-flow-exporter)# end	Returns to privileged EXEC mode.

Configuring a Flow Monitor

To configure a flow monitor for VXLAN-aware FNF, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	flow monitor <i>flow-monitor-name</i> Example: Device(config)# flow monitor vxlan_nf_monitor_input	Creates a flow monitor and enters flow monitor configuration mode. This command also allows you to modify an existing flow monitor.
Step 4	exporter <i>flow-exporter-name</i> Example: Device(config-flow-monitor)# exporter e1	Specifies the name of the flow exporter that was created previously and associates it with the specified flow monitor.
Step 5	record <i>flow-record-name</i> Example: Device(config-flow-monitor)# record vxlan_nf_record_input	Specifies the record for the flow monitor.
Step 6	end Example: Device(config-flow-monitor)# end	Returns to privileged EXEC mode.

Configuring Flexible NetFlow on an NVE Interface

To configure VXLAN-aware FNF on the NVE interface of a VTEP, perform the following steps:

Procedure

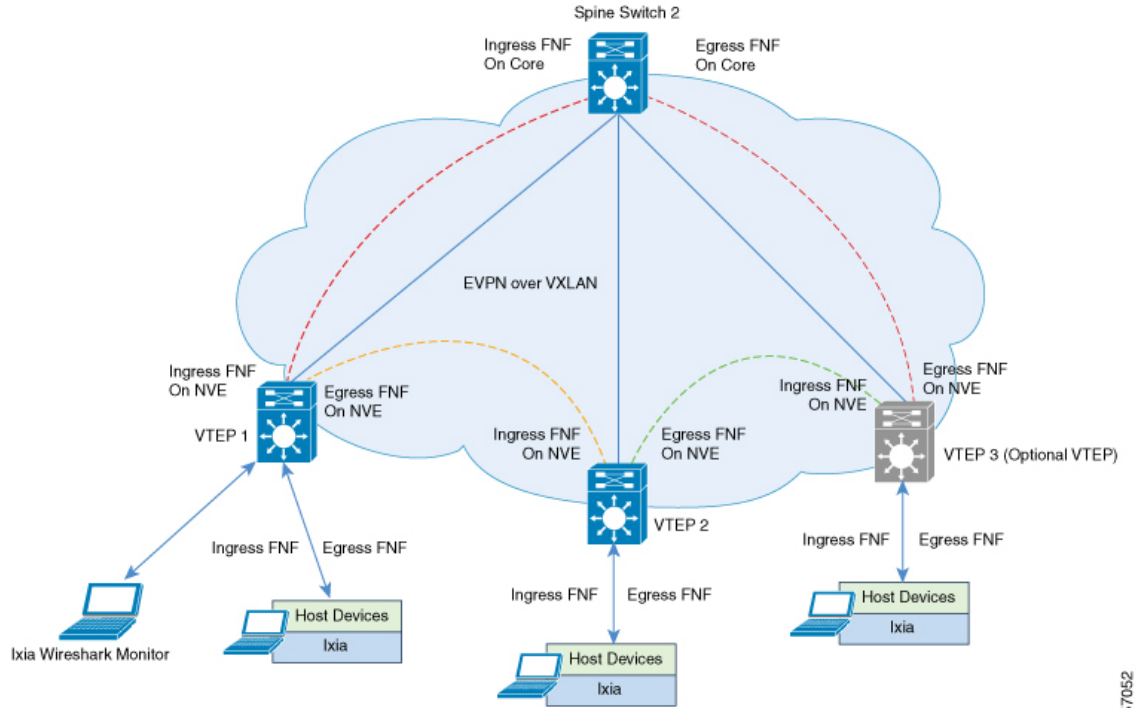
	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>nve-interface-number</i> Example: Device(config)# interface nve1	Specifies the network virtualization edge (NVE) interface number and enters interface configuration mode.
Step 4	ip flow monitor <i>flow-monitor-name</i> { input output } Example: Device(config-if)# ip flow monitor vxlan_nf_monitor_input input	Associates the IPv4 flow monitor to the NVE interface for input or output packets.
Step 5	ipv6 flow monitor <i>flow-monitor-name</i> { input output } Example: Device(config-if)# ipv6 flow monitor vxlan_nf_v6monitor_input input	Associates the IPv6 flow monitor to the NVE interface for input or output packets.
Step 6	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuration Examples for VXLAN-Aware Flexible NetFlow

This section provides configuration examples for VXLAN-aware FNF using the following topology:

Figure 13: EVPN VXLAN Topology with VXLAN-Aware Flexible NetFlow



357052

Configuring VTEP 1 to enable VXLAN-Aware Flexible NetFlow

The following table provides a sample configuration for VTEP 1 to enable VXLAN-aware FNF:

Table 28: Configuring VTEP 1 to enable VXLAN-Aware Flexible NetFlow

VTEP 1
<pre> Leaf-01# show running-config <snip: only config relevant to vxlan netflow is shown> flow record vxlan_nf_record_input match datalink vlan input match datalink mac source address input match datalink mac destination address input match routing vrf input match ipv4 ttl match ipv4 protocol match ipv4 source address match ipv4 destination address match transport source-port match transport destination-port match transport icmp ipv4 type match transport icmp ipv4 code match transport igmp type match interface input match flow direction match vxlan vnid match vxlan vtep input match vxlan vtep output collect counter bytes long collect counter packets long collect timestamp absolute first collect timestamp absolute last ! flow record vxlan_nf_record_output match datalink mac destination address output match ipv4 protocol match ipv4 source address match ipv4 destination address match transport source-port match transport destination-port match datalink vlan output match vxlan vnid match vxlan vtep input match vxlan vtep output collect counter bytes long collect counter packets long collect timestamp absolute first collect timestamp absolute last ! </pre>

VTEP 1

```
flow record vxlan_nf_v6record_input
match datalink vlan input
match routing vrf input
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
match vxlan vnid
match vxlan vtep input
match vxlan vtep output
collect counter bytes long
collect counter packets long
collect timestamp absolute first
collect timestamp absolute last
!
flow record vxlan_nf_v6record_output
match datalink vlan output
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
match vxlan vnid
match vxlan vtep input
match vxlan vtep output
collect counter bytes long
collect counter packets long
collect timestamp absolute first
collect timestamp absolute last
!
flow exporter e1
destination 172.16.103.2
source TenGigabitEthernet1/5/0/3
ttl 4
transport udp 2055
export-protocol ipfix
!
flow monitor vxlan_nf_monitor_input
exporter e1
cache timeout inactive 100
cache timeout active 100
record vxlan_nf_record_input
!
!
flow monitor vxlan_nf_monitor_output
exporter e1
cache timeout inactive 100
cache timeout active 100
record vxlan_nf_record_output
!
!
flow monitor vxlan_nf_v6monitor_input
exporter e1
cache timeout inactive 100
cache timeout active 100
record vxlan_nf_v6record_input
!
!
```



```

(13vni5001)          12 AAAA.CCCC.1004          AAAA.BBBB.1004          3
                    192.168.12.3          192.168.12.2          0          0
                    0          0 Null          Input          10012
2.2.2.2              1.1.1.1              61          64          43517376
                    43172          14:00:41.391          14:01:34.391

Leaf-01#
    
```

Checking IPv4 Output Flow Monitor Cache Output

The following example provides a sample output to check the IPv4 output flow monitor cache output on VTEP 1:

```

Leaf-01# configure terminal
Leaf-01(config)# show flow monitor vxlan_nf_monitor_output cache format table

Cache type:                               Normal (Platform cache)

Cache size:                               10000

Current entries:                          4

Flows added:                              8

Flows aged:                              4

- Inactive timeout ( 100 secs)           4

DATALINK MAC DST ADDR OUTPUT  IPV4 SRC ADDR  IPV4 DST ADDR  TRNS SRC PORT  TRNS DST
PORT  DATALINK VLAN OUTPUT    VXLAN VXLAN VNID  VXLAN VXLAN VTEP INPUT  VXLAN VXLAN VTEP
OUTPUT IP PROT                bytes long      pkts long  time abs first  time abs last
=====
=====
=====
=====
=====
AAAA.CCCC.1002          192.168.10.2   192.168.10.3   0
0                      10            10010  1.1.1.1        2.2.2.2
61                    44812536      43172          14:00:41.391   14:01:34.391
AAAA.CCCC.1004          192.168.12.2   192.168.12.3   0
0                      12            10012  1.1.1.1        2.2.2.2
61                    44812536      43172          14:00:41.391   14:01:34.391
AAAA.CCCC.1003          192.168.13.2   192.168.13.3   0
0                      13            10013  1.1.1.1        2.2.2.2
61                    44812536      43172          14:00:41.391   14:01:34.391
AAAA.CCCC.1001          192.168.11.2   192.168.11.3   0
0                      11            10011  1.1.1.1        2.2.2.2
61                    44812536      43172          14:00:41.391   14:01:34.391

Leaf-01#
    
```

Checking IPv6 Input Flow Monitor Cache Output

The following example provides a sample output to check the IPv6 input flow monitor cache output on VTEP 1:

```

Leaf-01# configure terminal
Leaf-01(config)# show flow monitor vxlan_nf_v6monitor_input cache format table

Cache type:                               Normal (Platform cache)

Cache size:                               10000
    
```




CHAPTER 8

Configuring Tenant Routed Multicast

- [Restrictions for Tenant Routed Multicast](#) , on page 237
- [Information about Tenant Routed Multicast](#), on page 238
- [How to Configure Tenant Routed Multicast](#), on page 250
- [Verifying Tenant Routed Multicast](#), on page 257
- [Troubleshooting Tenant Routed Multicast](#), on page 258
- [Configuration Examples for Tenant Routed Multicast](#), on page 258

Restrictions for Tenant Routed Multicast

- Layer 2 tenant routed multicast (TRM) is not supported. Only Layer 3 TRM is supported.
- TRM uses only default multicast distribution tree (MDT) for handoff to external MVPN networks.
- In the underlay network, the default MDT supports only PIM sparse mode, and Data MDT supports PIM sparse mode and PIM source specific multicast (SSM) mode.



Note The spine switch can be configured as the rendezvous point (RP). The RP for the underlay network can also be elected using PIM Bootstrap Router (BSR) or Auto-RP.

- TRM does not support dual-homing of hosts.
- TRM data MDT is supported for TRM in PIM sparse mode and TRM in PIM source specific mode for IPv4 connections only.
- TRM data MDT supports IPv4 C-multicast flows only.
- TRM data MDT supports rate-based switchover only for IPv4 multicast flows.
- TRM data MDT supports Layer 2, IP, and VRF-Lite handoffs only.

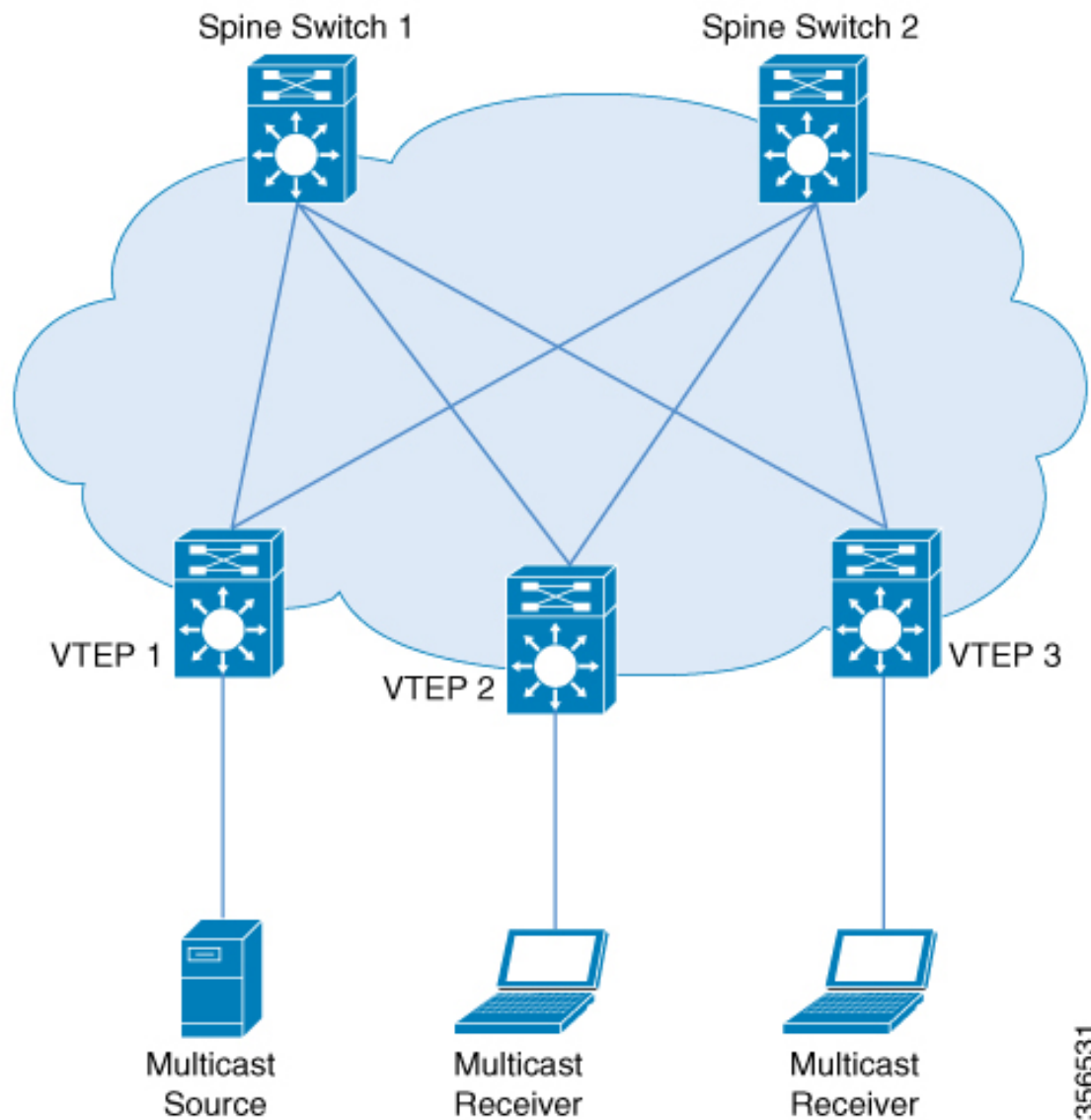
Information about Tenant Routed Multicast

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.

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 in 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. Thus, TRM allows a seamless integration with existing multicast networks with newer enterprise fabric.

Figure 14: Tenant Routed Multicast Topology



356531

For IPv4 and IPv6 multicast traffic, TRM uses BGP EVPN and multicast virtual private network (MVPN) routes to perform multicast routing. All the VTEPs in the network 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. Source reachability is distributed via EVPN route type 2 and EVPN route type 5 in the fabric. RPF is installed based on these routes. Source-active and receiver-join information is carried in the MVPN address family using route types 5, 6, and 7.

In an EVPN VXLAN network, TRM is supported in the overlay network in PIM sparse mode and PIM source specific multicast (SSM) mode. VTEPs have BGP peering in MVPN as well as EVPN address families to exchange routes for TRM.

TRM in PIM Sparse Mode

PIM-SM distributes information about active sources by forwarding data packets on the shared tree. Because PIM-SM uses shared trees, it requires the use of a rendezvous point (RP). An RP is used for the initial convergence of multicast traffic between sources and receivers.

The following section describes the different ways the RP can be configured for TRM in PIM sparse mode.

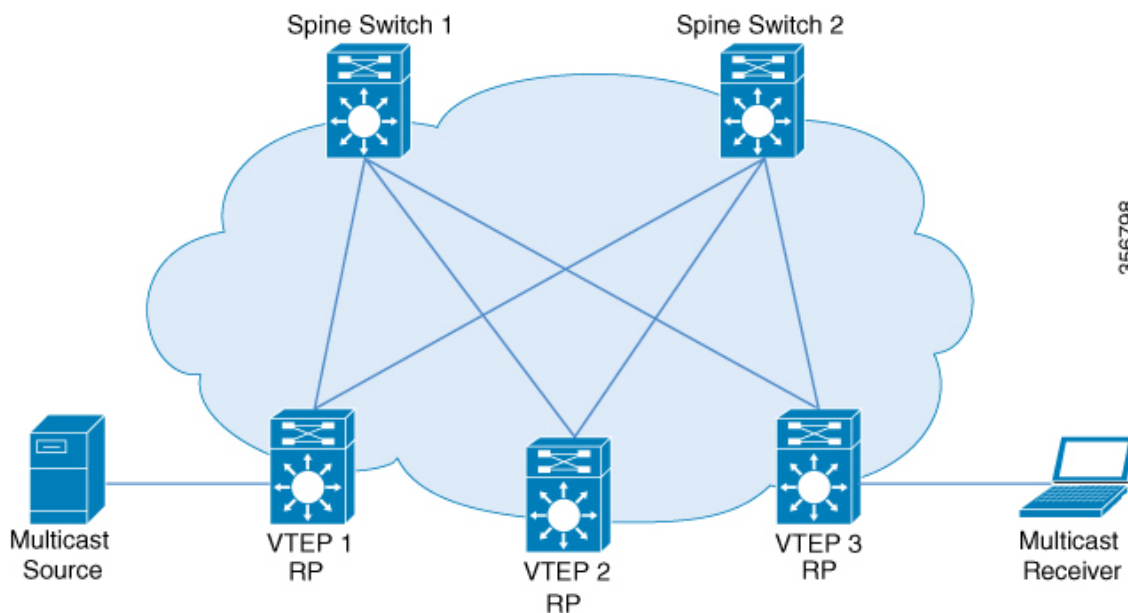
RP Placement

For TRM in PIM sparse mode, the overlay RP can be configured either within the BGP EVPN VXLAN fabric, or outside of the fabric.

Anycast RP in the Overlay Network

In Anycast RP in the overlay network, every VTEP acts as an RP.

Figure 15: Anycast RP in the Overlay Network



RP Inside the BGP EVPN VXLAN Fabric

In a TRM scenario where all sources and receivers are within the EVPN VXLAN network, the overlay RP can be placed on a border spine or on a VTEP.

Figure 16: Border Spine Switch as an RP

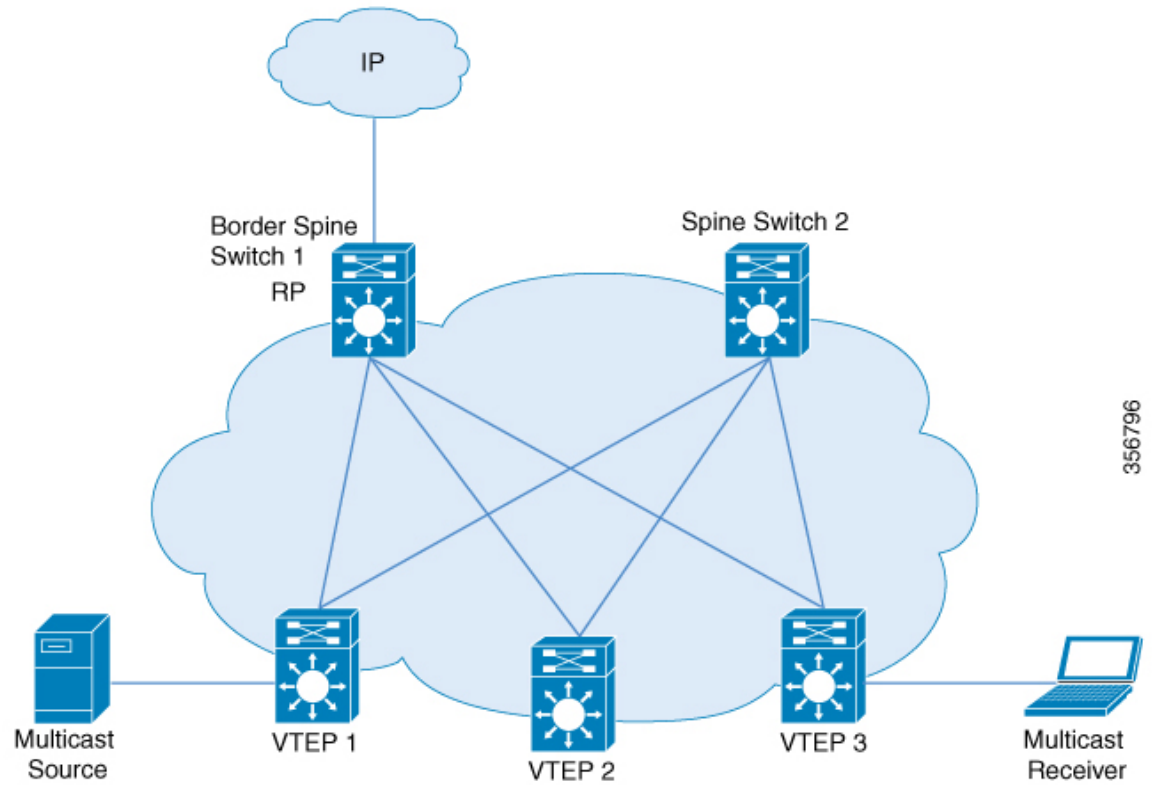
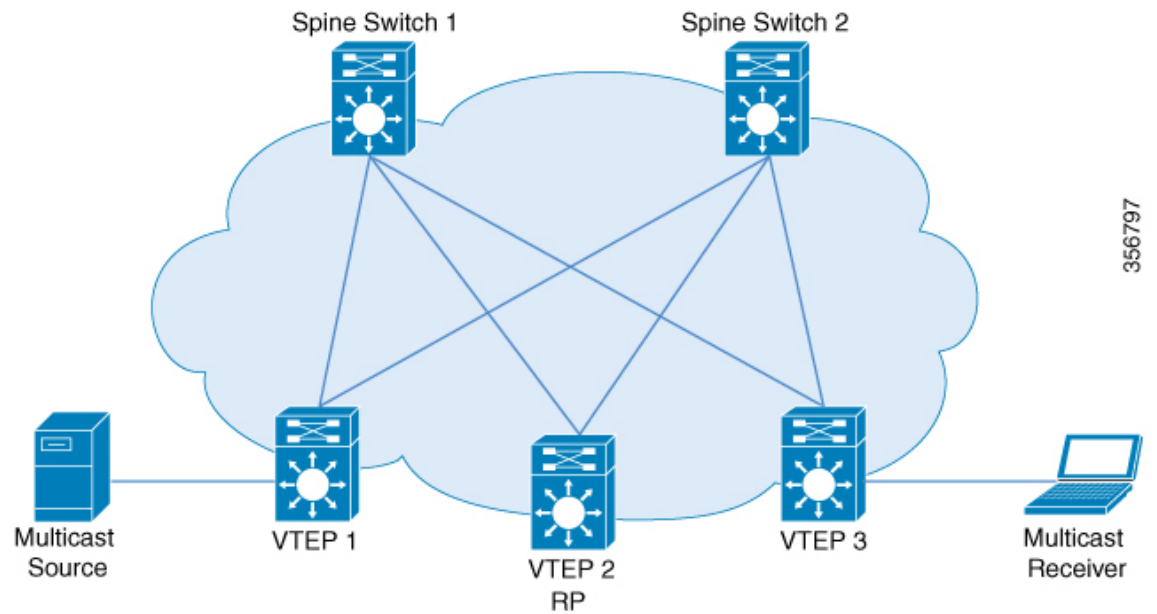


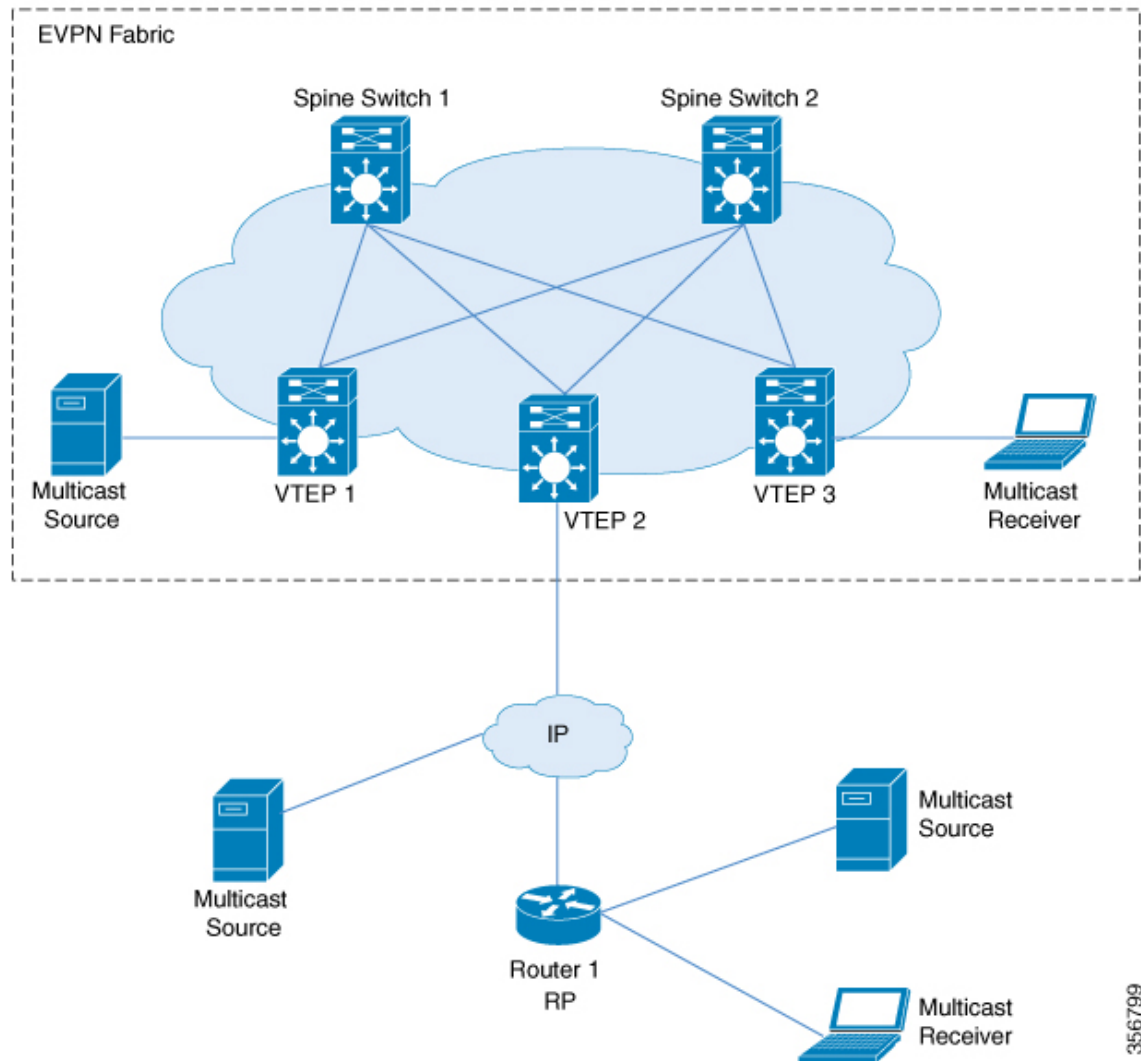
Figure 17: VTEP as an RP



RP Outside the BGP EVPN VXLAN Fabric

In a TRM scenario where sources and receivers need to interwork with the overlay network and the EVPN VXLAN network, the RP can be placed on an external router connected to a VTEP.

Figure 18: RP Outside the BGP EVPN VXLAN Fabric



In PIM sparse mode, TRM can be configured in three different ways depending on how the RP is configured:

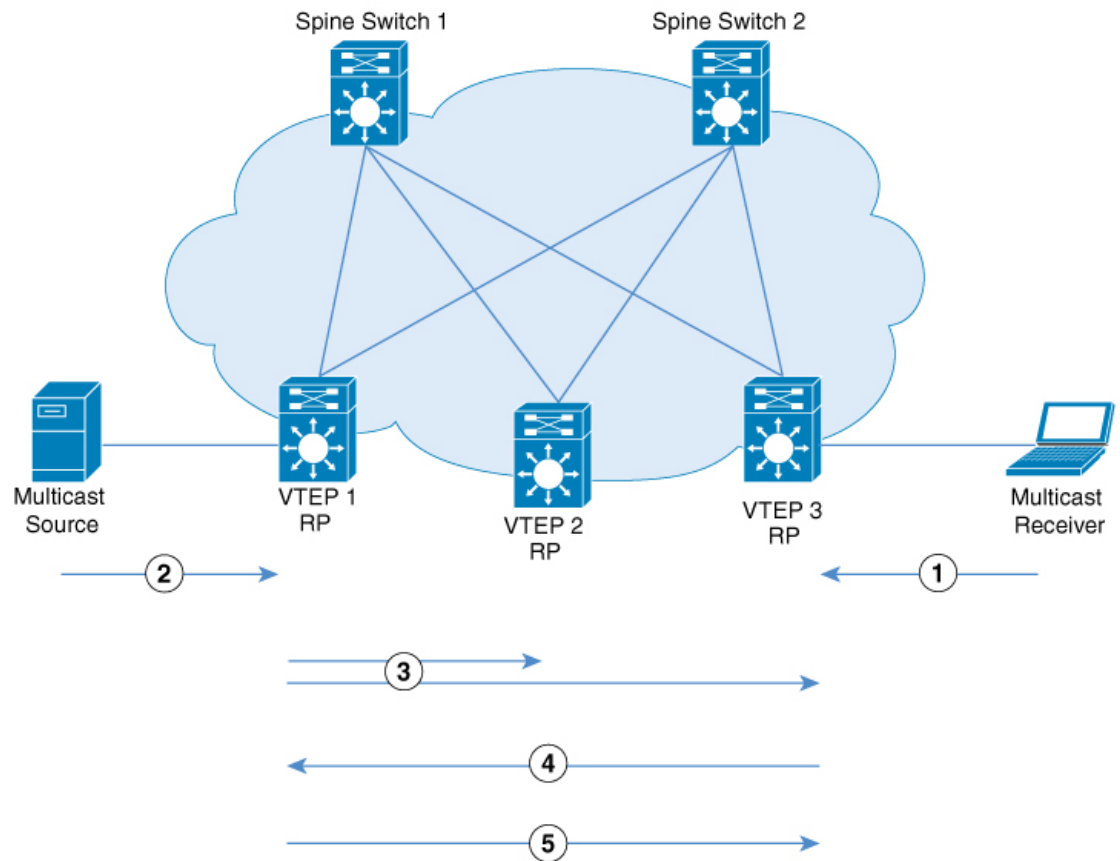
- PIM Sparse Mode with Anycast RP
- PIM Sparse Mode with RP Inside the BGP EVPN VXLAN Fabric
- PIM Sparse Mode with RP Outside the BGP EVPN VXLAN Fabric

PIM Sparse Mode with Anycast RP

In PIM sparse mode with anycast RP, every VTEP in the EVPN VXLAN network acts as an RP in the overlay network for its respective multicast group. The RPs in the underlay network must be configured on the spine switches.

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 19: PIM Sparse Mode with Anycast RP



In PIM sparse mode with anycast RP, 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.
3. VTEP 1 performs self-source-registration since it is also an RP.

The source VTEP (VTEP 1) advertises Source A-D Routes (also called MVPN route type 5) for the (S,G) to all the other VTEPs which are BGP peers in the MVPN address family.

4. VTEP 2 and VTEP 3 receive and install the Source A-D Route for the (S,G).

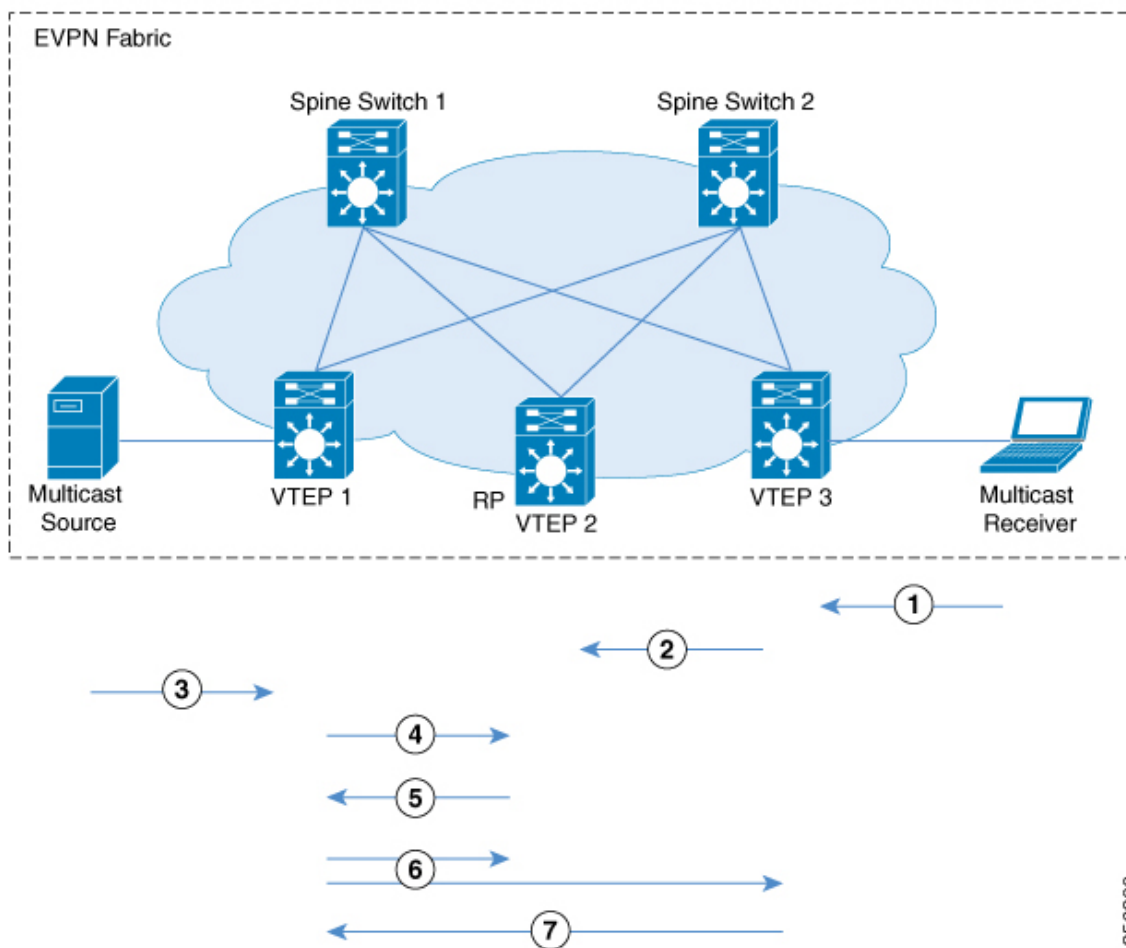
(S,G) is created at VTEP 3. VTEP 3 now has an overlay route for the (S,G) and also has a unicast route to the source device from the EVPN Control plane. It then sends an MVPN route type 7 (S,G) BGP join to VTEP 1 and starts accepting traffic.

- 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 the (S,G) and starts forwarding traffic.

PIM Sparse Mode with RP Inside the BGP EVPN VXLAN Fabric

In PIM sparse mode with RP inside the BGP EVPN VXLAN Fabric, the RP can be any VTEP in the EVPN VXLAN network.

Figure 20: PIM Sparse Mode with RP Inside the BGP EVPN VXLAN Fabric



The following sequence of events occurs when TRM is enabled in PIM sparse mode with the RP inside the fabric:

- Receiver sends (*,G) IGMP Join to VTEP 3. (*,G) is created at VTEP 3.
- VTEP 3 sends MVPN route type 6 to VTEP 2 which is the RP. (*,G) is created at VTEP 2.
- The source device starts streaming data and (S,G) is created on VTEP 1.

4. VTEP 1 performs source-registration at VTEP 2 since it is the RP. (S,G) is created at VTEP 2.
5. Since the RP has a receiver for (S,G), it sends an MVPN route type 7 to VTEP 1 and forwards PIM-register traffic towards receivers in the (*,G) tree.
6. VTEP 1 receives and installs MVPN route type 7 from VTEP 2. It uses the Layer 3 VNI's SVI as the forwarding interface for (S,G).

The source VTEP (VTEP 1) advertises Source A-D Routes for (S,G) to all the other VTEPs which are BGP peers in the MVPN address family.

VTEP 2 and VTEP 3 receive and install the Source A-D Routes for (S,G).

7. (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 to VTEP 1 and starts accepting traffic.

VTEP 1 receives and installs MVPN route type 7 from VTEP 3 and starts forwarding traffic.



Note For the receiver VTEP to be able to send an MVPN route type 7 to the source VTEP, there can be two triggers:

- The (*,G) packets being forwarded to the receiver VTEP from the RP.
- The Source A-D route received from the source VTEP.

Once either of these are received, the receiver VTEP sends MVPN route type 7 to the source VTEP.

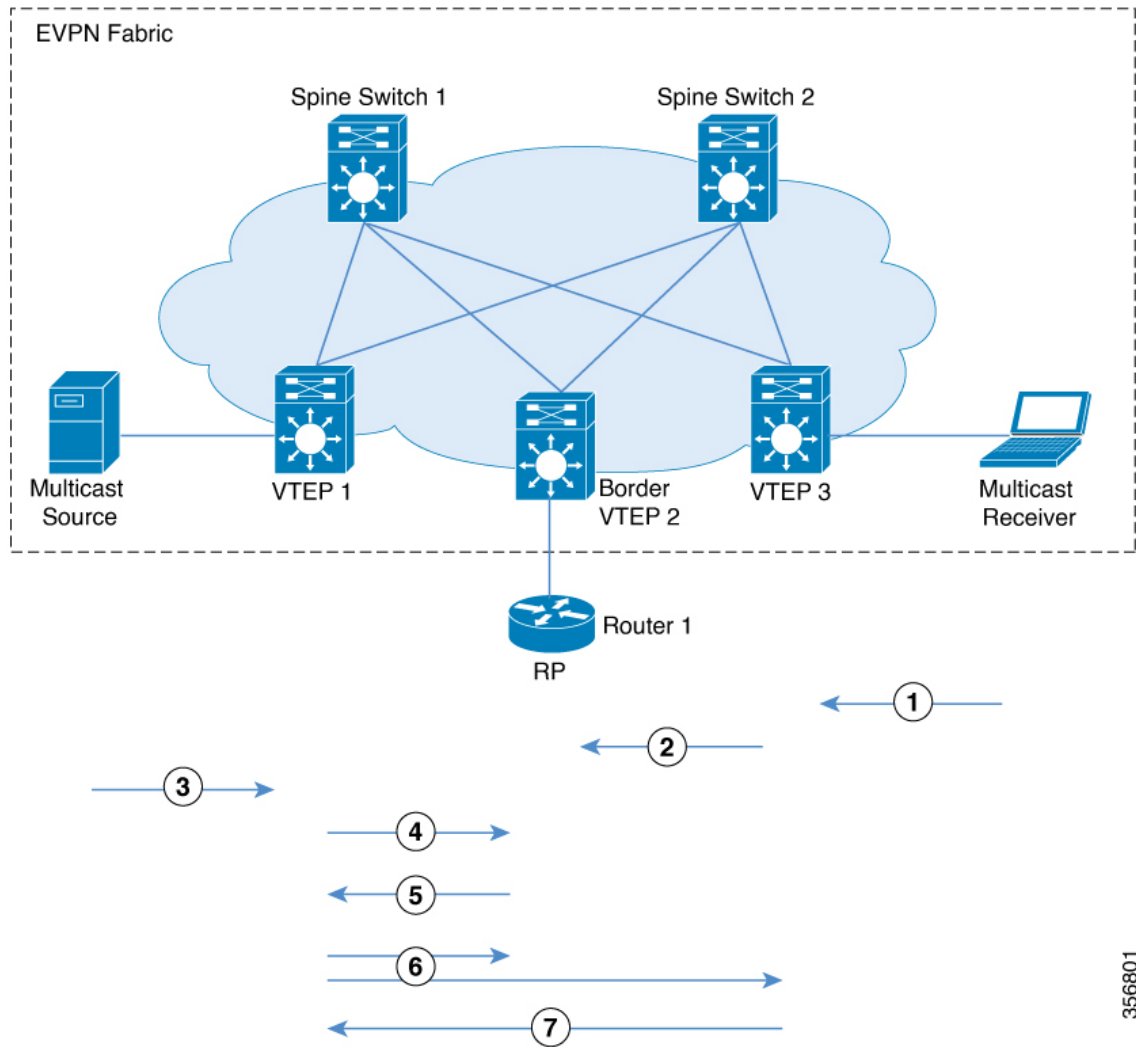
PIM Sparse Mode with RP Outside the BGP EVPN VXLAN Fabric

In PIM sparse mode with RP outside the BGP EVPN VXLAN Fabric, the RP can be a PIM router behind any VTEP in the EVPN VXLAN network.



Note When the RP is configured outside the BGP EVPN VXLAN fabric, TRM in PIM sparse mode functions the same way as it does when the RP is inside the fabric.

Figure 21: PIM Sparse Mode with RP Outside the BGP EVPN VXLAN Fabric



The chronological traffic flow from the image above is as follows:

1. Receiver sends (*,G) IGMP Join to VTEP 3. (*,G) is created at VTEP 3.
2. VTEP3 sends and MVPN route type 6 to VTEP 2 which has the RP in its overlay network. This route is converted to a (*,G) join towards the RP by VTEP2.
3. The source device starts streaming data and (S,G) is created on VTEP 1.
4. VTEP1 performs source registration with RP. (S,G) join from RP creates (S,G) state at VTEP 2.
5. Since the RP has a receiver for (S,G), it sends an MVPN route type 7 to VTEP 1 and forwards PIM-register traffic towards receivers in the (*,G) tree.
6. VTEP 1 receives and installs MVPN route type 7 from VTEP 2. It uses the Layer 3 VNI's SVI as the forwarding interface for (S,G).

The source VTEP (VTEP 1) advertises Source A-D Routes for (S,G) to all the other VTEPs which are BGP peers in the MVPN address family.

356801

VTEP 2 and VTEP 3 receive and install the Source A-D Routes for (S,G).

7. (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 to VTEP 1 and starts accepting traffic.

VTEP 1 receives and installs MVPN route type 7 from VTEP 3 and starts forwarding traffic.



Note For the receiver VTEP to be able to send an MVPN route type 7 to the source VTEP, there can be two triggers:

- The (*,G) packets being forwarded to the receiver VTEP from the RP.
- The Source A-D route received from the source VTEP.

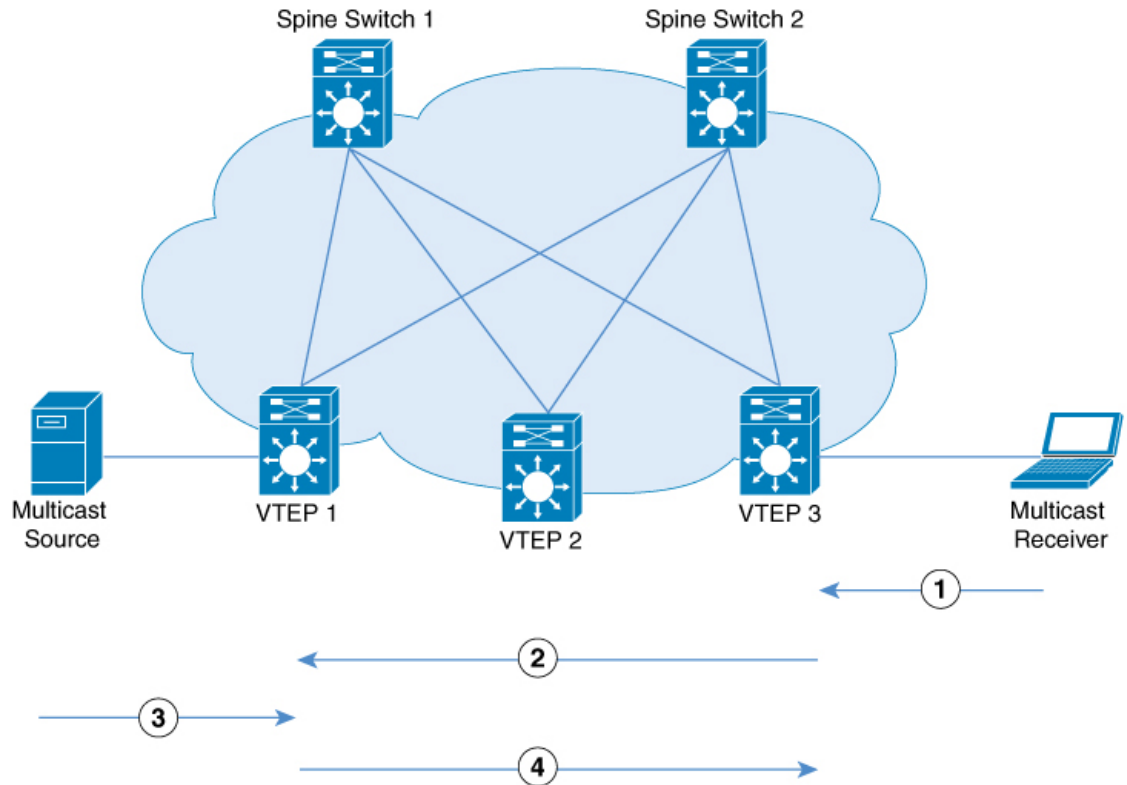
Once either of these are received, the receiver VTEP sends MVPN route type 7 to the source VTEP.

See [PIM Sparse Mode with RP Inside the BGP EVPN VXLAN Fabric, on page 244](#) for the sequence of events that happen when TRM is enabled in PIM sparse mode with the RP outside the fabric.

TRM in PIM Source Specific Mode

In PIM source specific 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 22: PIM Source Specific Mode



In PIM Source Specific 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.
The receiver sends an (S,G) IGMP join towards VTEP 3 and an (S,G) entry is created.
2. 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.
3. 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).
The source device sends (S,G) data to VTEP 1.
4. VTEP 1 starts forwarding the traffic to VTEP 3.

Data MDT

Data MDTs are purpose built underlay MDTs to provide optimized forwarding in the MVPN and EVPN core. Threshold and access control configurations can be used to control the characteristics of overlay streams for which data MDTs will be built. The threshold at which the data MDT is created can be configured on a per-VRF basis only. When the multicast transmission exceeds the defined threshold, the sending VTEP device creates the data MDT and sends a MVPN route type 3 message, which contains information about the data MDT, to all devices on the default MDT. The statistics to determine whether a multicast stream has exceeded

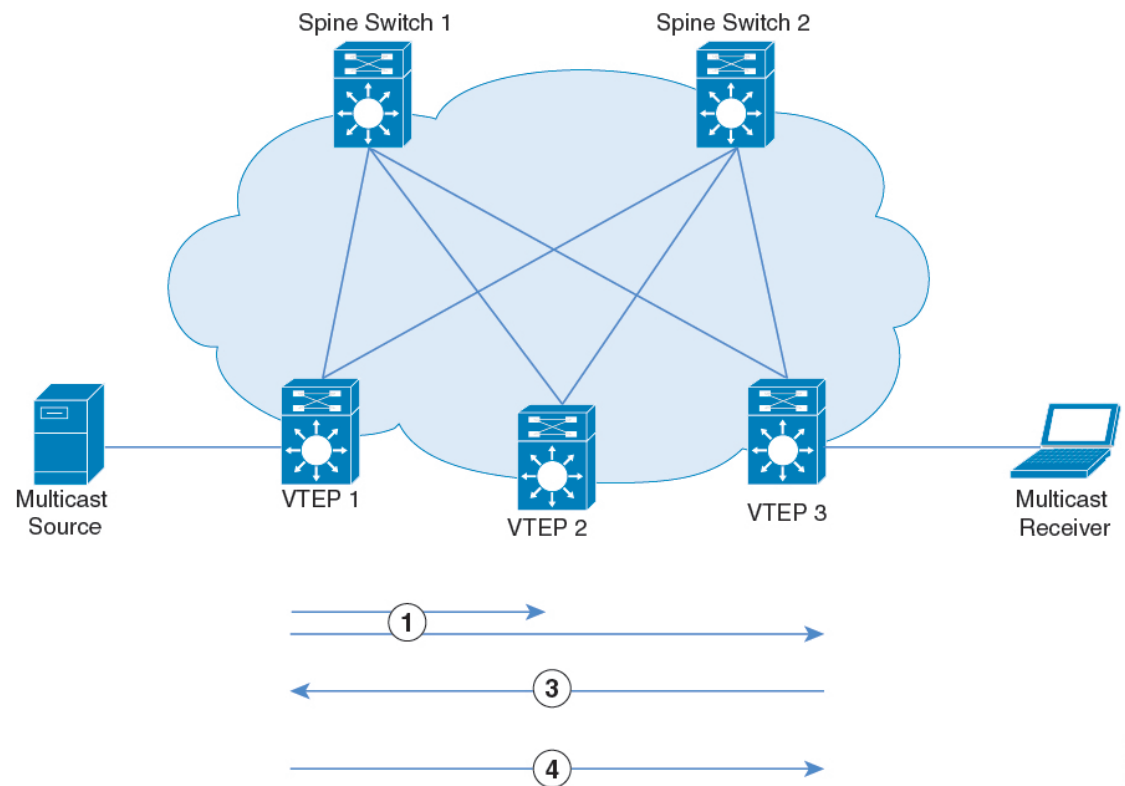
the data MDT threshold are examined periodically. After a VTEP device sends the MVPN route type 3 message, it waits 3 more seconds before switching over.

With Data MDT, replication load on the EVPN spine nodes are greatly reduced and there are less number of copies in the EVPN underlay overall. DataMDT lowers the bandwidth usage between the spine nodes and VTEPs. The traffic load at VTEPs that do not require certain overlay streams are completely avoided. Data MDT also offers flexibility in deployment through ACLs and threshold-based switchovers, and also enables load-balancing capabilities.

Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate. An ACL can be configured to control the overlay streams that will be allowed onto the data MDT irrespective of the threshold.

Data MDT is supported for all TRM modes: PIM sparse mode with anycast RP, PIM sparse mode with RP inside the BGP EVPN VXLAN fabric, PIM sparse mode with RP outside the BGP EVPN VXLAN fabric, and PIM source specific multicast mode.

Figure 23: Data MDT Mode



After the sequence of events of PIM sparse mode with RP outside the BGP EVPN VXLAN fabric or PIM source specific mode is completed, continue with the following sequence of events for Data MDT mode:

1. VTEP1 sends MVPN route type 3 to all other nodes. The threshold is exceeded for stream and MVPN route type 3 is sent to advertise data MDT for the stream. After 3 seconds, VTEP1 switches stream from the default MDT to data MDT using MVPN route type 3. Only VTEP3 receives stream data.
2. VTEP2 has no receiver, and so it does not join the advertised data MDT underlay.

3. VTEP3 has a receiver, and joins the advertised data MDT underlay.

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 99](#) 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 TRM with PIM Sparse Mode

To configure TRM with PIM Sparse Mode, perform the following tasks:

- [Configuring the TRM Multicast Distribution Tree in the VRF, on page 250](#)
- [Configuring Multicast Routing on the Overlay VRF, on page 252](#)
- [Configuring Multicast on Switch Virtual Interfaces for the Core-facing and Access-facing VLANs, on page 252](#)
- [Configuring BGP with MVPN Address Family on VTEP, on page 253](#)
- [Configuring RP for Underlay Network, on page 254](#)
- [Configuring RP for Overlay Network, on page 255](#)

Configuring the TRM Multicast Distribution Tree in the VRF

To configure the TRM MDT, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Device (config)# vrf definition green	Names the VRF and enters VRF configuration mode.

	Command or Action	Purpose
Step 4	address-family { ipv4 ipv6 } Example: Device (config-vrf) # address-family ipv4	Specifies the VRF and enters VRF address family configuration mode. <ul style="list-style-type: none"> • Use the ipv4 keyword to configure IPv4 address family. • Use the ipv6 keyword to configure IPv6 address family.
Step 5	mdt default vxlan group-address Example: Device (config-vrf-af) # mdt default vxlan 225.2.2.2	Configures the multicast group address range for default MDT groups for a VRF in a VXLAN.
Step 6	mdt auto-discovery vxlan [inter-as] Example: Device (config-vrf-af) # mdt auto-discovery vxlan	Enables VXLAN with BGP auto-discovery. Use the inter-as keyword for the MVPN address family routes to cross the BGP autonomous system (AS) boundaries.
Step 7	mdt overlay use-bgp [spt-only] Example: Device (config-vrf-af) # mdt overlay use-bgp spt-only	Configures the mechanism that is used by TRM in PIM sparse mode to operate within the BGP EVPN VXLAN fabric. Specifies BGP as the overlay protocol. <ul style="list-style-type: none"> • Use the mdt overlay use-bgp spt-only command to configure PIM sparse mode with anycast RP. • Use the mdt overlay use-bgp command to configure PIM sparse mode with a single RP either inside or outside the BGP EVPN VXLAN fabric.
Step 8	mdt data vxlan data-mdt-subnet data-mdt-mask [list access-list-number] Example: Device (config-vrf-af) # mdt data vxlan 225.2.2.0 0.0.0.255 list 101	(Optional) Configures the multicast group address range for data MDT groups for a VRF in a VXLAN.
Step 9	mdt data threshold kb/s Example: Device (config-vrf-af) # mdt data threshold 111	(Optional) Defines the bandwidth threshold value in kilobits per second (kb/s). The range is from 1 to 4294967, and the default value is 0.
Step 10	exit-address-family Example: Device (config-vrf-af) # exit-address-family	Exits VRF address family configuration mode and returns to VRF configuration mode.

	Command or Action	Purpose
Step 11	end Example: Device(config-vrf)# end	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
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip multicast-routing vrf vrf-name Example: Device(config)# ip multicast-routing vrf green	Enables IP multicast forwarding on the overlay VRF.
Step 4	ipv6 unicast-routing Example: Device(config)# ipv6 unicast-routing	Enables IPv6 unicast forwarding.
Step 5	ipv6 multicast-routing vrf vrf-name Example: Device(config)# ipv6 multicast-routing vrf green	Enables IPv6 multicast forwarding on the overlay VRF.
Step 6	end Example: Device(config)# end	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

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface vlan <i>core-facing-vlan-id</i> Example: Device(config)# interface vlan 200	Enters interface configuration mode for the specified VLAN.
Step 4	ip pim sparse-mode Example: Device(config-if) # ip pim sparse-mode	Enables IPv4 multicast on the core-facing SVI.
Step 5	exit Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	interface vlan <i>access-facing-vlan-id</i> Example: Device(config)# interface vlan 202	Enters interface configuration mode for the specified VLAN.
Step 7	ip pim sparse-mode Example: Device(config-if) # ip pim sparse-mode	Enables IPv4 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.
Step 8	end Example: Device(config-if)# end	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

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

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 1	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
Step 4	address-family {ipv4 ipv6} mvpn Example: Device(config-router)# address-family ipv4 mvpn	Specifies the MVPN address family and enters address family configuration mode. <ul style="list-style-type: none"> • Use the ipv4 keyword to configure IPv4 MVPN address family. • Use the ipv6 keyword to configure IPv6 MVPN address family.
Step 5	neighbor ip-address activate Example: Device(config-router-af)# neighbor 10.2.2.20 activate	Enables the exchange of information with a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 6	neighbor ip-address send-community extended Example: Device(config-router-af)# neighbor 10.2.2.20 send-community both	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 7	neighbor ip-address advertisement-interval <i>seconds</i> Example: Device(config-router-af)# neighbor 10.2.2.20 advertisement-interval 10	(Optional) Sets the minimum route advertisement interval (MRAI) between the sending of BGP routing updates.
Step 8	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 9	end Example: Device(config-router)# end	Returns to privileged EXEC mode.

Configuring RP for Underlay Network

To configure RP for the underlay network, perform the following steps:



Note It is recommended that you configure the Spine Switch as the RP for the underlay network.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip pim rp-address ip-address-of-rp Example: Device(config)# ip pim rp-address <rp-ip-address>	Configures the RP in the underlay network. For information about RP redundancy, see to <i>IP Multicast Routing Configuration Guide</i> .
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

Configuring RP for Overlay Network

To configure RP for the overlay network, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface loopback-interface Example: Device(config)# interface Loopback 13	Enters interface configuration mode for the specified Loopback interface.
Step 4	vrf forwarding vrf-name Example:	Configures forwarding table for the Loopback interface.

	Command or Action	Purpose
	Device (config-if) # vrf forwarding green	
Step 5	ip-address <i>ip-address subnet-mask</i> Example: Device (config-if) # ip address 10.1.13.13 255.255.255.255	Configures the IP address for the Loopback interface.
Step 6	ip pim sparse-mode Example: Device (config-if) # ip pim sparse-mode	Enables IPv4 multicast on the Loopback interface. Note 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.
Step 7	exit Example: Device (config-if) # exit	Returns to global configuration mode.
Step 8	{ip ipv6 } pim vrf vrf-name rp-address rp-address Example: Device (config) # ip pim vrf green rp-address 10.1.13.13	Configures the address of the local VTEP as the PIM RP for the multicast group. <ul style="list-style-type: none">• In PIM-SM with Anycast RP mode, use the address of the loopback interface of the local VTEP.• In PIM-SM with RP either inside or outside the BGP EVPN VXLAN fabric, use the IP address of the RP. Note The loopback interface specified must be part of the same VRF.
Step 9	{ip ipv6 } pim vrf vrf-name register-source loopback-address-of-vtep Example: Device (config) # ip pim vrf green register-source loopback901	Configures a unique IP address for the loopback interface of the VTEP that acts as the first hop router to multicast traffic.
Step 10	end Example: Device (config) # end	Returns to privileged EXEC mode.

Configuring TRM with PIM Source Specific Mode

To configure TRM with PIM Source Specific Mode, perform the following tasks:

- [Configuring the TRM Multicast Distribution Tree in the VRF, on page 250](#)
- [Configuring Multicast Routing on the Overlay VRF, on page 252](#)
- [Configuring Multicast on Switch Virtual Interfaces for the Core-facing and Access-facing VLANs, on page 252](#)
- [Configuring BGP with MVPN Address Family on VTEP, on page 253](#)
- [Configuring RP for Underlay Network , on page 254](#)
- [Configuring SSM for Overlay Network, on page 257](#)

Configuring SSM for Overlay Network

To configure SSM for the overlay network, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip pim [vrf vrf-name] ssm { default range access-list } Example: Device(config)# ip pim vrf green ssm default	Configures an SSM range for TRM. The default keyword defines the SSM range access list as 232/8. The range keyword specifies the standard IP access list number or name that defines the SSM range.
Step 4	end Example: Device(config)# end	Returns to privileged EXEC mode.

Verifying Tenant Routed Multicast

The following table lists the **show** commands that are used to verify TRM:

Command	Purpose
show nve peers	Displays NVE interface state information for peer leaf switches.

Command	Purpose
<code>show l2vpn evpn peers vxlan</code>	Displays Layer 2 EVPN peer route counts in the VXLAN and up time.
<code>show ip igmp vrf green groups</code>	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.
<code>show bgp ipv4 mvpn all</code>	Displays the MVPN options for BGP MVPN C-route signaling.
<code>show ip mroute vrf green</code>	Displays the contents of the mroute table that pertain to a specific MVRF instance.
<code>show ip mfib vrf green</code>	Displays forwarding entries and interfaces in the IPv4 Multicast Forwarding Information Base (MFIB) associated with MVRF instances.
<code>show ip mroute</code>	Displays multicast routing table information.
<code>show ip mfib</code>	Displays the forwarding entries and interfaces in the IPv4 MFIB.

Troubleshooting Tenant Routed Multicast

See [Troubleshoot EVPN VxLAN TRM on Catalyst 9000 Switches](#) document to learn how to troubleshoot issues with TRM in a BGP EVPN VXLAN fabric.

Configuration Examples for Tenant Routed Multicast

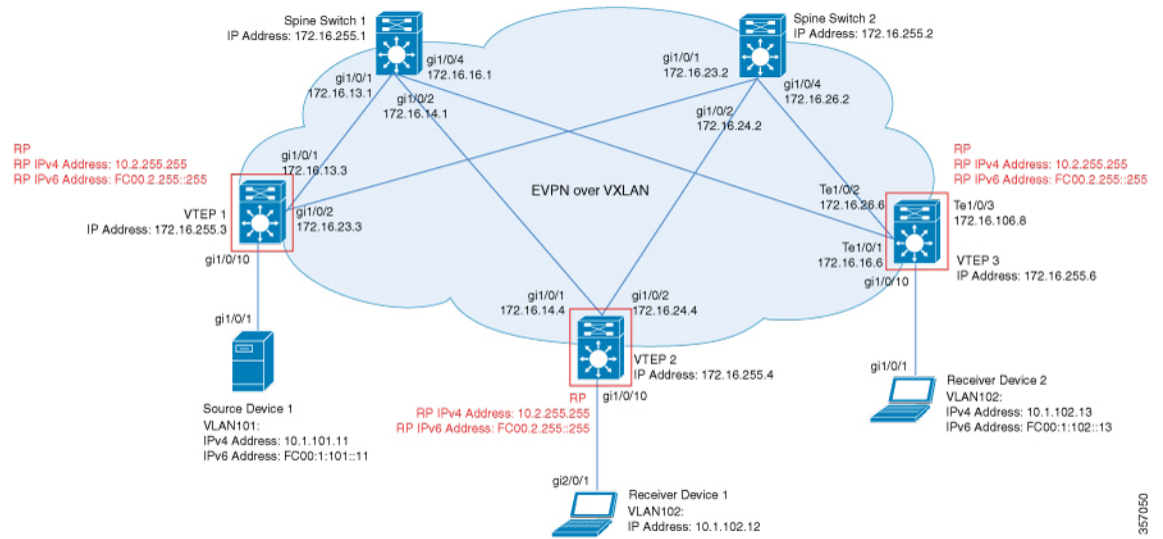
The following sections provide configuration examples for TRM in different scenarios.



Note These configuration examples do not have the Data MDT feature enabled.

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

This example shows how to configure and verify Layer 3 TRM with PIM-SM for IPv4 and IPv6 multicast traffic when the every VTEP is an RP is inside the BGP EVPN VXLAN fabric.



The topology shows an EVPN VXLAN network with a receiver device and a source device connected to VTEP 1 and VTEP 2 respectively. The IPv4 multicast group is 226.1.1.1 and the IPv6 multicast group is FF06:1::1 in this topology. The following tables provide sample configurations for the devices in this topology:

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

Table 29: Configuring VTEP 1, VTEP 2, and VTEP 3 to Configure TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic

VTEP 1	VTEP 2	VTEP 3
<pre> Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp spt-only 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp spt-only 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp spt-only 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp spt-only 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! </pre>	<pre> Leaf-03# show running-config hostname Leaf-03 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp spt-only 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp spt-only 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! </pre>

VTEP 1	VTEP 2	VTEP 3
<pre> interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip pim sparse-mode 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 Loopback255 vrf forwarding green ip address 10.2.255.255 255.255.255.255 ip pim sparse-mode ipv6 address FC00:2:255::255/128 ipv6 enable ! interface Loopback901 vrf forwarding green ip address 10.1.255.1 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::1/128 ipv6 enable ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access ! </pre>	<pre> 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 Loopback255 vrf forwarding green ip address 10.2.255.255 255.255.255.255 ip pim sparse-mode ipv6 address FC00:2:255::2/128 ipv6 enable ! interface Loopback901 vrf forwarding green ip address 10.1.255.2 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::2/128 ipv6 enable ! 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 GigabitEthernet1/0/2 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 access vlan 102 switchport mode access ! </pre>	<pre> 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 Loopback255 vrf forwarding green ip address 10.2.255.255 255.255.255.255 ip pim sparse-mode ipv6 address FC00:2:255::255/128 ipv6 enable ! interface Loopback901 vrf forwarding green ip address 10.1.255.3 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::3/128 ipv6 enable ! 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 ! interface TenGigabitEthernet1/0/2 no switchport ip address 172.16.26.6 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/10 switchport access vlan 102 switchport mode access ! </pre>

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

VTEP 1	VTEP 2	VTEP 3
<pre> interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode 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 mcast-group 225.0.0.102 ! 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 ! </pre>	<pre> interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode 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 mcast-group 225.0.0.102 ! 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 ! </pre>	<pre> interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode 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 mcast-group 225.0.0.102 ! router ospf 1 router-id 172.16.255.6 ! 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 ! </pre>

VTEP 1	VTEP 2	VTEP 3
<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 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 ! 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 vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end Leaf-01# </pre>	<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 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 ! 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 vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end Leaf-02# </pre>	<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 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 ! 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 vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end Leaf-03# </pre>

Table 30: Configuring Spine Switch 1 and Spine Switch 2 to Configure TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic

Spine Switch 1	Spine Switch 2
----------------	----------------

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! ipv6 unicast-routing ! system mtu 9198 ! 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 GigabitEthernet1/0/4 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 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 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! ipv6 unicast-routing ! system mtu 9198 ! 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 ! interface GigabitEthernet1/0/4 no switchport ip address 172.16.26.2 255.255.255.0 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

Spine Switch 1	Spine Switch 2
<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family ipv6 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family ipv6 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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>

Verifying TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic

The following sections provide sample outputs for **show** commands to verify TRM with PIM-SM on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 267](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 274](#)
- [Outputs to Verify the Configuration on VTEP 3, on page 281](#)
- [Outputs to Verify the Configuration on Spine Switch 1, on page 289](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 294](#)

Outputs to Verify the Configuration on VTEP 1

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

```
Leaf-01# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.6   0c75.bd67.ef48 50901      UP  A/-/4 01:47:43
nve1     50901    L3CP 172.16.254.4     7c21.0dbd.9548 50901      UP  A/-/4 01:47:43
nve1     50901    L3CP 172.16.254.6     0c75.bd67.ef48 50901      UP  A/M/6 01:47:43
nve1     50901    L3CP 172.16.254.4     7c21.0dbd.9548 50901      UP  A/M/6 01:47:43
nve1     10102    L2CP 172.16.254.4      7             10102      UP   N/A   01:47:43
nve1     10102    L2CP 172.16.254.6      7             10102      UP   N/A   01:47:43
Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1     10102    172.16.254.4    7         10102    01:47:43
nve1     10102    172.16.254.6    7         10102    01:47:43
```

```
Leaf-01#show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 8, main routing table version 8
3 network entries using 1176 bytes of memory
4 path entries using 640 bytes of memory
2/2 BGP path/bestpath attribute entries using 608 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
14 BGP extended community entries using 1848 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 4456 total bytes of memory
BGP activity 69/0 prefixes, 92/2 paths, scan interval 60 secs
3 networks peaked at 11:32:31 Sep 16 2020 UTC (01:42:43.716 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   140    127     8     0     0 01:48:48      1
172.16.255.2  4      65001   146    128     8     0     0 01:48:40      1
Leaf-01#
```

The following example shows the output for the **show bgp ipv6 mvpn all** command on VTEP 1:

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

Leaf-01# show bgp ipv6 mvpn all
BGP table version is 8, local router ID is 172.16.255.3
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: 1:1 (default for vrf green)
*> [5][1:1][FC00:1:101::11][FF06:1::1]/42
      ::
      32768 ?
*>i [7][1:1][65001][FC00:1:101::11][FF06:1::1]/46
      172.16.255.4          0      100      0 ?
Route Distinguisher: 172.16.254.3:101
* i [7][172.16.254.3:101][65001][FC00:1:101::11][FF06:1::1]/46
      172.16.255.4          0      100      0 ?
*>i          172.16.255.4          0      100      0 ?
Leaf-01#

```

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

```

Leaf-01# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 65, main routing table version 65
42 network entries using 16128 bytes of memory
60 path entries using 12720 bytes of memory
11/11 BGP path/bestpath attribute entries using 3168 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
14 BGP extended community entries using 1848 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 34048 total bytes of memory
BGP activity 69/0 prefixes, 92/2 paths, scan interval 60 secs
42 networks peaked at 11:27:30 Sep 16 2020 UTC (01:47:45.010 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   140    127     65    0    0 01:48:48    18
172.16.255.2  4      65001   146    128     65    0    0 01:48:40    18
Leaf-01#

```

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

```

Leaf-01# show bgp l2vpn evpn summary
BGP table version is 65, local router ID is 172.16.255.3
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: 172.16.254.3:101
*> [2][172.16.254.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
      ::
      32768 ?
*> [2][172.16.254.3:101][0][48][10B3D56A8FC1][128][FC00:1:101::1]/36

```



```

::
32768 ?
*> [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [0] [*]/20
::
32768 ?
*> [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [32] [10.1.101.11]/24
::
32768 ?
*> [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FC00:1:101::11]/36
::
32768 ?
*> [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FE80::F6CF:E2FF:FE43:34C1]/36
::
32768 ?
Route Distinguisher: 172.16.254.3:102
*>i [2] [172.16.254.3:102] [0] [48] [0C75BD67EF4D] [32] [10.1.102.1]/24
172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [0C75BD67EF4D] [128] [FC00:1:102::1]/36
172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [44D3CA286CC5] [0] [*]/20
172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [44D3CA286CC5] [32] [10.1.102.12]/24
172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [44D3CA286CC5] [128] [FC00:1:102::12]/36
172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [44D3CA286CC5] [128] [FE80::46D3:CAFF:FE28:6CC5]/36
172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [7C210DBD954D] [32] [10.1.102.1]/24
172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [7C210DBD954D] [128] [FC00:1:102::1]/36
172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [ECE1A93792C5] [0] [*]/20
172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [ECE1A93792C5] [32] [10.1.102.13]/24
172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [ECE1A93792C5] [128] [FC00:1:102::13]/36
172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.3:102] [0] [48] [ECE1A93792C5] [128] [FE80::EEE1:A9FF:FE37:92C5]/36
172.16.254.6 0 100 0 ?
Route Distinguisher: 172.16.254.4:102
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [0] [*]/20
172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [32] [10.1.102.12]/24
172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [128] [FC00:1:102::12]/36
172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [128] [FE80::46D3:CAFF:FE28:6CC5]/36
172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [7C210DBD954D] [32] [10.1.102.1]/24
172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [7C210DBD954D] [128] [FC00:1:102::1]/36
172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
Route Distinguisher: 172.16.254.6:102
*>i [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [32] [10.1.102.1]/24
172.16.254.6 0 100 0 ?
* i 172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [128] [FC00:1:102::1]/36
172.16.254.6 0 100 0 ?
* i 172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [0] [*]/20
172.16.254.6 0 100 0 ?
* i 172.16.254.6 0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [32] [10.1.102.13]/24

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

172.16.254.6          0 100 0 ?
* i                  172.16.254.6          0 100 0 ?
*>i [2][172.16.254.6:102][0][48][ECE1A93792C5][128][FC00:1:102::13]/36
172.16.254.6          0 100 0 ?
* i                  172.16.254.6          0 100 0 ?
*>i [2][172.16.254.6:102][0][48][ECE1A93792C5][128][FE80::EEE1:A9FF:FE37:92C5]/36
172.16.254.6          0 100 0 ?
* i                  172.16.254.6          0 100 0 ?
Route Distinguisher: 1:1 (default for vrf green)
*> [5][1:1][0][24][10.1.101.0]/17
0.0.0.0                0          32768 ?
*>i [5][1:1][0][24][10.1.102.0]/17
172.16.254.4          0 100 0 ?
* i                  172.16.254.4          0 100 0 ?
*> [5][1:1][0][32][10.1.255.1]/17
0.0.0.0                0          32768 ?
*>i [5][1:1][0][32][10.1.255.2]/17
172.16.254.4          0 100 0 ?
* i                  172.16.254.4          0 100 0 ?
*>i [5][1:1][0][32][10.1.255.3]/17
172.16.254.6          0 100 0 ?
* i                  172.16.254.6          0 100 0 ?
*> [5][1:1][0][32][10.2.255.255]/17
0.0.0.0                0          32768 ?
*> [5][1:1][0][64][FC00:1:101::]/29
::                      0          32768 ?
*>i [5][1:1][0][64][FC00:1:102::]/29
172.16.254.4          0 100 0 ?
* i                  172.16.254.4          0 100 0 ?
*> [5][1:1][0][128][FC00:1:255::1]/29
::                      0          32768 ?
*>i [5][1:1][0][128][FC00:1:255::2]/29
172.16.254.4          0 100 0 ?
* i                  172.16.254.4          0 100 0 ?
*>i [5][1:1][0][128][FC00:1:255::3]/29
172.16.254.6          0 100 0 ?
* i                  172.16.254.6          0 100 0 ?
*> [5][1:1][0][128][FC00:2:255::255]/29
::                      0          32768 ?
Leaf-01#

```

The following example shows the output for the **show ipv6 pim vrf vrf-name group-map** command on VTEP 1:

```

Leaf-01# show ipv6 pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: Tu2,FC00:2:255::255 (us)
  Info source: Static
  Uptime: 01:49:05, Groups: 1
Leaf-01#

```

The following example shows the output for the **show ipv6 route vrf** command on VTEP 1:

```

Leaf-01# show ipv6 route vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "connected", distance 0, metric 0, type receive, connected
  Redistributing via bgp 65001

```

```

Route count is 1/1, share count 0
Routing paths:
  receive via Loopback255
  Last updated 01:49:06 ago
Leaf-01#

```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on VTEP 1:

```

Leaf-01# show ipv6 mld vrf green groups
No groups found.
Leaf-01#

```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on VTEP 1:

```

Leaf-01# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(FC00:1:101::11, FF06:1::1), 01:42:44/00:03:19, flags: SFTGq
  Incoming interface: Vlan101
  RPF nbr: FE80::F6CF:E2FF:FE43:34C1
  Immediate Outgoing interface list:
    Vlan901, Forward, 01:42:44/never
Leaf-01#

```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on VTEP 1:

```

Leaf-01# show ipv6 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,
            e - Encap helper tunnel flag.
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
(FC00:1:101::11,FF06:1::1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 1/0/1
  HW Forwarding: 3161/0/118/0, Other: 0/0/0

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

Vlan101 Flags: A
Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
Pkts: 0/0/0 Rate: 0 pps
Leaf-01#

```

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

```

Leaf-01# 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,
e - encap-helper tunnel flag
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), 01:48:56/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, 01:48:56/00:02:56

(172.16.254.3, 239.1.1.1), 01:42:42/00:03:20, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
GigabitEthernet1/0/2, Forward/Sparse, 01:42:42/00:03:03

(*, 224.0.1.40), 01:49:06/00:02:55, RP 172.16.255.255, flags: SJCL
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
Loopback0, Forward/Sparse, 01:49:05/00:02:55

(*, 225.0.0.102), 01:48:56/stopped, RP 172.16.255.255, flags: SJCx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
Tunnel0, Forward/Sparse, 01:48:56/00:02:56

(172.16.254.4, 225.0.0.102), 01:48:17/00:01:19, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
Tunnel0, Forward/Sparse, 01:48:17/00:02:56

(172.16.254.6, 225.0.0.102), 01:48:23/00:01:18, flags: Tx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
Tunnel0, Forward/Sparse, 01:48:23/00:02:56

(*, 225.0.0.101), 01:49:01/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, 01:48:56/00:02:56

(172.16.254.3, 225.0.0.101), 01:49:01/00:02:45, flags: FTx

```

```

Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 01:48:25/00:03:12
Leaf-01#

```

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

```

Leaf-01# 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,
                  e - Encap helper tunnel flag.
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
GigabitEthernet1/0/2 Flags: A NS
Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 2/0/125/0, Other: 1/0/1
  HW Forwarding: 554/0/163/0, Other: 0/0/0
Null0 Flags: A
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/172/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 561/0/176/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 1/1/0
  HW Forwarding: 504/0/205/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

    Pkts: 0/0/0    Rate: 0 pps
  (*,232.0.0.0/8) Flags: HW
    SW Forwarding: 0/0/0/0, Other: 0/0/0
    HW Forwarding:  0/0/0/0, Other: 0/0/0
  (*,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
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0    Rate: 0 pps
  (172.16.254.3,239.1.1.1) Flags: HW
    SW Forwarding: 1/0/150/0, Other: 1/1/0
    HW Forwarding:  3071/0/156/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0    Rate: 0 pps
Leaf-01#

```

Return to [Verifying TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic](#), on page 267

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.6  0c75.bd67.ef48 50901      UP  A/-/4 01:52:57
nve1     50901    L3CP 172.16.254.3  10b3.d56a.8fc8 50901      UP  A/-/4 01:52:57
nve1     50901    L3CP 172.16.254.6  0c75.bd67.ef48 50901      UP  A/M/6 01:52:57
nve1     50901    L3CP 172.16.254.3  10b3.d56a.8fc8 50901      UP  A/M/6 01:52:57
nve1     10101    L2CP 172.16.254.3    7              10101      UP  N/A   01:52:57
nve1     10102    L2CP 172.16.254.6    7              10102      UP  N/A   01:52:57
Leaf-02#

```

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

```

Leaf-02# show l2vpn evpn peers vxlan

Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1     10101    172.16.254.3    7          10101    01:52:57
nve1     10102    172.16.254.6    7          10102    01:52:57
Leaf-02#

```

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on VTEP 2:

```

Leaf-02# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 5, main routing table version 5
2 network entries using 784 bytes of memory
3 path entries using 480 bytes of memory
2/2 BGP path/bestpath attribute entries using 608 bytes of memory
4 BGP rinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
14 BGP extended community entries using 1848 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 3904 total bytes of memory
BGP activity 70/0 prefixes, 101/6 paths, scan interval 60 secs
2 networks peaked at 11:37:07 Sep 16 2020 UTC (01:47:58.150 ago)

Neighbor      V          AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4          65001   150    133      5     0     0 01:53:34      1
172.16.255.2  4          65001   151    134      5     0     0 01:53:30      1
Leaf-02#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on VTEP 2:

```

Leaf-02# show bgp ipv6 mvpn all
BGP table version is 5, local router ID is 172.16.255.4
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: 1:1 (default for vrf green)
* i   [5][1:1][FC00:1:101::11][FF06:1::1]/42
              172.16.255.3          0    100    0 ?
*>i     172.16.255.3          0    100    0 ?
Route Distinguisher: 172.16.254.3:101
*>    [7][172.16.254.3:101][65001][FC00:1:101::11][FF06:1::1]/46
              ::                      32768 ?
Leaf-02#

```

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

```

Leaf-02# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 43, main routing table version 43
42 network entries using 16128 bytes of memory
64 path entries using 13568 bytes of memory
12/12 BGP path/bestpath attribute entries using 3456 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
14 BGP extended community entries using 1848 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 35184 total bytes of memory
BGP activity 70/0 prefixes, 101/6 paths, scan interval 60 secs
42 networks peaked at 11:32:07 Sep 16 2020 UTC (01:52:58.436 ago)

Neighbor      V          AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4          65001   150    133      43    0     0 01:53:35      20
172.16.255.2  4          65001   151    134      43    0     0 01:53:31      20
Leaf-02#

```

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

```

Leaf-02# show bgp l2vpn evpn
BGP table version is 43, local router ID is 172.16.255.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

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: 172.16.254.3:101
*>i [2][172.16.254.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
    172.16.254.3          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
*>i [2][172.16.254.3:101][0][48][10B3D56A8FC1][128][FC00:1:101::1]/36
    172.16.254.3          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
*>i [2][172.16.254.3:101][0][48][F4CFE24334C1][0][*]/20
    172.16.254.3          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
*>i [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24
    172.16.254.3          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
*>i [2][172.16.254.3:101][0][48][F4CFE24334C1][128][FC00:1:101::11]/36
    172.16.254.3          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
*>i [2][172.16.254.3:101][0][48][F4CFE24334C1][128][FE80::F6CF:E2FF:FE43:34C1]/36
    172.16.254.3          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:101
*>i [2][172.16.254.4:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
    172.16.254.3          0    100    0 ?
*>i [2][172.16.254.4:101][0][48][10B3D56A8FC1][128][FC00:1:101::1]/36
    172.16.254.3          0    100    0 ?
*>i [2][172.16.254.4:101][0][48][F4CFE24334C1][0][*]/20
    172.16.254.3          0    100    0 ?
*>i [2][172.16.254.4:101][0][48][F4CFE24334C1][32][10.1.101.11]/24
    172.16.254.3          0    100    0 ?
*>i [2][172.16.254.4:101][0][48][F4CFE24334C1][128][FC00:1:101::11]/36
    172.16.254.3          0    100    0 ?
*>i [2][172.16.254.4:101][0][48][F4CFE24334C1][128][FE80::F6CF:E2FF:FE43:34C1]/36
    172.16.254.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:102
*>i [2][172.16.254.4:102][0][48][0C75BD67EF4D][32][10.1.102.1]/24
    172.16.254.6          0    100    0 ?
*>i [2][172.16.254.4:102][0][48][0C75BD67EF4D][128][FC00:1:102::1]/36
    172.16.254.6          0    100    0 ?
*> [2][172.16.254.4:102][0][48][44D3CA286CC5][0][*]/20
    ::                      32768 ?
*> [2][172.16.254.4:102][0][48][44D3CA286CC5][32][10.1.102.12]/24
    ::                      32768 ?
*> [2][172.16.254.4:102][0][48][44D3CA286CC5][128][FC00:1:102::12]/36
    ::                      32768 ?
*> [2][172.16.254.4:102][0][48][44D3CA286CC5][128][FE80::46D3:CAFF:FE28:6CC5]/36
    ::                      32768 ?
*> [2][172.16.254.4:102][0][48][7C210DBD954D][32][10.1.102.1]/24
    ::                      32768 ?
*> [2][172.16.254.4:102][0][48][7C210DBD954D][128][FC00:1:102::1]/36
    ::                      32768 ?
*>i [2][172.16.254.4:102][0][48][ECE1A93792C5][0][*]/20
    172.16.254.6          0    100    0 ?
*>i [2][172.16.254.4:102][0][48][ECE1A93792C5][32][10.1.102.13]/24
    172.16.254.6          0    100    0 ?
*>i [2][172.16.254.4:102][0][48][ECE1A93792C5][128][FC00:1:102::13]/36
    172.16.254.6          0    100    0 ?
*>i [2][172.16.254.4:102][0][48][ECE1A93792C5][128][FE80::EEE1:A9FF:FE37:92C5]/36
    172.16.254.6          0    100    0 ?

```



```

Route Distinguisher: 172.16.254.6:102
*>i [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [32] [10.1.102.1]/24
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
*>i [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [128] [FC00:1:102::1]/36
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
*>i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [0] [*]/20
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
*>i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [32] [10.1.102.13]/24
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
*>i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [128] [FC00:1:102::13]/36
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
*>i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [128] [FE80::EEE1:A9FF:FE37:92C5]/36
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
Route Distinguisher: 1:1 (default for vrf green)
*>i [5] [1:1] [0] [24] [10.1.101.0]/17
      172.16.254.3          0      100      0 ?
* i          172.16.254.3          0      100      0 ?
*> [5] [1:1] [0] [24] [10.1.102.0]/17
      0.0.0.0                0                32768 ?
*>i [5] [1:1] [0] [32] [10.1.255.1]/17
      172.16.254.3          0      100      0 ?
* i          172.16.254.3          0      100      0 ?
*> [5] [1:1] [0] [32] [10.1.255.2]/17
      0.0.0.0                0                32768 ?
*>i [5] [1:1] [0] [32] [10.1.255.3]/17
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
* i [5] [1:1] [0] [32] [10.2.255.255]/17
      172.16.254.3          0      100      0 ?
* i          172.16.254.3          0      100      0 ?
*> [5] [1:1] [0] [32] [10.2.255.255]/17
      0.0.0.0                0                32768 ?
*>i [5] [1:1] [0] [64] [FC00:1:101::]/29
      172.16.254.3          0      100      0 ?
* i          172.16.254.3          0      100      0 ?
*> [5] [1:1] [0] [64] [FC00:1:102::]/29
      ::                    0                32768 ?
*>i [5] [1:1] [0] [128] [FC00:1:255::1]/29
      172.16.254.3          0      100      0 ?
* i          172.16.254.3          0      100      0 ?
*> [5] [1:1] [0] [128] [FC00:1:255::2]/29
      ::                    0                32768 ?
*>i [5] [1:1] [0] [128] [FC00:1:255::3]/29
      172.16.254.6          0      100      0 ?
* i          172.16.254.6          0      100      0 ?
* i [5] [1:1] [0] [128] [FC00:2:255::255]/29
      172.16.254.3          0      100      0 ?
* i          172.16.254.3          0      100      0 ?
*> [5] [1:1] [0] [128] [FC00:2:255::255]/29
      ::                    0                32768 ?
Leaf-02#

```

The following example shows the output for the **show ipv6 pim vrf vrf-name group-map** command on VTEP 2:

```

Leaf-02# show ipv6 pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

```

```

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: Tu2,FC00:2:255::255 (us)
  Info source: Static
  Uptime: 01:54:21, Groups: 1
Leaf-02#

```

The following example shows the output for the **show ipv6 route vrf** command on VTEP 2:

```

Leaf-02# show ipv6 route vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "connected", distance 0, metric 0, type receive, connected
  Redistributing via bgp 65001
  Route count is 1/1, share count 0
  Routing paths:
    receive via Loopback255
      Last updated 01:54:21 ago
Leaf-02#

```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on VTEP 2:

```

Leaf-02# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
  Uptime    Expires
FF06:1::1                                     Vlan102
  01:53:45  00:03:52
Leaf-02#

```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on VTEP 2:

```

Leaf-02# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 01:53:45/never, RP FC00:2:255::255, flags: SCJ
  Incoming interface: Tunnel2
  RPF nbr: FC00:2:255::255
  Immediate Outgoing interface list:
    Vlan102, Forward, 01:53:45/never

(FC00:1:101::11, FF06:1::1), 01:47:58/never, flags: STgQ
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.3
  Inherited Outgoing interface list:
    Vlan102, Forward, 01:53:45/never
Leaf-02#

```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on VTEP 2:

```
Leaf-02# show ipv6 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,
                  e - Encap helper tunnel flag.
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
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel2 Flags: A NS
  Vlan102 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(FC00:1:101::11,FF06:1::1) Flags: HW
  SW Forwarding: 1/0/100/0, Other: 0/0/0
  HW Forwarding: 3225/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan102 Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
Leaf-02#
```

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

```
Leaf-02# 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,
        e - encap-helper tunnel flag
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), 01:54:12/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 01:54:12/00:00:41
```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```
(172.16.254.3, 239.1.1.1), 01:47:56/00:02:39, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 01:47:56/00:00:03

(*, 224.0.1.40), 01:54:21/00:02:39, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 01:54:20/00:02:39

(*, 225.0.0.102), 01:54:12/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, 01:54:12/00:00:41

(172.16.254.6, 225.0.0.102), 01:53:36/00:00:58, flags: Tx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 01:53:36/00:00:41

(172.16.254.4, 225.0.0.102), 01:53:47/00:02:53, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 01:53:36/00:02:40, A

(*, 225.0.0.101), 01:54:12/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 01:54:12/00:00:41

(172.16.254.3, 225.0.0.101), 01:53:11/00:01:39, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 01:53:11/00:00:48
Leaf-02#
```

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

```
Leaf-02# 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,
               e - Encap helper tunnel flag.
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
```

```

GigabitEthernet1/0/2 Flags: A NS
Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 1/0/190/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
SW Forwarding: 1/0/172/0, Other: 0/0/0
HW Forwarding: 529/0/177/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
SW Forwarding: 2/0/163/0, Other: 3/1/2
HW Forwarding: 631/0/163/0, Other: 0/0/0
Null0 Flags: A
GigabitEthernet1/0/2 Flags: F
  Pkts: 0/0/2   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 530/0/205/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 1/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 0/0/0
HW Forwarding: 3224/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
Leaf-02#

```

Return to [Verifying TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic, on page 267](#)

Outputs to Verify the Configuration on VTEP 3

The following example shows the output for the **show nve peers** command on VTEP 3:

```

Leaf-03# show nve peers
Interface  VNI      Type Peer-IP      RMAC/Num_RTs  eVNI      state flags UP time
nve1      50901    L3CP 172.16.254.3  10b3.d56a.8fc8 50901      UP  A/-/4 02:01:22
nve1      50901    L3CP 172.16.254.4  7c21.0dbd.9548 50901      UP  A/-/4 02:01:22
nve1      50901    L3CP 172.16.254.3  10b3.d56a.8fc8 50901      UP  A/M/6 02:01:22
nve1      50901    L3CP 172.16.254.4  7c21.0dbd.9548 50901      UP  A/M/6 02:01:22
nve1      10101    L2CP 172.16.254.3  7          10101      UP  N/A   02:01:22

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```
nve1      10102      L2CP 172.16.254.4      7              10102      UP      N/A      02:01:22
Leaf-03#
```

The following example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 3:

```
Leaf-03# show l2vpn evpn peers vxlan
```

Interface	VNI	Peer-IP	Num routes	eVNI	UP time
nve1	10101	172.16.254.3	7	10101	02:01:23
nve1	10102	172.16.254.4	7	10102	02:01:23

```
Leaf-03#
```

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on VTEP 3:

```
Leaf-03# show bgp ipv6 mvpn all summary
```

```
BGP router identifier 172.16.255.6, local AS number 65001
BGP table version is 5, main routing table version 5
2 network entries using 784 bytes of memory
3 path entries using 480 bytes of memory
2/2 BGP path/bestpath attribute entries using 608 bytes of memory
4 BGP rinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
14 BGP extended community entries using 1848 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 3904 total bytes of memory
BGP activity 66/0 prefixes, 97/0 paths, scan interval 60 secs
2 networks peaked at 11:29:08 Sep 16 2020 UTC (01:56:22.908 ago)
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	160	143	5	0	0	02:01:59	1
172.16.255.2	4	65001	159	142	5	0	0	02:01:59	1

```
Leaf-03#
```

The following example shows the output for the **show bgp ipv6 mvpn all** command on VTEP 3:

```
Leaf-03# show bgp ipv6 mvpn all
```

```
BGP table version is 5, local router ID is 172.16.255.6
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: 1:1 (default for vrf green)					
* i [5] [1:1] [FC00:1:101::11] [FF06:1::1] /42	172.16.255.3	0	100	0	?
*>i	172.16.255.3	0	100	0	?
Route Distinguisher: 172.16.254.3:101					
*> [7] [172.16.254.3:101] [65001] [FC00:1:101::11] [FF06:1::1] /46	::			32768	?

```
Leaf-03#
```

The following example shows the output for the **show bgp l2vpn evpn summary** command on VTEP 3:

```
Leaf-03# show bgp l2vpn evpn summary
BGP router identifier 172.16.255.6, local AS number 65001
BGP table version is 51, main routing table version 51
42 network entries using 16128 bytes of memory
68 path entries using 14416 bytes of memory
12/12 BGP path/bestpath attribute entries using 3456 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
14 BGP extended community entries using 1848 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 36032 total bytes of memory
BGP activity 66/0 prefixes, 97/0 paths, scan interval 60 secs
42 networks peaked at 11:24:07 Sep 16 2020 UTC (02:01:24.200 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   160    143     51    0    0 02:02:00    22
172.16.255.2  4      65001   159    142     51    0    0 02:01:59    22
Leaf-03#
```

The following example shows the output for the **show bgp l2vpn evpn** command on VTEP 3:

```
Leaf-03# show bgp l2vpn evpn
BGP table version is 51, local router ID is 172.16.255.6
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: 172.16.254.3:101
*>i [2] [172.16.254.3:101] [0] [48] [10B3D56A8FC1] [32] [10.1.101.1]/24
      172.16.254.3      0      100      0 ?
* i      172.16.254.3      0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [10B3D56A8FC1] [128] [FC00:1:101::1]/36
      172.16.254.3      0      100      0 ?
* i      172.16.254.3      0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [0] [*]/20
      172.16.254.3      0      100      0 ?
* i      172.16.254.3      0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [32] [10.1.101.11]/24
      172.16.254.3      0      100      0 ?
* i      172.16.254.3      0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FC00:1:101::11]/36
      172.16.254.3      0      100      0 ?
* i      172.16.254.3      0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FE80::F6CF:E2FF:FE43:34C1]/36
      172.16.254.3      0      100      0 ?
* i      172.16.254.3      0      100      0 ?
Route Distinguisher: 172.16.254.4:102
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [0] [*]/20
      172.16.254.4      0      100      0 ?
* i      172.16.254.4      0      100      0 ?
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [32] [10.1.102.12]/24
      172.16.254.4      0      100      0 ?
* i      172.16.254.4      0      100      0 ?
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [128] [FC00:1:102::12]/36
```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [128] [FE80::46D3:CAFF:FE28:6CC5] /36
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [7C210DBD954D] [32] [10.1.102.1] /24
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.4:102] [0] [48] [7C210DBD954D] [128] [FC00:1:102::1] /36
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.254.6:101
*>i [2] [172.16.254.6:101] [0] [48] [10B3D56A8FC1] [32] [10.1.101.1] /24
172.16.254.3          0 100 0 ?
*>i [2] [172.16.254.6:101] [0] [48] [10B3D56A8FC1] [128] [FC00:1:101::1] /36
172.16.254.3          0 100 0 ?
*>i [2] [172.16.254.6:101] [0] [48] [F4CFE24334C1] [0] [*] /20
172.16.254.3          0 100 0 ?
*>i [2] [172.16.254.6:101] [0] [48] [F4CFE24334C1] [32] [10.1.101.11] /24
172.16.254.3          0 100 0 ?
*>i [2] [172.16.254.6:101] [0] [48] [F4CFE24334C1] [128] [FC00:1:101::11] /36
172.16.254.3          0 100 0 ?
*>i [2] [172.16.254.6:101] [0] [48] [F4CFE24334C1] [128] [FE80::F6CF:E2FF:FE43:34C1] /36
172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.254.6:102
*> [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [32] [10.1.102.1] /24
::                      32768 ?
*> [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [128] [FC00:1:102::1] /36
::                      32768 ?
*>i [2] [172.16.254.6:102] [0] [48] [44D3CA286CC5] [0] [*] /20
172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [44D3CA286CC5] [32] [10.1.102.12] /24
172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [44D3CA286CC5] [128] [FC00:1:102::12] /36
172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [44D3CA286CC5] [128] [FE80::46D3:CAFF:FE28:6CC5] /36
172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [7C210DBD954D] [32] [10.1.102.1] /24
172.16.254.4          0 100 0 ?
*>i [2] [172.16.254.6:102] [0] [48] [7C210DBD954D] [128] [FC00:1:102::1] /36
172.16.254.4          0 100 0 ?
*> [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [0] [*] /20
::                      32768 ?
*> [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [32] [10.1.102.13] /24
::                      32768 ?
*> [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [128] [FC00:1:102::13] /36
::                      32768 ?
*> [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [128] [FE80::EEE1:A9FF:FE37:92C5] /36
::                      32768 ?
Route Distinguisher: 1:1 (default for vrf green)
*>i [5] [1:1] [0] [24] [10.1.101.0] /17
172.16.254.3          0 100 0 ?
* i                   172.16.254.3          0 100 0 ?
* i [5] [1:1] [0] [24] [10.1.102.0] /17
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
*> 0.0.0.0             0 32768 ?
*>i [5] [1:1] [0] [32] [10.1.255.1] /17
172.16.254.3          0 100 0 ?
* i                   172.16.254.3          0 100 0 ?
*>i [5] [1:1] [0] [32] [10.1.255.2] /17
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
*> [5] [1:1] [0] [32] [10.1.255.3] /17

```



```

0.0.0.0          0          32768 ?
* i  [5] [1:1] [0] [32] [10.2.255.255]/17
      172.16.254.3      0 100 0 ?
* i      172.16.254.3      0 100 0 ?
*>      0.0.0.0          0          32768 ?
*>i  [5] [1:1] [0] [64] [FC00:1:101::]/29
      172.16.254.3      0 100 0 ?
* i      172.16.254.3      0 100 0 ?
* i  [5] [1:1] [0] [64] [FC00:1:102::]/29
      172.16.254.4      0 100 0 ?
* i      172.16.254.4      0 100 0 ?
*>      ::              0          32768 ?
*>i  [5] [1:1] [0] [128] [FC00:1:255::1]/29
      172.16.254.3      0 100 0 ?
* i      172.16.254.3      0 100 0 ?
*>i  [5] [1:1] [0] [128] [FC00:1:255::2]/29
      172.16.254.4      0 100 0 ?
* i      172.16.254.4      0 100 0 ?
*>  [5] [1:1] [0] [128] [FC00:1:255::3]/29
      ::              0          32768 ?
* i  [5] [1:1] [0] [128] [FC00:2:255::255]/29
      172.16.254.3      0 100 0 ?
* i      172.16.254.3      0 100 0 ?
*>      ::              0          32768 ?
Leaf-03#

```

The following example shows the output for the **show ipv6 pim vrf vrf-name group-map** command on VTEP 3:

```

Leaf-03# show ipv6 pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: Tu2,FC00:2:255::255 (us)
  Info source: Static
  Uptime: 02:02:54, Groups: 1
Leaf-03#

```

The following example shows the output for the **show ipv6 route vrf** command on VTEP 3:

```

Leaf-03# show ipv6 route vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "connected", distance 0, metric 0, type receive, connected
  Redistributing via bgp 65001
  Route count is 1/1, share count 0
  Routing paths:
    receive via Loopback255
    Last updated 02:02:55 ago
Leaf-03#

```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on VTEP 3:

```

Leaf-03# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
  Uptime    Expires
FF06:1::1                                     Vlan102

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```
02:02:06 00:03:50
Leaf-03#
```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on VTEP 3:

```
Leaf-03# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 02:02:06/never, RP FC00:2:255::255, flags: SCJ
  Incoming interface: Tunnel2
  RPF nbr: FC00:2:255::255
  Immediate Outgoing interface list:
    Vlan102, Forward, 02:02:06/never

(FC00:1:101::11, FF06:1::1), 01:56:23/never, flags: STgQ
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.3
  Inherited Outgoing interface list:
    Vlan102, Forward, 02:02:06/never
Leaf-03#
```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on VTEP 3:

```
Leaf-03# show ipv6 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,
            e - Encap helper tunnel flag.
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
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel2 Flags: A NS
  Vlan102 Flags: F NS
    Pkts: 0/0/0    Rate: 0 pps
(FC00:1:101::11,FF06:1::1) Flags: HW
  SW Forwarding: 1/0/100/0, Other: 0/0/0
```

```

HW Forwarding: 3475/0/126/0, Other: 0/0/0
Vlan901, VXLAN Decap Flags: A
Vlan102 Flags: F NS
Pkts: 0/0/1 Rate: 0 pps
Leaf-03#

```

The following example shows the output for the **show ip mroute** command on VTEP 3:

```

Leaf-03# 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,
       e - encap-helper tunnel flag
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), 02:02:45/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 02:02:45/00:01:07

(172.16.254.3, 239.1.1.1), 01:56:21/00:02:07, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 01:56:21/00:00:37

(*, 224.0.1.40), 02:02:55/00:02:10, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback0, Forward/Sparse, 02:02:54/00:02:10

(*, 225.0.0.102), 02:02:45/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 02:02:45/00:01:07

(172.16.254.4, 225.0.0.102), 02:01:56/00:02:46, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 02:01:56/00:01:07

(172.16.254.6, 225.0.0.102), 02:02:08/00:02:37, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/2, Forward/Sparse, 02:01:58/00:02:59

(*, 225.0.0.101), 02:02:45/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 02:02:45/00:01:07

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```
(172.16.254.3, 225.0.0.101), 02:01:36/00:01:06, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 02:01:36/00:01:23
Leaf-03#
```

The following example shows the output for the **show ip mfib** command on VTEP 3:

```
Leaf-03# 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,
             e - Encap helper tunnel flag.
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: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel0, VXLAN Decap Flags: NS
(*,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
  Loopback0 Flags: F IC NS
  Pkts: 0/0/0 Rate: 0 pps
(*,225.0.0.0/8) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/190/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0 Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/172/0, Other: 0/0/0
  HW Forwarding: 568/0/177/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1 Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/172/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0 Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 632/0/176/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
```

```

      Pkts: 0/0/1      Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 2/0/180/0, Other: 3/0/3
HW Forwarding: 610/0/189/0, Other: 0/0/0
Null0 Flags: A
TenGigabitEthernet1/0/2 Flags: F NS
      Pkts: 0/0/1      Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 1/0/168/0, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
      Pkts: 0/0/0      Rate: 0 pps
(172.16.254.3,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 0/0/0
HW Forwarding: 3474/0/168/0, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
      Pkts: 0/0/1      Rate: 0 pps
Leaf-03#

```

Return to [Verifying TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic, on page 267](#)

Outputs to Verify the Configuration on Spine Switch 1

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Spine Switch 1:

```

Spine-01# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 20, main routing table version 20
2 network entries using 784 bytes of memory
5 path entries using 800 bytes of memory
2/2 BGP path/bestpath attribute entries using 608 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
13 BGP extended community entries using 1808 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 4144 total bytes of memory
BGP activity 1001/969 prefixes, 7359/7288 paths, scan interval 60 secs
2 networks peaked at 11:16:15 Sep 16 2020 UTC (02:20:36.059 ago)

Neighbor          V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2     4      65001   161    167     20    0    0 02:07:55      2
172.16.255.3     4      65001   148    161     20    0    0 02:08:00      1
172.16.255.4     4      65001   148    165     20    0    0 02:07:32      1
172.16.255.6     4      65001   149    166     20    0    0 02:07:32      1
Spine-01#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Spine Switch 1:

```

Spine-01# show bgp ipv6 mvpn all
BGP table version is 20, local router ID is 172.16.255.1
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,

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

        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: 1:1
 * i [5][1:1][FC00:1:101::11][FF06:1::1]/42
      172.16.255.3      0 100 0 ?
 *>i      172.16.255.3      0 100 0 ?
Route Distinguisher: 172.16.254.3:101
 * i [7][172.16.254.3:101][65001][FC00:1:101::11][FF06:1::1]/46
      172.16.255.4      0 100 0 ?
 *>i      172.16.255.4      0 100 0 ?
 * i      172.16.255.6      0 100 0 ?
Spine-01#

```

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 785, main routing table version 785
30 network entries using 10320 bytes of memory
66 path entries using 13728 bytes of memory
11/11 BGP path/bestpath attribute entries using 3168 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
13 BGP extended community entries using 1808 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 29168 total bytes of memory
BGP activity 1001/969 prefixes, 7359/7288 paths, scan interval 60 secs
44 networks peaked at 10:13:07 Aug 6 2020 UTC (5w6d ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.2	4	65001	161	167	785	0	0	02:07:55	30
172.16.255.3	4	65001	148	161	785	0	0	02:08:00	12
172.16.255.4	4	65001	148	165	785	0	0	02:07:33	12
172.16.255.6	4	65001	149	166	785	0	0	02:07:33	12

Spine-01#

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

```

Spine-01# show bgp l2vpn evpn
BGP table version is 785, local router ID is 172.16.255.1
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: 172.16.254.3:101
 * i [2][172.16.254.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
      172.16.254.3      0 100 0 ?
 *>i      172.16.254.3      0 100 0 ?
 * i [2][172.16.254.3:101][0][48][10B3D56A8FC1][128][FC00:1:101::1]/36
      172.16.254.3      0 100 0 ?
 *>i      172.16.254.3      0 100 0 ?

```

```

* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [0] [*]/20
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [32] [10.1.101.11]/24
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FC00:1:101::11]/36
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FE80::F6CF:E2FF:FE43:34C1]/36
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
Route Distinguisher: 172.16.254.4:102
* i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [0] [*]/20
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?
* i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [32] [10.1.102.12]/24
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?
* i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [128] [FC00:1:102::12]/36
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?
* i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [128] [FE80::46D3:CAFF:FE28:6CC5]/36
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?
* i [2] [172.16.254.4:102] [0] [48] [7C210DBD954D] [32] [10.1.102.1]/24
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?
* i [2] [172.16.254.4:102] [0] [48] [7C210DBD954D] [128] [FC00:1:102::1]/36
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?
Route Distinguisher: 172.16.254.6:102
* i [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [32] [10.1.102.1]/24
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
* i [2] [172.16.254.6:102] [0] [48] [0C75BD67EF4D] [128] [FC00:1:102::1]/36
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
* i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [0] [*]/20
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
* i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [32] [10.1.102.13]/24
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
* i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [128] [FC00:1:102::13]/36
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
* i [2] [172.16.254.6:102] [0] [48] [ECE1A93792C5] [128] [FE80::EEE1:A9FF:FE37:92C5]/36
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
Route Distinguisher: 1:1
* i [5] [1:1] [0] [24] [10.1.101.0]/17
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
*>i [5] [1:1] [0] [24] [10.1.102.0]/17
    172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
* i 172.16.254.6 0 100 0 ?
* i [5] [1:1] [0] [32] [10.1.255.1]/17
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
* i [5] [1:1] [0] [32] [10.1.255.2]/17
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

* i [5][1:1][0][32][10.1.255.3]/17
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
* i [5][1:1][0][32][10.2.255.255]/17
    172.16.254.4 0 100 0 ?
* i 172.16.254.6 0 100 0 ?
* i 172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
* i [5][1:1][0][64][FC00:1:101::]/29
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
*>i [5][1:1][0][64][FC00:1:102::]/29
    172.16.254.4 0 100 0 ?
* i 172.16.254.4 0 100 0 ?
* i 172.16.254.6 0 100 0 ?
* i [5][1:1][0][128][FC00:1:255::1]/29
    172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
* i [5][1:1][0][128][FC00:1:255::2]/29
    172.16.254.4 0 100 0 ?
*>i 172.16.254.4 0 100 0 ?
* i [5][1:1][0][128][FC00:1:255::3]/29
    172.16.254.6 0 100 0 ?
*>i 172.16.254.6 0 100 0 ?
* i [5][1:1][0][128][FC00:2:255::255]/29
    172.16.254.4 0 100 0 ?
* i 172.16.254.6 0 100 0 ?
* i 172.16.254.3 0 100 0 ?
*>i 172.16.254.3 0 100 0 ?
Spine-01#

```

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 (?)
Spine-01#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 1:

```

Spine-01# 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

```



```

(*, 224.0.1.40), 1w2d/00:02:38, RP 172.16.255.255, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Sparse, 1w2d/00:02:38

(*, 225.0.0.102), 1w2d/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, 225.0.0.102), 02:04:11/00:02:37, flags: PA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null

(172.16.254.6, 225.0.0.102), 02:07:34/00:00:34, flags: PTA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null
Spine-01#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 1:

```

Spine-01# 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: 84/84/0
  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
  Tunnell Flags: A
  Loopback2 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 4/0/193/0, Other: 41/0/41
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
  GigabitEthernet1/0/2 Flags: NS
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/206/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  GigabitEthernet1/0/4 Flags: A NS
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0

```

```
HW Forwarding: 0/0/0/0, Other: 0/0/0
Spine-01#
```

Return to [Verifying TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic](#), on page 267

Outputs to Verify the Configuration on Spine Switch 2

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Spine Switch 2:

```
Spine-02# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 20, main routing table version 20
2 network entries using 784 bytes of memory
5 path entries using 800 bytes of memory
2/2 BGP path/bestpath attribute entries using 608 bytes of memory
3 BGP rinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
13 BGP extended community entries using 1808 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 4144 total bytes of memory
BGP activity 1031/999 prefixes, 7443/7372 paths, scan interval 60 secs
2 networks peaked at 11:17:12 Sep 16 2020 UTC (02:22:21.833 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   169    163     20    0    0 02:09:41      2
172.16.255.3  4      65001   150    169     20    0    0 02:09:38      1
172.16.255.4  4      65001   151    168     20    0    0 02:09:14      1
172.16.255.6  4      65001   150    167     20    0    0 02:09:18      1
Spine-02#
```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Spine Switch 2:

```
Spine-02# show bgp ipv6 mvpn all
BGP table version is 20, local router ID is 172.16.255.2
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: 1:1
* i   [5][1:1][FC00:1:101::11][FF06:1::1]/42
      172.16.255.3      0    100    0 ?
*>i   172.16.255.3      0    100    0 ?
Route Distinguisher: 172.16.254.3:101
*>i   [7][172.16.254.3:101][65001][FC00:1:101::11][FF06:1::1]/46
      172.16.255.4      0    100    0 ?
* i   172.16.255.4      0    100    0 ?
* i   172.16.255.6      0    100    0 ?
Spine-02#
```

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 712, main routing table version 712
30 network entries using 10320 bytes of memory
66 path entries using 13728 bytes of memory
11/11 BGP path/bestpath attribute entries using 3168 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
13 BGP extended community entries using 1808 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 29168 total bytes of memory
BGP activity 1031/999 prefixes, 7443/7372 paths, scan interval 60 secs
44 networks peaked at 10:13:54 Aug 6 2020 UTC (5w6d ago)

Neighbor          V            AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1     4            65001    169    163     712   0    0 02:09:41      30
172.16.255.3     4            65001    150    169     712   0    0 02:09:38      12
172.16.255.4     4            65001    151    168     712   0    0 02:09:15      12
172.16.255.6     4            65001    150    167     712   0    0 02:09:18      12
Spine-02#

```

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

```

Spine-02# show bgp l2vpn evpn
BGP table version is 712, local router ID is 172.16.255.2
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: 172.16.254.3:101
* i [2] [172.16.254.3:101] [0] [48] [10B3D56A8FC1] [32] [10.1.101.1]/24
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.254.3:101] [0] [48] [10B3D56A8FC1] [128] [FC00:1:101::1]/36
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [0] [*]/20
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [32] [10.1.101.11]/24
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FC00:1:101::11]/36
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [128] [FE80::F6CF:E2FF:FE43:34C1]/36
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:102
* i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [0] [*]/20
      172.16.254.4          0      100      0 ?
*>i          172.16.254.4          0      100      0 ?
* i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [32] [10.1.102.12]/24
      172.16.254.4          0      100      0 ?
*>i          172.16.254.4          0      100      0 ?
* i [2] [172.16.254.4:102] [0] [48] [44D3CA286CC5] [128] [FC00:1:102::12]/36
      172.16.254.4          0      100      0 ?

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

*>i          172.16.254.4          0 100 0 ?
* i [2][172.16.254.4:102][0][48][44D3CA286CC5][128][FE80::46D3:CAFF:FE28:6CC5]/36
    172.16.254.4          0 100 0 ?
*>i          172.16.254.4          0 100 0 ?
* i [2][172.16.254.4:102][0][48][7C210DBD954D][32][10.1.102.1]/24
    172.16.254.4          0 100 0 ?
*>i          172.16.254.4          0 100 0 ?
* i [2][172.16.254.4:102][0][48][7C210DBD954D][128][FC00:1:102::1]/36
    172.16.254.4          0 100 0 ?
*>i          172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.254.6:102
* i [2][172.16.254.6:102][0][48][0C75BD67EF4D][32][10.1.102.1]/24
    172.16.254.6          0 100 0 ?
*>i          172.16.254.6          0 100 0 ?
* i [2][172.16.254.6:102][0][48][0C75BD67EF4D][128][FC00:1:102::1]/36
    172.16.254.6          0 100 0 ?
*>i          172.16.254.6          0 100 0 ?
* i [2][172.16.254.6:102][0][48][ECE1A93792C5][0][*]/20
    172.16.254.6          0 100 0 ?
*>i          172.16.254.6          0 100 0 ?
* i [2][172.16.254.6:102][0][48][ECE1A93792C5][32][10.1.102.13]/24
    172.16.254.6          0 100 0 ?
*>i          172.16.254.6          0 100 0 ?
* i [2][172.16.254.6:102][0][48][ECE1A93792C5][128][FC00:1:102::13]/36
    172.16.254.6          0 100 0 ?
*>i          172.16.254.6          0 100 0 ?
* i [2][172.16.254.6:102][0][48][ECE1A93792C5][128][FE80::EEE1:A9FF:FE37:92C5]/36
    172.16.254.6          0 100 0 ?
*>i          172.16.254.6          0 100 0 ?
Route Distinguisher: 1:1
* i [5][1:1][0][24][10.1.101.0]/17
    172.16.254.3          0 100 0 ?
*>i          172.16.254.3          0 100 0 ?
*>i [5][1:1][0][24][10.1.102.0]/17
    172.16.254.4          0 100 0 ?
* i          172.16.254.4          0 100 0 ?
* i          172.16.254.6          0 100 0 ?
* i [5][1:1][0][32][10.1.255.1]/17
    172.16.254.3          0 100 0 ?
*>i          172.16.254.3          0 100 0 ?
* i [5][1:1][0][32][10.1.255.2]/17
    172.16.254.4          0 100 0 ?
*>i          172.16.254.4          0 100 0 ?
* i [5][1:1][0][32][10.1.255.3]/17
    172.16.254.6          0 100 0 ?
*>i          172.16.254.6          0 100 0 ?
* i [5][1:1][0][32][10.2.255.255]/17
    172.16.254.4          0 100 0 ?
* i          172.16.254.6          0 100 0 ?
* i          172.16.254.3          0 100 0 ?
*>i          172.16.254.3          0 100 0 ?
* i [5][1:1][0][64][FC00:1:101::]/29
    172.16.254.3          0 100 0 ?
*>i          172.16.254.3          0 100 0 ?
*>i [5][1:1][0][64][FC00:1:102::]/29
    172.16.254.4          0 100 0 ?
* i          172.16.254.4          0 100 0 ?
* i          172.16.254.6          0 100 0 ?
* i [5][1:1][0][128][FC00:1:255::1]/29
    172.16.254.3          0 100 0 ?
*>i          172.16.254.3          0 100 0 ?
* i [5][1:1][0][128][FC00:1:255::2]/29
    172.16.254.4          0 100 0 ?
*>i          172.16.254.4          0 100 0 ?

```

```

* i [5][1:1][0][128][FC00:1:255::3]/29
    172.16.254.6          0    100    0 ?
*>i          172.16.254.6          0    100    0 ?
* i [5][1:1][0][128][FC00:2:255::255]/29
    172.16.254.4          0    100    0 ?
* i          172.16.254.6          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
*>i          172.16.254.3          0    100    0 ?
Spine-02#

```

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 (?)
Spine-02#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 2:

```

Spine-02# 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), 22:51:54/00:03:26, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 02:09:47/00:03:26
    GigabitEthernet1/0/2, Forward/Sparse, 02:09:20/00:02:34
    GigabitEthernet1/0/4, Forward/Sparse, 02:09:16/00:03:12

(172.16.254.3, 239.1.1.1), 02:03:40/00:02:43, flags: TA
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
  Outgoing interface list:
    GigabitEthernet1/0/4, Forward/Sparse, 02:03:40/00:03:12
    GigabitEthernet1/0/2, Forward/Sparse, 02:03:40/00:02:46

(*, 224.0.1.40), 1w2d/00:03:18, RP 172.16.255.255, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:02:09/00:03:18
    GigabitEthernet1/0/1, Forward/Sparse, 00:02:10/00:03:17
    Loopback2, Forward/Sparse, 1w2d/00:02:45

(*, 225.0.0.102), 1w2d/00:03:22, RP 172.16.255.255, flags: S

```

Example: Configuring TRM in PIM Sparse Mode with Anycast RP

```

Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/1, Forward/Sparse, 02:09:47/00:02:35
  GigabitEthernet1/0/2, Forward/Sparse, 02:09:20/00:03:16
  GigabitEthernet1/0/4, Forward/Sparse, 02:09:16/00:03:22

(172.16.254.6, 225.0.0.102), 02:09:47/00:01:33, flags: MT
Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 02:09:16/00:03:16
  GigabitEthernet1/0/1, Forward/Sparse, 02:09:16/00:03:05

(172.16.254.4, 225.0.0.102), 02:09:47/00:02:06, flags: MT
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
Outgoing interface list:
  GigabitEthernet1/0/1, Forward/Sparse, 02:09:19/00:03:10
  GigabitEthernet1/0/4, Forward/Sparse, 02:09:16/00:03:22

(*, 225.0.0.101), 1w2d/00:03:29, RP 172.16.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/1, Forward/Sparse, 02:09:47/00:03:29
  GigabitEthernet1/0/2, Forward/Sparse, 02:09:20/00:02:31
  GigabitEthernet1/0/4, Forward/Sparse, 02:09:16/00:03:29

(172.16.254.3, 225.0.0.101), 02:09:22/00:03:25, flags: TA
Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 02:09:20/00:02:39
  GigabitEthernet1/0/4, Forward/Sparse, 02:09:16/00:03:29
Spine-02#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 2:

```

Spine-02# 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: 1/1/0
  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
Tunnel1 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps

```

```

Loopback2 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
SW Forwarding: 2/0/140/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 604/0/178/0, Other: 0/0/0
GigabitEthernet1/0/1 Flags: A NS
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
SW Forwarding: 4/0/124/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 668/0/176/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 600/0/205/0, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 17/0/397/0, Other: 4/4/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 3693/0/168/0, Other: 0/0/0
GigabitEthernet1/0/1 Flags: A
GigabitEthernet1/0/2 Flags: F NS

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

Pkts: 0/0/0    Rate: 0 pps
GigabitEthernet1/0/4  Flags: F NS
Pkts: 0/0/0    Rate: 0 pps
Spine-02#

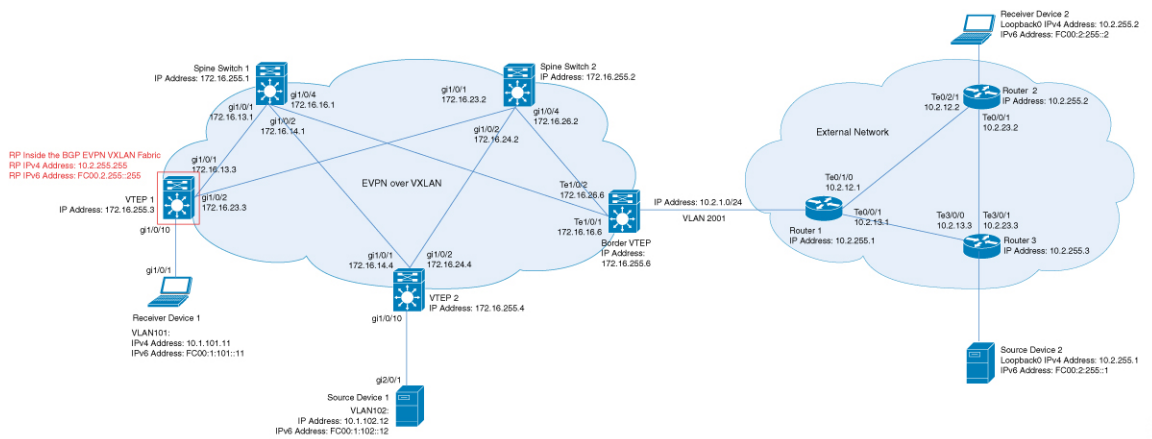
```

Return to [Verifying TRM in PIM-SM with Anycast RP for IPv4 and IPv6 Multicast Traffic, on page 267](#)

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

This example shows how to configure and verify Layer 3 TRM with PIM-SM for IPv4 multicast traffic when the RP is inside the BGP EVPN VXLAN fabric. The example uses the following topology:

Figure 24: TRM with PIM-SM when the RP is Inside the BGP EVPN VXLAN Fabric



The topology shows an EVPN VXLAN network, with two spine switches and three VTEPs, connected to an external network with three routers. VTEP 1 inside the BGP EVPN VXLAN fabric acts as the RP in this topology and Border VTEP connects the fabric to the external network through Router 1. The IPv4 multicast group is 226.1.1.1 in this topology. The following tables provide sample configurations for the devices in this topology:

Table 31: Configuring VTEP 1, Border VTEP, and VTEP 2 to Configure TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre>Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! 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 ! system mtu 9198 ! 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 pim sparse-mode ip ospf 1 area 0</pre>	<pre>Border# show running-config hostname Border ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! vlan 2001 !</pre>	<pre>Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-famil ! ip routing ! ip multicast-routing ip multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 !</pre>

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> ! interface Loopback1 ip address 172.16.254.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback255 vrf forwarding green ip address 10.2.255.255 255.255.255.255 ip pim sparse-mode ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode no autostate ! </pre>	<pre> 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 ! interface TenGigabitEthernet1/0/2 no switchport ip address 172.16.26.6 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/5 switchport trunk allowed vlan 2001 switchport mode trunk ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode no autostate ! </pre>	<pre> 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 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 GigabitEthernet1/0/2 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 access vlan 102 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode no autostate </pre>

VTEP 1	Border VTEP	VTEP 2
<pre> 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! address-family ipv4 mvpn 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 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 </pre>	<pre> interface Vlan2001 vrf forwarding green ip address 10.2.1.1 255.255.255.0 ip mtu 1500 ip pim sparse-mode ip ospf network point-to-point ip ospf 2 area 0 ! 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 mcast-group 225.0.0.102 ! router ospf 2 vrf green redistribute bgp 65001 ! router ospf 1 router-id 172.16.255.6 ! 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 ipv4 mvpn 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 </pre>	<pre> ! 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! address-family ipv4 mvpn 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 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 ! </pre>

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! end ! Leaf-01# </pre>	<pre> ! address-family l2vpn evpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both exit-address-family ! address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static redistribute ospf 2 match internal external 1 external 2 exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! end ! Border# </pre>	<pre> address-family ipv4 vrf green advertise l2vpn evpn redistribute connected redistribute static exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! end ! Leaf-02# </pre>

Table 32: Configuring Spine Switch 1 and Spine Switch 2 to Configure TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 GigabitEthernet1/0/4 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 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 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 ! interface GigabitEthernet1/0/4 no switchport ip address 172.16.26.2 255.255.255.0 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>

Spine Switch 1	Spine Switch 2
<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-01# </pre>	<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-02# </pre>

Table 33: Configuring Router 1, Router 2, and Router 3 to Configure TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

Router 1	Router 2	Router 3
<pre> Router-01# show running-config hostname R1 ! ip multicast-routing distributed ! interface Loopback0 ip address 10.2.255.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet0/0/1.2001 encapsulation dot1Q 2001 ip address 10.2.1.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 router ospf 1 router-id 10.2.255.1 ! ip pim rp-address 10.2.255.255 ! end ! R1# </pre>	<pre> Router-02# show running-config hostname R2 ! ip multicast-routing distributed ! interface Loopback0 ip address 10.2.255.2 255.255.255.255 ip pim sparse-mode ip igmp join-group 226.1.1.1 ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 10.2.255.2 ! ip pim rp-address 10.2.255.255 ! end ! R2# </pre>	<pre> Router-03# show running-config hostname R3 ! ip multicast-routing distributed ! interface Loopback0 ip address 10.2.255.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 10.2.255.3 ! ip pim rp-address 10.2.255.255 ! end ! R3# </pre>

Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

The following sections provide sample outputs for **show** commands to verify TRM with PIM-SM on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1 \(RP Inside the BGP EVPN VXLAN Fabric\), on page 308](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 314](#)
- [Outputs to Verify the Configuration on Border VTEP, on page 319](#)

- [Outputs to Verify the Configuration on Spine Switch 1, on page 325](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 328](#)

Outputs to Verify the Configuration on VTEP 1 (RP Inside the BGP EVPN VXLAN Fabric)

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

```
Leaf-01# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1      50901  L3CP 172.16.254.6    0c75.bd67.ef48 50901      UP  A/-/4 1d05h
nve1      50901  L3CP 172.16.254.4    7c21.0dbd.9548 50901      UP  A/-/4 1d05h
nve1      50901  L3CP 172.16.254.6    0c75.bd67.ef48 50901      UP  A/M/6 1d05h
nve1      50901  L3CP 172.16.254.4    7c21.0dbd.9548 50901      UP  A/M/6 1d05h
nve1      10102  L2CP 172.16.254.4    7              10102      UP  N/A    1d05h
nve1      10102  L2CP 172.16.254.6    5              10102      UP  N/A    1d05h
Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1      10102  172.16.254.4    7          10102    1d05h
nve1      10102  172.16.254.6    5          10102    1d05h
Leaf-01#
```

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on VTEP 1:

```
Leaf-01# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 111, main routing table version 111
7 network entries using 2128 bytes of memory
9 path entries using 1224 bytes of memory
5/5 BGP path/bestpath attribute entries using 1560 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
18 BGP extended community entries using 2396 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 7492 total bytes of memory
BGP activity 140/45 prefixes, 240/112 paths, scan interval 60 secs
9 networks peaked at 12:22:24 Aug 6 2020 UTC (1d05h ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   2104   1988    111   0     0 1d05h    2
172.16.255.2  4      65001   2099   1988    111   0     0 1d05h    2
Leaf-01#
```

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

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



```

Group(s): 224.0.0.0/4, Static
        RP: 10.2.255.255 (?)
Leaf-01#

```

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

```

Leaf-01# show ip routing vrf green 10.2.255.255
Routing Table: green
Routing entry for 10.2.255.255/32
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via bgp 65001
  Advertised by bgp 65001
  Routing Descriptor Blocks:
    * directly connected, via Loopback255
      Route metric is 0, traffic share count is 1
Leaf-01#

```

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

```

Leaf-01# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter    Group Accounted
226.1.1.1          Vlan102           1d05h    00:02:50   10.1.102.12
224.0.1.40         Vlan901           1d05h    00:02:03   172.16.254.4
Leaf-01#

```

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

```

Leaf-01# 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,
       e - encaps-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.1.1.1), 1d01h/stopped, RP 10.2.255.255, flags: SJCGx
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan101, Forward/Sparse, 1d01h/00:02:33
    Vlan901, Forward/Sparse, 03:54:15/stopped

(10.2.255.1, 226.1.1.1), 00:01:13/00:01:50, flags: Tgx
  Incoming interface: Vlan901, RPF nbr 172.16.254.6
  Outgoing interface list:
    Vlan101, Forward/Sparse, 00:01:13/00:02:33

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```
(10.1.102.12, 226.1.1.1), 00:01:36/00:01:24, flags: Tgx
Incoming interface: Vlan901, RPF nbr 172.16.254.4
Outgoing interface list:
  Vlan101, Forward/Sparse, 00:01:36/00:02:33

(*, 224.0.1.40), 1d05h/00:02:09, RP 10.2.255.255, flags: SJCLGx
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Loopback901, Forward/Sparse, 1d05h/00:02:09
  Vlan901, Forward/Sparse, 03:54:15/stoppped
Leaf-01#
```

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

```
Leaf-01# 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,
             e - Encap helper tunnel flag.
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: 2/2/0
  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
  Tunnel6 Flags: A
  Loopback901 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
  Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
    Pkts: 0/0/0   Rate: 0 pps
(*,226.1.1.1) Flags: C HW
  SW Forwarding: 1/0/100/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel6 Flags: A
  Vlan101 Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
  Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
    Pkts: 0/0/1   Rate: 0 pps
(10.1.102.12,226.1.1.1) Flags: HW
  SW Forwarding: 2/0/100/0, Other: 0/0/0
  HW Forwarding: 44/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan101 Flags: F NS
    Pkts: 0/0/2   Rate: 0 pps
(10.2.255.1,226.1.1.1) Flags: HW
  SW Forwarding: 5/0/100/0, Other: 12576/1/12575
  HW Forwarding: 3801/1/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan901 Flags: SP
```

```
Vlan101 Flags: F NS
Pkts: 0/0/5 Rate: 0 pps
Leaf-01#
```

The following example shows the output for the **show bgp ipv4 mvpn all** command on VTEP 1:

```
Leaf-01# show bgp ipv4 mvpn all
BGP table version is 94, local router ID is 172.16.255.3
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: 1:1 (default for vrf green)
* i   [5][1:1][10.1.102.12][226.1.1.1]/18
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
*>i   [5][1:1][10.2.255.1][226.1.1.1]/18
      172.16.255.6          0      100      0 ?
* i   172.16.255.6          0      100      0 ?
* i   [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
* i   [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
*>   [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      0.0.0.0                32768 ?
Route Distinguisher: 172.16.254.4:102
*>   [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      0.0.0.0                32768 ?
Leaf-01#
```

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

```
Leaf-01# 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,
       e - encap-helper tunnel flag
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), 1d05h/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, 1d05h/00:02:10
```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```
(172.16.254.6, 239.1.1.1), 00:01:11/00:01:48, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:01:11/00:01:48

(172.16.254.3, 239.1.1.1), 00:01:37/00:01:22, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0, Registering
Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:01:37/00:02:51, A

(172.16.254.4, 239.1.1.1), 04:17:32/00:02:31, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
    Tunnel0, Forward/Sparse, 04:17:32/00:00:27

(*, 224.0.1.40), 1d05h/00:02:12, RP 172.16.255.255, flags: SJCL
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
    Loopback0, Forward/Sparse, 1d05h/00:02:12

(*, 225.0.0.102), 1d05h/stopped, RP 172.16.255.255, flags: SJCx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:02:10

(172.16.254.4, 225.0.0.102), 1d05h/00:01:20, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:01:32

(172.16.254.6, 225.0.0.102), 1d05h/00:02:44, flags: JTx
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:02:10

(*, 225.0.0.101), 1d05h/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, 1d05h/00:02:10

(172.16.254.3, 225.0.0.101), 1d05h/00:02:36, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:03:20
Leaf-01#
```

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

```
Leaf-01# 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,
             e - Encap helper tunnel flag.
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
  GigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/114/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 13/0/127/0, Other: 2/2/0
  HW Forwarding: 12525/0/165/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2/0/172/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 9155/0/176/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 3762/0/163/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 2/2/0
  HW Forwarding: 15/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 2/1/1
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F
    Pkts: 0/0/0   Rate: 0 pps
  Tunnel4 Flags: F
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding: 7707/0/167/0, Other: 0/0/0

```

```

GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 2/0/150/0, Other: 0/0/0
HW Forwarding: 68/1/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
Leaf-01#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric, on page 307](#)

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.6     0c75.bd67.ef48 50901     UP  A/-/4 1d05h
nve1     50901    L3CP 172.16.254.3     10b3.d56a.8fc8 50901     UP  A/-/4 1d05h
nve1     50901    L3CP 172.16.254.6     0c75.bd67.ef48 50901     UP  A/M/6 1d05h
nve1     50901    L3CP 172.16.254.3     10b3.d56a.8fc8 50901     UP  A/M/6 1d05h
nve1     10101    L2CP 172.16.254.3       6           10101     UP  N/A 1d05h
nve1     10102    L2CP 172.16.254.6       5           10102     UP  N/A 1d05h
Leaf-02#

```

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

```

Leaf-02# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1     10101    172.16.254.3     6          10101     1d05h
nve1     10102    172.16.254.6     5          10102     1d05h
Leaf-02#

```

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on VTEP 2:

```

Leaf-02# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 62, main routing table version 62
7 network entries using 2128 bytes of memory
9 path entries using 1224 bytes of memory
4/4 BGP path/bestpath attribute entries using 1248 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2372 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 7156 total bytes of memory
BGP activity 121/28 prefixes, 202/77 paths, scan interval 60 secs
9 networks peaked at 12:22:24 Aug 6 2020 UTC (16:43:21.423 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001  1229   1151    62     0     0 16:57:50  2
172.16.255.2  4      65001  1227   1152    62     0     0 16:57:51  2

```

```
Leaf-02#
```

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

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

Group(s): 224.0.0.0/4, Static
          RP: 10.2.255.255 (?)
Leaf-02#
```

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

```
Leaf-02# show ip routing vrf green 10.2.255.255
Routing Table: green
Routing entry for 10.2.255.255/32
  Known via "bgp 65001", distance 200, metric 0, type internal
  Last update from 172.16.254.3 on Vlan901, 03:59:59 ago
  Routing Descriptor Blocks:
  * 172.16.254.3 (default), from 172.16.255.1, 03:59:59 ago, via Vlan901
    opaque_ptr 0x7F65B8B9E9F0
    Route metric is 0, traffic share count is 1
    AS Hops 0
    MPLS label: none
Leaf-02#
```

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

```
Leaf-02# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter    Group Accounted
226.1.1.1          Vlan102           16:58:00  00:02:11  10.1.102.12
224.0.1.40         Vlan901           16:58:37  00:02:33  172.16.254.4
Leaf-02#
```

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

```
Leaf-02# 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,
       e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```
(* , 226.1.1.1), 1d05h/stopped, RP 10.2.255.255, flags: SJCFg
Incoming interface: Vlan901, RPF nbr 172.16.254.3
Outgoing interface list:
  Vlan102, Forward/Sparse, 1d05h/00:02:50

(10.2.255.1, 226.1.1.1), 00:06:57/00:02:09, flags: TgQ
Incoming interface: Vlan901, RPF nbr 172.16.254.6
Outgoing interface list:
  Vlan102, Forward/Sparse, 00:06:57/00:02:50

(10.1.102.12, 226.1.1.1), 00:07:21/00:01:45, flags: FTGqx
Incoming interface: Vlan102, RPF nbr 0.0.0.0
Outgoing interface list:
  Vlan901, Forward/Sparse, 00:07:21/stopped

(* , 224.0.1.40), 1d05h/00:02:03, RP 10.2.255.255, flags: SJPClgx
Incoming interface: Vlan901, RPF nbr 172.16.254.3
Outgoing interface list: Null
Leaf-02#
```

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

```
Leaf-02# 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,
             e - Encap helper tunnel flag.
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
  Vlan901, VXLAN Decap Flags: A IC NS
(*,226.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 3/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A NS
  Vlan102 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(10.1.102.12,226.1.1.1) Flags: HW
  SW Forwarding: 2/0/100/0, Other: 0/0/0
  HW Forwarding: 215/0/118/0, Other: 0/0/0
  Vlan102 Flags: A
  Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
    Pkts: 0/0/2   Rate: 0 pps
(10.2.255.1,226.1.1.1) Flags: HW
  SW Forwarding: 2/0/100/0, Other: 0/0/0
Leaf-02#
```


The following example shows the output for the **show bgp ipv4 mvpn all** command on VTEP 2:

```
Leaf-02# show bgp ipv4 mvpn all
BGP table version is 94, local router ID is 172.16.255.3
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: 1:1 (default for vrf green)
* i   [5][1:1][10.1.102.12][226.1.1.1]/18
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
*>i   [5][1:1][10.2.255.1][226.1.1.1]/18
      172.16.255.6          0      100      0 ?
* i   172.16.255.6          0      100      0 ?
* i   [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
* i   [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
*>   [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      0.0.0.0                32768 ?
Route Distinguisher: 172.16.254.4:102
*>   [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      0.0.0.0                32768 ?
Leaf-02#
```

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

```
Leaf-02# 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,
       e - encaps-helper tunnel flag
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), 1d05h/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, 1d05h/00:01:59

(172.16.254.6, 239.1.1.1), 00:06:55/00:01:59, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:06:55/00:02:04
```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```
(172.16.254.4, 239.1.1.1), 04:23:16/00:03:29, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 04:23:16/00:02:57

(*, 224.0.1.40), 1d05h/00:02:02, RP 172.16.255.255, flags: SJCL
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
Outgoing interface list:
  Loopback0, Forward/Sparse, 1d05h/00:02:02
Leaf-02#
```

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

```
Leaf-02# 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,
             e - Encap helper tunnel flag.
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
  GigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2/0/170/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding: 12469/0/177/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/224/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 2/0/163/0, Other: 3/1/2
  HW Forwarding: 9233/0/164/0, Other: 0/0/0
```

```

Null0 Flags: A
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 1/0/206/0, Other: 0/0/0
HW Forwarding:  3767/0/163/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding:  8/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 22/18/4
HW Forwarding:  7870/0/156/0, Other: 0/0/0
Null0 Flags: A
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 2/0/150/0, Other: 0/0/0
HW Forwarding:  412/1/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
Leaf-02#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric, on page 307](#)

Outputs to Verify the Configuration on Border VTEP

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

```

Border# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RT's  eVNI      state flags UP time
nve1     50901      L3CP 172.16.254.3      10b3.d56a.8fc8 50901      UP  A/-/4 1d05h
nve1     50901      L3CP 172.16.254.4      7c21.0dbd.9548 50901      UP  A/-/4 1d05h
nve1     50901      L3CP 172.16.254.3      10b3.d56a.8fc8 50901      UP  A/M/6 1d05h
nve1     50901      L3CP 172.16.254.4      7c21.0dbd.9548 50901      UP  A/M/6 1d05h
nve1     10101      L2CP 172.16.254.3        6             10101      UP  N/A 1d05h
nve1     10102      L2CP 172.16.254.4        7             10102      UP  N/A 1d05h
Border#

```

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

```

Border# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1     10101      172.16.254.3    6          10101     1d05h
nve1     10102      172.16.254.4    7          10102     1d05h
Border#

```

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on Border VTEP:

```
Border# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.6, local AS number 65001
BGP table version is 102, main routing table version 102
6 network entries using 1824 bytes of memory
8 path entries using 1088 bytes of memory
5/5 BGP path/bestpath attribute entries using 1560 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
20 BGP extended community entries using 2706 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 7362 total bytes of memory
BGP activity 133/39 prefixes, 265/144 paths, scan interval 60 secs
8 networks peaked at 12:14:22 Aug 6 2020 UTC (1d05h ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   2114   1995    102   0    0 1d05h    2
172.16.255.2  4      65001   2112   1990    102   0    0 1d05h    2
Border#
```

The following example shows the output for the **show ip pim vrf vrf-name rp mapping** command on Border VTEP:

```
Border# show ip pim vrf green rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 10.2.255.255 (?)
Border#
```

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

```
Border# show ip routing vrf green 10.2.255.255
Routing Table: green
Routing entry for 10.2.255.255/32
  Known via "bgp 65001", distance 200, metric 0, type internal
  Redistributing via ospf 2
  Advertised by ospf 2 subnets
  Last update from 172.16.254.3 on Vlan901, 04:02:51 ago
  Routing Descriptor Blocks:
  * 172.16.254.3 (default), from 172.16.255.1, 04:02:51 ago, via Vlan901
    opaque_ptr 0x7FEF6836D190
    Route metric is 0, traffic share count is 1
    AS Hops 0
    MPLS label: none
Border#
```

The following example shows the output for the **show ip igmp vrf vrf-name groups** command on Border VTEP:

```
Border# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address  Interface      Uptime    Expires    Last Reporter  Group Accounted
224.0.1.40     Vlan901        1d05h     00:01:58   172.16.254.6
Border#
```

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

```
Border# 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,
       e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.1.1.1), 1d05h/00:03:07, RP 10.2.255.255, flags: SJgx
  Incoming interface: Vlan901, RPF nbr 172.16.254.3
  Outgoing interface list:
    Vlan2001, Forward/Sparse, 04:02:51/00:03:07

(10.2.255.1, 226.1.1.1), 00:09:51/00:01:02, flags: TGqx
  Incoming interface: Vlan2001, RPF nbr 10.2.1.2
  Outgoing interface list:
    Vlan901, Forward/Sparse, 00:09:49/stopped

(10.1.102.12, 226.1.1.1), 00:10:12/00:03:09, flags: TgQx
  Incoming interface: Vlan901, RPF nbr 172.16.254.4
  Outgoing interface list:
    Vlan2001, Forward/Sparse, 00:10:12/00:03:07

(*, 224.0.1.40), 1d05h/00:03:10, RP 10.2.255.255, flags: SJCLgx
  Incoming interface: Vlan901, RPF nbr 172.16.254.3
  Outgoing interface list:
    Vlan2001, Forward/Sparse, 04:02:51/00:03:10
Border#
```

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

```
Border# 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,
            e - Encap helper tunnel flag.
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
```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

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
  Vlan901, VXLAN Decap Flags: A IC NS
  Vlan2001 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(*,226.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 6/0/122/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A NS
  Vlan2001 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(10.1.102.12,226.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 304/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan2001 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(10.2.255.1,226.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 1/0/1
  HW Forwarding: 582/1/122/0, Other: 0/0/0
  Vlan2001 Flags: A
  Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
    Pkts: 0/0/0   Rate: 0 pps
Border#

```

The following example shows the output for the **show bgp ipv4 mvpn all** command on Border VTEP:

```

Border# show bgp ipv4 mvpn allBGP table version is 102, local router ID is 172.16.255.6
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: 1:1 (default for vrf green)
* i   [5][1:1][10.1.102.12][226.1.1.1]/18
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
*>   [5][1:1][10.2.255.1][226.1.1.1]/18
      0.0.0.0                32768 ?
*>   [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      0.0.0.0                32768 ?
*>   [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      0.0.0.0                32768 ?
*>i   [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      172.16.255.3          0      100      0 ?
* i   172.16.255.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:102
*>   [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      0.0.0.0                32768 ?
Border#

```

The following example shows the output for the **show ip mroute** command on Border VTEP:

```

Border# 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,
       e - encap-helper tunnel flag
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), 1d05h/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:01:56

(172.16.254.6, 239.1.1.1), 00:09:47/00:02:24, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/2, Forward/Sparse, 00:09:47/00:02:33

(172.16.254.4, 239.1.1.1), 04:26:08/00:02:10, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 04:26:08/00:00:51

(*, 224.0.1.40), 1d05h/00:02:56, RP 172.16.255.255, flags: SJCL
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 1d05h/00:02:56

(*, 225.0.0.102), 1d05h/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:01:56

(172.16.254.4, 225.0.0.102), 1d05h/00:01:27, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:01:56

(172.16.254.6, 225.0.0.102), 1d05h/00:01:53, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:02:43, A

(*, 225.0.0.101), 1d05h/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:01:56

(172.16.254.3, 225.0.0.101), 1d05h/00:01:10, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

Outgoing interface list:
  Tunnel0, Forward/Sparse, 1d05h/00:02:08
Border#

```

The following example shows the output for the **show ip mfib** command on Border VTEP:

```

Border# 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,
                  e - Encap helper tunnel flag.
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
  TenGigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 8/0/146/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 7/0/125/0, Other: 0/0/0
  HW Forwarding: 12570/0/177/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/7   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/172/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 9199/0/176/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 17/0/174/0, Other: 10/9/1
  HW Forwarding: 3789/0/151/0, Other: 0/0/0
  Null0 Flags: A

```



```

TenGigabitEthernet1/0/2 Flags: F
  Pkts: 0/0/16   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  9/0/168/0, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding:  7961/0/167/0, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 2/0/150/0, Other: 2/2/0
  HW Forwarding:  580/1/156/1, Other: 0/0/0
Null0 Flags: A
TenGigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
Border#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric](#), on page 307

Outputs to Verify the Configuration on Spine Switch 1

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on Spine Switch 1:

```

Spine-01# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 204, main routing table version 204
6 network entries using 1824 bytes of memory
16 path entries using 2176 bytes of memory
4/4 BGP path/bestpath attribute entries using 1216 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2356 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 7716 total bytes of memory
BGP activity 266/218 prefixes, 3029/2926 paths, scan interval 60 secs
8 networks peaked at 12:20:11 Aug 6 2020 UTC (1d05h ago)

Neighbor          V            AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2      4            65001   2496   2445    204   0    0 1d08h      6
172.16.255.3      4            65001   1985   2115    204   0    0 1d05h      2
172.16.255.4      4            65001   1995   2111    204   0    0 1d05h      4
172.16.255.6      4            65001   1999   2118    204   0    0 1d05h      4
Spine-01#

```

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

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

Group(s): 224.0.0.0/4, Static
RP: 172.16.255.255 (?)
Spine-01#

```

The following example shows the output for the **show bgp ipv4 mvpn all** command on Spine Switch 1:

```

Spine-01# show bgp ipv4 mvpn all
BGP table version is 204, local router ID is 172.16.255.1
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: 1:1
* i   [5][1:1][10.1.102.12][226.1.1.1]/18
      172.16.255.4          0    100    0 ?
*>i   172.16.255.4          0    100    0 ?
* i   [5][1:1][10.2.255.1][226.1.1.1]/18
      172.16.255.6          0    100    0 ?
*>i   172.16.255.6          0    100    0 ?
* i   [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      172.16.255.4          0    100    0 ?
*>i   172.16.255.4          0    100    0 ?
* i   172.16.255.6          0    100    0 ?
* i   [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      172.16.255.4          0    100    0 ?
*>i   172.16.255.4          0    100    0 ?
* i   172.16.255.6          0    100    0 ?
* i   [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      172.16.255.4          0    100    0 ?
* i   172.16.255.3          0    100    0 ?
*>i   172.16.255.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:102
* i   [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      172.16.255.6          0    100    0 ?
* i   172.16.255.3          0    100    0 ?
*>i   172.16.255.3          0    100    0 ?
Spine-01#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 1:

```

Spine-01# 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), 04:29:40/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(172.16.254.6, 239.1.1.1), 00:13:17/00:02:24, flags: PA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null

(172.16.254.4, 239.1.1.1), 04:27:38/00:02:41, flags: PTA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null

(*, 224.0.1.40), 1w0d/00:02:43, RP 172.16.255.255, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Sparse, 1w0d/00:02:43

(*, 225.0.0.102), 1w0d/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null
Spine-01#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 1:

```

Spine-01# 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: 83/83/0
  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
  Tunnell Flags: A
  Loopback2 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 1/0/206/0, Other: 282/0/282
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnell Flags: A

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

GigabitEthernet1/0/2 Flags: NS
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/4 Flags: NS
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 4/3/1
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 7/0/158/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/4 Flags: NS
Spine-01#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric, on page 307](#)

Outputs to Verify the Configuration on Spine Switch 2

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on Spine Switch 2:

```

Spine-02# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 164, main routing table version 164
6 network entries using 1824 bytes of memory
16 path entries using 2176 bytes of memory
4/4 BGP path/bestpath attribute entries using 1216 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2356 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 7716 total bytes of memory
BGP activity 297/249 prefixes, 3131/3028 paths, scan interval 60 secs
8 networks peaked at 12:20:59 Aug 6 2020 UTC (1d05h ago)

Neighbor          V      AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down   State/PfxRcd
172.16.255.1      4      65001   2448   2499     164    0    0 1d08h      6
172.16.255.3      4      65001   1988   2114     164    0    0 1d05h      2
172.16.255.4      4      65001   1998   2110     164    0    0 1d05h      4
172.16.255.6      4      65001   1996   2119     164    0    0 1d05h      4
Spine-02#

```

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 (?)
Spine-02#

```

The following example shows the output for the **show bgp ipv4 mvpn all** command on Spine Switch 2:

```

Spine-02# show bgp ipv4 mvpn all
BGP table version is 164, local router ID is 172.16.255.2
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: 1:1
* i   [5][1:1][10.1.102.12][226.1.1.1]/18
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
* i   [5][1:1][10.2.255.1][226.1.1.1]/18
      172.16.255.6          0      100      0 ?
*>i   172.16.255.6          0      100      0 ?
*>i   [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      172.16.255.4          0      100      0 ?
* i   172.16.255.4          0      100      0 ?
* i   172.16.255.6          0      100      0 ?
*>i   [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      172.16.255.4          0      100      0 ?
* i   172.16.255.4          0      100      0 ?
* i   172.16.255.6          0      100      0 ?
* i   [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      172.16.255.4          0      100      0 ?
* i   172.16.255.3          0      100      0 ?
*>i   172.16.255.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:102
* i   [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      172.16.255.6          0      100      0 ?
* i   172.16.255.3          0      100      0 ?
*>i   172.16.255.3          0      100      0 ?
Spine-02#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 2:

```

Spine-02# 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), 3d12h/00:03:14, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:02:31
    GigabitEthernet1/0/1, Forward/Sparse, 1d05h/00:03:14
    GigabitEthernet1/0/4, Forward/Sparse, 1d05h/00:03:09

(172.16.254.6, 239.1.1.1), 00:15:48/00:01:26, flags: T
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 00:15:48/00:03:24
    GigabitEthernet1/0/2, Forward/Sparse, 00:15:48/00:03:26

(172.16.254.4, 239.1.1.1), 04:32:09/00:01:28, flags: T
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
  Outgoing interface list:
    GigabitEthernet1/0/4, Forward/Sparse, 04:32:09/00:03:09
    GigabitEthernet1/0/1, Forward/Sparse, 04:32:09/00:03:14

(*, 224.0.1.40), 1w0d/00:03:29, RP 172.16.255.255, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:02:48
    GigabitEthernet1/0/1, Forward/Sparse, 1d05h/00:03:29
    GigabitEthernet1/0/4, Forward/Sparse, 1d05h/00:03:17
    Loopback2, Forward/Sparse, 1w0d/00:02:34

(*, 225.0.0.102), 1w0d/00:03:28, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:02:55
    GigabitEthernet1/0/1, Forward/Sparse, 1d05h/00:03:00
    GigabitEthernet1/0/4, Forward/Sparse, 1d05h/00:03:28

(172.16.254.4, 225.0.0.102), 1d05h/00:02:09, flags: MT
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
  Outgoing interface list:
    GigabitEthernet1/0/4, Forward/Sparse, 1d05h/00:03:28
    GigabitEthernet1/0/1, Forward/Sparse, 1d05h/00:03:17

(172.16.254.6, 225.0.0.102), 1d05h/00:01:40, flags: MT
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:02:59
    GigabitEthernet1/0/1, Forward/Sparse, 1d05h/00:03:05

(*, 225.0.0.101), 3d12h/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, 1d05h/00:02:33
    GigabitEthernet1/0/1, Forward/Sparse, 1d05h/00:03:21
    GigabitEthernet1/0/4, Forward/Sparse, 1d05h/00:02:47

(172.16.254.3, 225.0.0.101), 1d05h/00:02:05, flags: TA
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:02:57
    GigabitEthernet1/0/4, Forward/Sparse, 1d05h/00:02:47
Spine-02#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 2:

```
Spine-02# 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
  Tunnell Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  Loopback2 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 9/0/112/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/2   Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/2   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 4/0/132/0, Other: 0/0/0
  HW Forwarding: 12607/0/177/0, Other: 0/0/0
  GigabitEthernet1/0/1 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/4   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 27/0/101/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
```

```

HW Forwarding: 9232/0/176/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 3789/0/163/0, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 10/0/150/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 8144/0/167/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 942/1/168/1, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
Spine-02#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric](#), on page 307

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

This example shows how to configure and verify Layer 3 TRM with PIM-SM for IPv4 and IPv6 multicast traffic when the RP is inside the BGP EVPN VXLAN fabric. The example uses the topology in the [Figure 24: TRM with PIM-SM when the RP is Inside the BGP EVPN VXLAN Fabric](#), on page 300 figure.

The topology shows an EVPN VXLAN network, with two spine switches and three VTEPs, connected to an external network with three routers. VTEP 1 inside the BGP EVPN VXLAN fabric acts as the RP in this topology and Border VTEP connects the fabric to the external network through Router 1. The IPv4 multicast

group is 226.1.1.1 and the IPv6 multicast group is FF06:1::1 in this topology. The following tables provide sample configurations for the devices in this topology:

Table 34: Configuring VTEP 1, Border VTEP, and VTEP 2 to Configure TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre>Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 !</pre>	<pre>Border# show running-config hostname Border ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 !</pre>	<pre>Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 !</pre>

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip pim sparse-mode 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 Loopback255 vrf forwarding green ip address 10.2.255.255 255.255.255.255 ip pim sparse-mode ipv6 address FC00:2:255::255/128 ipv6 enable ! interface Loopback901 vrf forwarding green ip address 10.1.255.1 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::1/128 ipv6 enable ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable </pre>	<pre> vlan 2001 ! 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 Loopback901 vrf forwarding green ip address 10.1.255.4 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::4/128 ipv6 enable ! 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 ! interface TenGigabitEthernet1/0/2 no switchport ip address 172.16.26.6 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/5 switchport trunk allowed vlan 2001 switchport mode trunk ! interface TenGigabitEthernet1/0/10 switchport access vlan 102 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! </pre>	<pre> 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 Loopback901 vrf forwarding green ip address 10.1.255.2 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::2/128 ipv6 enable ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable </pre>

VTEP 1	Border VTEP	VTEP 2
<pre> ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! address-family ipv4 mvpn 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 ! </pre>	<pre> interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode ipv6 enable no autostate ! interface Vlan2001 vrf forwarding green ip address 10.2.1.1 255.255.255.0 ip mtu 1500 ip pim sparse-mode ip ospf network point-to-point ip ospf 2 area 0 ipv6 address FC00:2:1::1/64 ipv6 enable ip mtu 1500 ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! 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 mcast-group 225.0.0.102 ! router ospfv3 1 ! address-family ipv6 unicast vrf green redistribute bgp 65001 exit-address-family ! router ospf 2 vrf green redistribute bgp 65001 ! router ospf 1 router-id 172.16.255.6 ! 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 </pre>	<pre> ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 </pre>

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> address-family ipv6 mvpn 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 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 ! 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 ip pim vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end ! Leaf-01# </pre>	<pre> ! address-family ipv4 exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 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 ! 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 ip pim vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end ! Leaf-02# </pre>	<pre> ! 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 ! 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 ip pim vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end ! Leaf-02# </pre>

Table 35: Configuring Spine Switch 1 and Spine Switch 2 to Configure TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 GigabitEthernet1/0/4 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 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 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 ! interface GigabitEthernet1/0/4 no switchport ip address 172.16.26.2 255.255.255.0 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family ipv6 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-01# </pre>	<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family ipv6 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-02# </pre>

Table 36: Configuring Router 1, Router 2, and Router 3 to Configure TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Router 1	Router 2	Router 3
<pre> R1# show running-config hostname R1 ! ip multicast-routing distributed ! ipv6 unicast-routing ipv6 multicast-routing ! interface Loopback0 ip address 10.2.255.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ipv6 address FC00:2:255::1/128 ipv6 enable ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:12::1/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:13::1/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface GigabitEthernet0/0/1.2001 encapsulation dot1Q 2001 ip address 10.2.1.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ipv6 address FC00:2:1::2/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! </pre>	<pre> R2# show running-config hostname R2 ! ip multicast-routing distributed ! ipv6 unicast-routing ipv6 multicast-routing ! interface Loopback0 ip address 10.2.255.2 255.255.255.255 ip pim sparse-mode ip igmp join-group 226.1.1.1 ip ospf 1 area 0 ipv6 address FC00:2:255::2/128 ipv6 enable ipv6 mld join-group FF06:1::1 ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:12::2/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:23::2/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! router ospfv3 1 ! address-family ipv6 unicast exit-address-family ! router ospf 1 router-id 10.2.255.2 ! ip pim rp-address 10.2.255.255 </pre>	<pre> R3# show running-config hostname R3 ! ip multicast-routing distributed ! ipv6 unicast-routing ipv6 multicast-routing ! interface Loopback0 ip address 10.2.255.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ipv6 address FC00:2:255::3/128 ipv6 enable ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:13::3/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:23::3/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! router ospfv3 1 ! address-family ipv6 unicast exit-address-family ! router ospf 1 router-id 10.2.255.3 ! ip pim rp-address 10.2.255.255 ! </pre>

Router 1	Router 2	Router 3
<pre> router ospfv3 1 ! address-family ipv6 unicast exit-address-family ! router ospf 1 router-id 10.2.255.1 ! ip pim rp-address 10.2.255.255 ! ipv6 pim rp-address FC00:2:255::255 ! end ! R1# </pre>	<pre> ! ipv6 pim rp-address FC00:2:255::255 ! end ! R2# </pre>	<pre> ipv6 pim rp-address FC00:2:255::255 ! end ! R3# </pre>

Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

The following sections provide sample outputs for **show** commands to verify TRM with PIM-SM on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1 \(RP Inside the BGP EVPN VXLAN Fabric\), on page 340](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 348](#)
- [Outputs to Verify the Configuration on Border VTEP, on page 355](#)
- [Outputs to Verify the Configuration on Spine Switch 1, on page 363](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 366](#)

Outputs to Verify the Configuration on VTEP 1 (RP Inside the BGP EVPN VXLAN Fabric)

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

```

Leaf-01# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1      50901  L3CP 172.16.254.6    0c75.bd67.ef48 50901      UP    A/-/4 1d05h
nve1      50901  L3CP 172.16.254.4     7c21.0dbd.9548 50901      UP    A/-/4 1d05h
nve1      50901  L3CP 172.16.254.6     0c75.bd67.ef48 50901      UP    A/M/6 1d05h
nve1      50901  L3CP 172.16.254.4     7c21.0dbd.9548 50901      UP    A/M/6 1d05h
nve1      10102  L2CP 172.16.254.4      7              10102      UP    N/A   1d05h
nve1      10102  L2CP 172.16.254.6      5              10102      UP    N/A   1d05h
Leaf-01#

```

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

```

Leaf-01# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1      10102  172.16.254.4    7          10102    1d05h
nve1      10102  172.16.254.6    5          10102    1d05h
Leaf-01#

```


The following example shows the output for the **show bgp ipv6 mvpn all summary** command on VTEP 1:

```
Leaf-01# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 60, main routing table version 60
5 network entries using 1960 bytes of memory
8 path entries using 1280 bytes of memory
4/4 BGP path/bestpath attribute entries using 1248 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
18 BGP extended community entries using 2396 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 7068 total bytes of memory
BGP activity 139/41 prefixes, 275/138 paths, scan interval 60 secs
5 networks peaked at 15:46:09 Aug 6 2020 UTC (1d02h ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   2135   2003     60   0    0 1d05h    3
172.16.255.2  4      65001   2131   2003     60   0    0 1d05h    3
Leaf-01#
```

The following example shows the output for the **show ipv6 pim vrf vrf-name group-map** command on VTEP 1:

```
Leaf-01# show ipv6 pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF06:1::8*
  SM, RP: FC00:2:255::255
  RPF: Tu7,FC00:2:255::255 (us)
  Info source: Static
  Uptime: 1d02h, Groups: 1
Leaf-01#
```

The following example shows the output for the **show ipv6 routing vrf** command on VTEP 1:

```
Leaf-01# show ipv6 routing vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "connected", distance 0, metric 0, type receive, connected
  Redistributing via bgp 65001
  Route count is 1/1, share count 0
  Routing paths:
    receive via Loopback255
    Last updated 04:21:51 ago
Leaf-01#
```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on VTEP 1:

```
Leaf-01# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
  Uptime    Expires
FF06:1::1                                Vlan101
  1d02h    00:02:28
Leaf-01#
```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on VTEP 1:

```
Leaf-01# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       n - BGP Shared-Tree Prune received, N - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 1d02h/00:03:13, RP FC00:2:255::255, flags: SCJG
  Incoming interface: Tunnel7
  RPF nbr: FC00:2:255::255
  Immediate Outgoing interface list:
    Vlan101, Forward, 1d02h/00:03:13
    Vlan901, Forward, 04:21:51/never

(FC00:1:102::12, FF06:1::1), 00:02:17/00:03:04, flags: SJTg
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.4
  Immediate Outgoing interface list:
    Vlan101, Forward, 00:02:13/00:03:23

(FC00:2:255::1, FF06:1::1), 00:01:24/00:03:04, flags: SJTg
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.6
  Immediate Outgoing interface list:
    Vlan101, Forward, 00:01:20/00:03:13
Leaf-01#
```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on VTEP 1:

```
Leaf-01# show ipv6 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,
             e - Encap helper tunnel flag.
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
(*,FF00::/8) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 412/412/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel7 Flags: NS
```

```

(*,FF00::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF02::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 2/0/100/0, Other: 0/0/0
  HW Forwarding: 4/0/126/0, Other: 0/0/0
  Tunnel7 Flags: A NS
  Vlan101 Flags: F NS
    Pkts: 0/0/2    Rate: 0 pps
  Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
    Pkts: 0/0/2    Rate: 0 pps
(FC00:1:102::12,FF06:1::1) Flags: HW
  SW Forwarding: 1/0/100/0, Other: 0/0/0
  HW Forwarding: 64/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan101 Flags: F NS
    Pkts: 0/0/1    Rate: 0 pps
(FC00:2:255::1,FF06:1::1) Flags: HW
  SW Forwarding: 2/0/100/0, Other: 0/0/0
  HW Forwarding: 38/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan101 Flags: F NS
    Pkts: 0/0/2    Rate: 0 pps
(*,FF10::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF12::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF20::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF22::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF30::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF32::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF33::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF34::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF35::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF36::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF37::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF38::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF39::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3A::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

(*,FF3B::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3C::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3D::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3E::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3F::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF40::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF42::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF50::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF52::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF60::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF62::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF70::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF72::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF80::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF82::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF90::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF92::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD2::/16) Flags:

```

```

SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
Leaf-01#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on VTEP 1:

```

Leaf-01# show bgp ipv6 mvpn all
BGP table version is 60, local router ID is 172.16.255.3
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: 1:1 (default for vrf green)
* i   [5] [1:1] [FC00:1:102::12] [FF06:1::1]/42
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
* i   [5] [1:1] [FC00:2:255::1] [FF06:1::1]/42
      172.16.255.6          0      100      0 ?
*>i   172.16.255.6          0      100      0 ?
* i   [6] [1:1] [65001] [FC00:2:255::255] [FF06:1::1]/46
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
*>   [7] [1:1] [65001] [FC00:2:255::1] [FF06:1::1]/46
      ::                      32768 ?
Route Distinguisher: 172.16.254.4:102
*>   [7] [172.16.254.4:102] [65001] [FC00:1:102::12] [FF06:1::1]/46
      ::                      32768 ?
Leaf-01#

```

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

```

Leaf-01# 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,
       e - encap-helper tunnel flag
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), 1d05h/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, 1d05h/00:01:34

(172.16.254.3, 239.1.1.1), 00:02:17/00:02:05, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:02:17/00:03:09, A

(172.16.254.6, 239.1.1.1), 00:28:47/00:02:22, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:28:47/00:01:12

(172.16.254.4, 239.1.1.1), 04:45:08/00:01:03, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 04:45:08/00:02:51

(*, 224.0.1.40), 1d05h/00:02:38, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 1d05h/00:02:38

(*, 225.0.0.102), 1d05h/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:01:34

(172.16.254.4, 225.0.0.102), 1d05h/00:02:33, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:00:56

(172.16.254.6, 225.0.0.102), 1d05h/00:01:12, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:01:34

(*, 225.0.0.101), 1d05h/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, 1d05h/00:01:34

(172.16.254.3, 225.0.0.101), 1d05h/00:03:17, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:03:17
Leaf-01#

```

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

```

Leaf-01# 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,
                  e - Encap helper tunnel flag.

```

```

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
GigabitEthernet1/0/2 Flags: A NS
Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/114/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 13/0/127/0, Other: 2/2/0
  HW Forwarding: 12686/0/165/0, Other: 0/0/0
Null0 Flags: A
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2/0/172/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 9299/0/176/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 3817/0/163/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 4/4/0
  HW Forwarding: 15/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 6/5/1
  HW Forwarding: 0/0/0/0, Other: 0/0/0
Null0 Flags: A NS
GigabitEthernet1/0/2 Flags: F
  Pkts: 0/0/0   Rate: 0 pps

```

```

Tunnel4 Flags: F
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding: 8525/0/167/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 2/0/150/0, Other: 0/0/0
  HW Forwarding: 1629/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
Leaf-01#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric](#), on page 340

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI      Type Peer-IP      RMAC/Num_RTs  eVNI      state flags UP time
nve1      50901  L3CP 172.16.254.6  0c75.bd67.ef48 50901      UP  A/-/4 1d05h
nve1      50901  L3CP 172.16.254.3  10b3.d56a.8fc8 50901      UP  A/-/4 1d05h
nve1      50901  L3CP 172.16.254.6  0c75.bd67.ef48 50901      UP  A/M/6 1d05h
nve1      50901  L3CP 172.16.254.3  10b3.d56a.8fc8 50901      UP  A/M/6 1d05h
nve1      10101  L2CP 172.16.254.3  6           10101      UP  N/A   1d05h
nve1      10102  L2CP 172.16.254.6  5           10102      UP  N/A   1d05h
Leaf-02#

```

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

```

Leaf-02# show l2vpn evpn peers vxlan

Interface VNI      Peer-IP      Num routes eVNI      UP time
-----
nve1      10101  172.16.254.3  6          10101     1d05h
nve1      10102  172.16.254.6  5          10102     1d05h
Leaf-02#

```

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on VTEP 2:

```

Leaf-02# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 85, main routing table version 85
6 network entries using 2352 bytes of memory
8 path entries using 1280 bytes of memory
5/5 BGP path/bestpath attribute entries using 1560 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
18 BGP extended community entries using 2396 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 7772 total bytes of memory

```



```
BGP activity 145/47 prefixes, 249/117 paths, scan interval 60 secs
6 networks peaked at 15:50:41 Aug 6 2020 UTC (1d02h ago)
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	2143	2019	85	0	0	1d05h	2
172.16.255.2	4	65001	2139	2019	85	0	0	1d05h	2

Leaf-02#

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

```
Leaf-02# show ip pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: V1901,::FFFF:172.16.254.3
  Info source: Static
  Uptime: 1d05h, Groups: 1
Leaf-02#
```

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

```
Leaf-02# show ip routing vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "bgp 65001", distance 200, metric 0, type internal
  Route count is 1/1, share count 0
  Routing paths:
    172.16.254.3%default, Vlan901%default
    From AC10:FF01::
    opaque_ptr 0x7F65BA333EC0
    Last updated 04:26:58 ago
Leaf-02#
```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on VTEP 2:

```
Leaf-02# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
  Uptime    Expires
FF06:1::1                               Vlan102
  1d05h    00:03:53
Leaf-02#
```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on VTEP 2:

```
Leaf-02# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
  C - Connected, L - Local, I - Received Source Specific Host Report,
  P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
  J - Join SPT, Y - Joined MDT-data group,
  y - Sending to MDT-data group
  g - BGP signal originated, G - BGP Signal received,
  N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
  q - BGP Src-Active originated, Q - BGP Src-Active received
```

```

E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 1d05h/never, RP FC00:2:255::255, flags: SCJg
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.3
  Immediate Outgoing interface list:
    Vlan102, Forward, 1d05h/never

(FC00:1:102::12, FF06:1::1), 00:07:24/00:03:28, flags: SFJTGq
  Incoming interface: Vlan102
  RPF nbr: FE80::46D3:CAFF:FE28:6CC5
  Immediate Outgoing interface list:
    Vlan901, Forward, 00:07:24/never

(FC00:2:255::1, FF06:1::1), 00:06:31/00:00:32, flags: SJTgQ
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.6
  Inherited Outgoing interface list:
    Vlan102, Forward, 1d05h/never
Leaf-02#

```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on VTEP 2:

```

Leaf-02# show ipv6 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,
             e - Encap helper tunnel flag.
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
(*,FF00::/8) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF00::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF02::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 10/10/0
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 3/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A NS
  Vlan102 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(FC00:1:102::12,FF06:1::1) Flags: HW
  SW Forwarding: 2/0/100/0, Other: 0/0/0
  HW Forwarding: 217/0/118/0, Other: 0/0/0
  Vlan102 Flags: A F
    Pkts: 0/0/0   Rate: 0 pps

```

```

Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
  Pkts: 0/0/1   Rate: 0 pps
(FC00:2:255::1,FF06:1::1) Flags: HW
SW Forwarding: 2/0/100/0, Other: 0/0/0
HW Forwarding: 191/0/126/0, Other: 0/0/0
Vlan901, VXLAN Decap Flags: A
Vlan102 Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
(*,FF10::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF12::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF20::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF22::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF30::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF32::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF33::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF34::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF35::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF36::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF37::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF38::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF39::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3A::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3B::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3C::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3D::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3E::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3F::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF40::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF42::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF50::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF52::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF60::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF62::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF70::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF72::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF80::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF82::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF90::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF92::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
Leaf-02#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on VTEP 2:

```

Leaf-02# show bgp ipv6 mvpn all
BGP table version is 85, local router ID is 172.16.255.4
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: 1:1 (default for vrf green)
*> [5] [1:1] [FC00:1:102::12] [FF06:1::1]/42
      ::
      32768 ?
* i [5] [1:1] [FC00:2:255::1] [FF06:1::1]/42
      172.16.255.6          0    100    0 ?
*>i          172.16.255.6          0    100    0 ?
*> [6] [1:1] [65001] [FC00:2:255::255] [FF06:1::1]/46
      ::
      32768 ?
*>i [7] [1:1] [65001] [FC00:1:102::12] [FF06:1::1]/46
      172.16.255.3          0    100    0 ?
*> [7] [1:1] [65001] [FC00:2:255::1] [FF06:1::1]/46
      ::
      32768 ?
Route Distinguisher: 172.16.254.4:102
* i [7] [172.16.254.4:102] [65001] [FC00:1:102::12] [FF06:1::1]/46
      172.16.255.3          0    100    0 ?
*>i          172.16.255.3          0    100    0 ?
Leaf-02#

```

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

```

Leaf-02# 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,
       e - encap-helper tunnel flag
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), 1d05h/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, 1d05h/00:02:00

(172.16.254.6, 239.1.1.1), 00:33:54/00:01:36, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:33:54/00:02:05

(172.16.254.4, 239.1.1.1), 04:50:15/00:03:03, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0

```

```

Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 04:50:15/00:03:29

(*, 224.0.1.40), 1d05h/00:02:01, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 1d05h/00:02:01

(*, 225.0.0.102), 1d05h/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, 1d05h/00:02:00

(172.16.254.6, 225.0.0.102), 1d05h/00:02:05, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:02:00

(172.16.254.4, 225.0.0.102), 1d05h/00:02:29, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d05h/00:03:28

(*, 225.0.0.101), 1d05h/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:02:00

(172.16.254.3, 225.0.0.101), 1d05h/00:01:04, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d05h/00:02:00
Leaf-02#

```

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

```

Leaf-02# 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,
              e - Encap helper tunnel flag.
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
  GigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS

```

```

    Pkts: 0/0/0    Rate: 0 pps
(*,225.0.0.101) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 2/0/170/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0    Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
SW Forwarding: 1/0/150/0, Other: 0/0/0
HW Forwarding: 12630/0/177/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1    Rate: 0 pps
(*,225.0.0.102) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 1/0/224/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0    Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
SW Forwarding: 2/0/163/0, Other: 3/1/2
HW Forwarding: 9373/0/164/0, Other: 0/0/0
Null0 Flags: A
GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/1    Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 1/0/206/0, Other: 0/0/0
HW Forwarding: 3825/0/163/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1    Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 9/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0    Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 24/20/4
HW Forwarding: 8667/0/156/0, Other: 0/0/0
Null0 Flags: A
GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/1    Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 2/0/150/0, Other: 0/0/0
HW Forwarding: 1781/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/2    Rate: 0 pps
Leaf-02#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric](#), on page 340

Outputs to Verify the Configuration on Border VTEP

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

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

Border# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901   L3CP 172.16.254.3    10b3.d56a.8fc8 50901      UP  A/-/4 1d06h
nve1     50901   L3CP 172.16.254.4    7c21.0dbd.9548 50901      UP  A/-/4 1d06h
nve1     50901   L3CP 172.16.254.3    10b3.d56a.8fc8 50901      UP  A/M/6 1d06h
nve1     50901   L3CP 172.16.254.4    7c21.0dbd.9548 50901      UP  A/M/6 1d06h
nve1     10101   L2CP 172.16.254.3     6              10101      UP  N/A   1d06h
nve1     10102   L2CP 172.16.254.4     7              10102      UP  N/A   1d05h
Border#

```

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

```

Border# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1     10101   172.16.254.3    6          10101    1d06h
nve1     10102   172.16.254.4    7          10102    1d05h
Border#

```

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Border VTEP:

```

Border# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.6, local AS number 65001
BGP table version is 85, main routing table version 85
5 network entries using 1960 bytes of memory
7 path entries using 1120 bytes of memory
5/5 BGP path/bestpath attribute entries using 1560 bytes of memory
4 BGP rinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
20 BGP extended community entries using 2706 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 7530 total bytes of memory
BGP activity 137/41 prefixes, 272/148 paths, scan interval 60 secs
5 networks peaked at 15:42:39 Aug 6 2020 UTC (1d02h ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   2158   2031     85    0    0 1d06h    2
172.16.255.2  4      65001   2157   2025     85    0    0 1d06h    2
Border#

```

The following example shows the output for the **show ip pim vrf vrf-name group-map** command on Border VTEP:

```

Border# show ip pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: V1901,::FFFF:172.16.254.3
  Info source: Static
  Uptime: 1d06h, Groups: 1
Border#

```

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


```

Border# show ip routing vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "bgp 65001", distance 200, metric 0, type internal
  Redistributing via ospf 1
  Route count is 1/1, share count 0
  Routing paths:
    172.16.254.3%default, Vlan901%default
      From AC10:FF01::
        opaque_ptr 0x7FEF699AEC28
        Last updated 04:34:38 ago
Border#

```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on Border VTEP:

```

Border# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
  Uptime      Expires
FF06:1::1                                       Vlan102
  1d05h      00:02:29
Border#

```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on Border VTEP:

```

Border# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 1d05h/00:02:52, RP FC00:2:255::255, flags: SCg
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.3
  Immediate Outgoing interface list:
    Vlan102, Null, 1d05h/never
    Vlan2001, Forward, 04:34:39/00:02:52

(FC00:1:102::12, FF06:1::1), 00:15:05/00:02:32, flags: STgQ
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.4
  Immediate Outgoing interface list:
    Vlan2001, Forward, 00:15:03/00:02:32
  Inherited Outgoing interface list:
    Vlan102, Null, 1d05h/never

(FC00:2:255::1, FF06:1::1), 00:14:13/00:02:52, RP FC00:2:255::255, flags: SPR
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.3
  Immediate Outgoing interface list:
    Vlan2001, Null, 00:14:13/00:02:52
  Inherited Outgoing interface list:
    Vlan102, Null, 1d05h/never

```

```
(FC00:2:255::1, FF06:1::1), 00:14:12/never, flags: STGq
Incoming interface: Vlan2001
RPF nbr: FE80::A2B4:39FF:FE21:9183
Immediate Outgoing interface list:
  Vlan901, Forward, 00:14:12/never
Inherited Outgoing interface list:
  Vlan102, Null, 1d05h/never
Border#
```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on Border VTEP:

```
Border# show ipv6 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,
             e - Encap helper tunnel flag.
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
(*,FF00::/8) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF00::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF02::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 12/12/0
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 4/4/0
  HW Forwarding: 7/0/122/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan2001 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(FC00:1:102::12,FF06:1::1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 450/0/125/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan2001 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(FC00:2:255::1,FF06:1::1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 2/1/1
  HW Forwarding: 423/0/122/0, Other: 0/0/0
  Vlan2001 Flags: A
  Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
    Pkts: 0/0/0   Rate: 0 pps
(*,FF10::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF12::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF20::/15) Flags: HW
```

```

SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF22::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF30::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF32::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF33::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF34::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF35::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF36::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF37::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF38::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF39::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3A::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3B::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3C::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3D::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3E::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3F::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF40::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF42::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF50::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF52::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF60::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF62::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF70::/15) Flags: HW

```

```

SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF72::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF80::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF82::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF90::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF92::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
Border#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Border VTEP:

```

Border# show bgp ipv6 mvpn all
BGP table version is 85, local router ID is 172.16.255.6
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: 1:1 (default for vrf green)
* i   [5][1:1][FC00:1:102::12][FF06:1::1]/42
                172.16.255.4          0    100    0 ?

```

```

*>i          172.16.255.4          0    100    0 ?
*>  [5] [1:1] [FC00:2:255::1] [FF06:1::1]/42
      ::                               32768 ?
*>  [6] [1:1] [65001] [FC00:2:255::255] [FF06:1::1]/46
      ::                               32768 ?
* i  [7] [1:1] [65001] [FC00:2:255::1] [FF06:1::1]/46
      172.16.255.3          0    100    0 ?
*>i          172.16.255.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:102
*>  [7] [172.16.254.4:102] [65001] [FC00:1:102::12] [FF06:1::1]/46
      ::                               32768 ?
Border#

```

The following example shows the output for the **show ip mroute** command on Border VTEP:

```

Border# 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,
      e - encap-helper tunnel flag
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), 1d06h/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d06h/00:00:08

(172.16.254.6, 239.1.1.1), 00:41:35/00:02:45, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/2, Forward/Sparse, 00:41:35/00:03:11

(172.16.254.4, 239.1.1.1), 04:57:56/00:02:37, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 04:57:56/00:02:03

(*, 224.0.1.40), 1d06h/00:02:10, RP 172.16.255.255, flags: SJCL
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 1d06h/00:02:10

(*, 225.0.0.102), 1d06h/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d06h/00:00:08

(172.16.254.4, 225.0.0.102), 1d06h/00:01:56, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:

```

```

Tunnel0, Forward/Sparse, 1d06h/00:00:08

(172.16.254.6, 225.0.0.102), 1d06h/00:02:16, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0
Outgoing interface list:
  TenGigabitEthernet1/0/2, Forward/Sparse, 1d06h/00:03:21, A

(*, 225.0.0.101), 1d06h/stopped, RP 172.16.255.255, flags: SJCx
Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
Outgoing interface list:
  Tunnel0, Forward/Sparse, 1d06h/00:00:08

(172.16.254.3, 225.0.0.101), 1d06h/00:02:00, flags: JTx
Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
Outgoing interface list:
  Tunnel0, Forward/Sparse, 1d06h/00:00:20
Border#

```

The following example shows the output for the **show ip mfib** command on Border VTEP:

```

Border# 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,
             e - Encap helper tunnel flag.
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
  TenGigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 8/0/146/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 7/0/125/0, Other: 0/0/0
  HW Forwarding: 12768/0/177/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/7   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/172/0, Other: 0/0/0

```

```

TenGigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0    Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
SW Forwarding: 1/0/154/0, Other: 0/0/0
HW Forwarding:  9363/0/176/0, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1    Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 17/0/174/0, Other: 10/9/1
HW Forwarding:  3858/0/151/0, Other: 0/0/0
Null0 Flags: A
TenGigabitEthernet1/0/2 Flags: F
  Pkts: 0/0/16   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding:  10/0/168/0, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0    Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 0/0/0
HW Forwarding:  8909/0/167/0, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1    Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 2/0/150/0, Other: 2/2/0
HW Forwarding:  2018/0/156/0, Other: 0/0/0
Null0 Flags: A
TenGigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0    Rate: 0 pps
Border#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric](#), on page 340

Outputs to Verify the Configuration on Spine Switch 1

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Spine Switch 1:

```

Spine-01# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 78, main routing table version 78
5 network entries using 1960 bytes of memory
13 path entries using 2080 bytes of memory
4/4 BGP path/bestpath attribute entries using 1216 bytes of memory
3 BGP rinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2356 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 7756 total bytes of memory
BGP activity 270/220 prefixes, 3041/2934 paths, scan interval 60 secs
5 networks peaked at 15:48:28 Aug 6 2020 UTC (1d02h ago)

Neighbor          V           AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

172.16.255.2 4 65001 2540 2489 78 0 0 1d09h 5
172.16.255.3 4 65001 2020 2157 78 0 0 1d06h 2
172.16.255.4 4 65001 2030 2154 78 0 0 1d06h 3
172.16.255.6 4 65001 2033 2160 78 0 0 1d06h 3
Spine-01#

```

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 (?)
Spine-01#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Spine Switch 1:

```

Spine-01# show bgp ipv6 mvpn all
BGP table version is 78, local router ID is 172.16.255.1
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: 1:1
 * i [5][1:1][FC00:1:102::12][FF06:1::1]/42
   172.16.255.4          0 100 0 ?
 *>i 172.16.255.4          0 100 0 ?
 * i [5][1:1][FC00:2:255::1][FF06:1::1]/42
   172.16.255.6          0 100 0 ?
 *>i 172.16.255.6          0 100 0 ?
 * i [6][1:1][65001][FC00:2:255::255][FF06:1::1]/46
   172.16.255.6          0 100 0 ?
 * i 172.16.255.4          0 100 0 ?
 *>i 172.16.255.4          0 100 0 ?
 * i [7][1:1][65001][FC00:2:255::1][FF06:1::1]/46
   172.16.255.4          0 100 0 ?
 * i 172.16.255.3          0 100 0 ?
 *>i 172.16.255.3          0 100 0 ?
Route Distinguisher: 172.16.254.4:102
 * i [7][172.16.254.4:102][65001][FC00:1:102::12][FF06:1::1]/46
   172.16.255.6          0 100 0 ?
 * i 172.16.255.3          0 100 0 ?
 *>i 172.16.255.3          0 100 0 ?
Spine-01#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 1:

```

Spine-01# 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), 04:59:49/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(172.16.254.6, 239.1.1.1), 00:43:26/00:02:24, flags: PA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null

(172.16.254.4, 239.1.1.1), 04:57:47/00:01:01, flags: PTA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null

(*, 224.0.1.40), 1w0d/00:02:32, RP 172.16.255.255, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Sparse, 1w0d/00:02:32

(*, 225.0.0.102), 1w0d/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(172.16.254.6, 225.0.0.102), 02:11:00/00:01:54, flags: PA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null

(172.16.254.4, 225.0.0.102), 1d06h/00:02:27, flags: PA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null
Spine-01#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 1:

```

Spine-01# 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: 83/83/0
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
Tunnell Flags: A
Loopback2 Flags: F IC NS
Pkts: 0/0/0 Rate: 0 pps
(*,225.0.0.102) Flags: C HW
SW Forwarding: 1/0/206/0, Other: 282/0/282
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
(172.16.254.4,225.0.0.102) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/2 Flags: NS
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/4 Flags: NS
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 6/5/1
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 8/0/157/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/4 Flags: NS
Spine-01#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric, on page 340](#)

Outputs to Verify the Configuration on Spine Switch 2

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Spine Switch 2:

```

Spine-02# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 77, main routing table version 77
5 network entries using 1960 bytes of memory
13 path entries using 2080 bytes of memory
4/4 BGP path/bestpath attribute entries using 1216 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2356 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 7756 total bytes of memory
BGP activity 301/251 prefixes, 3143/3036 paths, scan interval 60 secs
5 networks peaked at 15:49:16 Aug 6 2020 UTC (1d02h ago)

```

```

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   2491   2541     77   0    0 1d09h    5
172.16.255.3  4      65001   2021   2155     77   0    0 1d06h    2
172.16.255.4  4      65001   2031   2152     77   0    0 1d06h    3
172.16.255.6  4      65001   2029   2161     77   0    0 1d06h    3
Spine-02#

```

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 (?)
Spine-02#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Spine Switch 2:

```

Spine-02# show bgp ipv6 mvpn all
BGP table version is 77, local router ID is 172.16.255.2
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: 1:1
* i   [5][1:1][FC00:1:102::12][FF06:1::1]/42
      172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
* i   [5][1:1][FC00:2:255::1][FF06:1::1]/42
      172.16.255.6          0      100      0 ?
*>i   172.16.255.6          0      100      0 ?
* i   [6][1:1][65001][FC00:2:255::255][FF06:1::1]/46
      172.16.255.6          0      100      0 ?
* i   172.16.255.4          0      100      0 ?
*>i   172.16.255.4          0      100      0 ?
* i   [7][1:1][65001][FC00:2:255::1][FF06:1::1]/46
      172.16.255.4          0      100      0 ?
* i   172.16.255.3          0      100      0 ?
*>i   172.16.255.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:102
* i   [7][172.16.254.4:102][65001][FC00:1:102::12][FF06:1::1]/46
      172.16.255.6          0      100      0 ?
* i   172.16.255.3          0      100      0 ?
*>i   172.16.255.3          0      100      0 ?
Spine-02#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 2:

```

Spine-02# 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,

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric

```

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), 3d13h/00:03:01, RP 172.16.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 1d06h/00:03:01
  GigabitEthernet1/0/1, Forward/Sparse, 1d06h/00:02:41
  GigabitEthernet1/0/4, Forward/Sparse, 1d06h/00:02:43

(172.16.254.6, 239.1.1.1), 00:44:52/00:02:29, flags: T
Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
Outgoing interface list:
  GigabitEthernet1/0/1, Forward/Sparse, 00:44:52/00:02:58
  GigabitEthernet1/0/2, Forward/Sparse, 00:44:52/00:03:02

(172.16.254.4, 239.1.1.1), 05:01:13/00:02:28, flags: T
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
Outgoing interface list:
  GigabitEthernet1/0/4, Forward/Sparse, 05:01:13/00:03:22
  GigabitEthernet1/0/1, Forward/Sparse, 05:01:13/00:02:56

(*, 224.0.1.40), 1w0d/00:03:12, RP 172.16.255.255, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 1d06h/00:03:12
  GigabitEthernet1/0/1, Forward/Sparse, 1d06h/00:02:54
  GigabitEthernet1/0/4, Forward/Sparse, 1d06h/00:02:44
  Loopback2, Forward/Sparse, 1w0d/00:02:32

(*, 225.0.0.102), 1w0d/00:03:26, RP 172.16.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 1d06h/00:03:21
  GigabitEthernet1/0/1, Forward/Sparse, 1d06h/00:03:26
  GigabitEthernet1/0/4, Forward/Sparse, 1d06h/00:02:56

(172.16.254.4, 225.0.0.102), 1d06h/00:02:18, flags: MT
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
Outgoing interface list:
  GigabitEthernet1/0/4, Forward/Sparse, 1d06h/00:03:15
  GigabitEthernet1/0/1, Forward/Sparse, 1d06h/00:03:26

(172.16.254.6, 225.0.0.102), 1d06h/00:02:40, flags: MT
Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 1d06h/00:03:28
  GigabitEthernet1/0/1, Forward/Sparse, 1d06h/00:03:28

(*, 225.0.0.101), 3d13h/00:03:13, RP 172.16.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 1d06h/00:02:59

```

```

GigabitEthernet1/0/1, Forward/Sparse, 1d06h/00:02:53
GigabitEthernet1/0/4, Forward/Sparse, 1d06h/00:03:13

(172.16.254.3, 225.0.0.101), 1d06h/00:03:09, flags: TA
Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
Outgoing interface list:
GigabitEthernet1/0/2, Forward/Sparse, 1d06h/00:03:27
GigabitEthernet1/0/4, Forward/Sparse, 1d06h/00:03:13
Spine-02#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 2:

```

Spine-02# 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
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
Loopback2 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 9/0/112/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 4/0/132/0, Other: 0/0/0
  HW Forwarding: 12790/0/177/0, Other: 0/0/0
GigabitEthernet1/0/1 Flags: A
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps

```

```

(*,225.0.0.102) Flags: C HW
  SW Forwarding: 27/0/101/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel1 Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 9381/0/176/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 3853/0/163/0, Other: 0/0/0
  GigabitEthernet1/0/4 Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 10/0/150/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel1 Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/4 Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/4 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/4 Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 9007/0/167/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2111/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/4 Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
Spine-02#

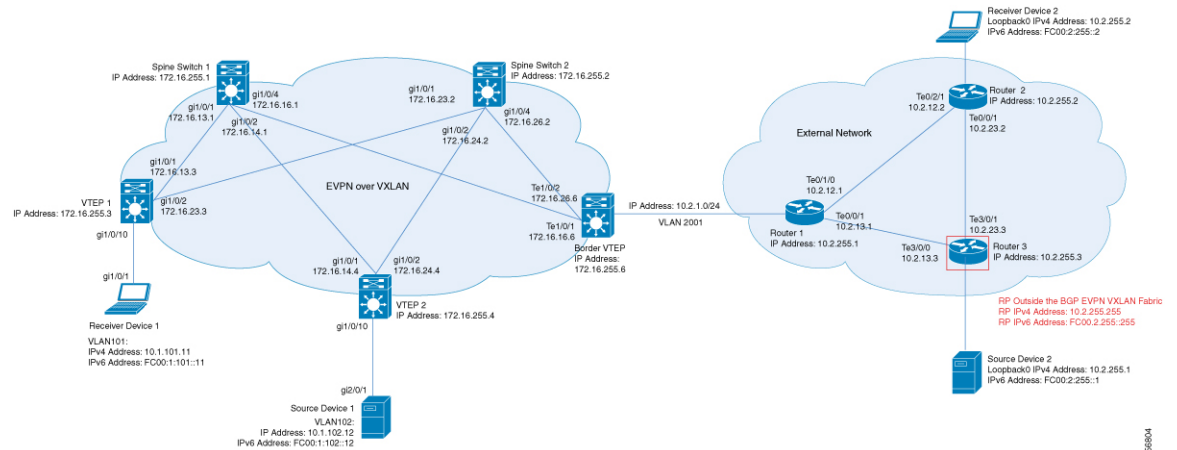
```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Inside the BGP EVPN VXLAN Fabric](#), on page 340

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

This example shows how to configure and verify Layer 3 TRM with PIM-SM for IPv4 multicast traffic when the RP is outside the BGP EVPN VXLAN fabric. The example uses the following topology:

Figure 25: TRM with PIM-SM when the RP is Outside the BGP EVPN VXLAN Fabric



The topology shows an EVPN VXLAN network, with two spine switches and three VTEPs, connected to an external network with three routers. Router 3 in the external network acts as the RP in this topology and Border VTEP connects the fabric to the external network through Router 1. The IPv4 multicast group is 226.1.1.1 for this topology. The following tables provide sample configurations for the devices in this topology:

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Table 37: Configuring VTEP 1, Border VTEP, and VTEP 2 to Configure TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! 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 ! system mtu 9198 ! 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 pim sparse-mode 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> Border# show running-config hostname Border ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 ! vlan 2001 ! 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 ! </pre>	<pre> Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp route-target export 1:1 route-target import 1:1 route-target export 1:1 stitching route-target import 1:1 stitching exit-address-famil ! ip routing ! ip multicast-routing ip multicast-routing vrf green ! 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 ! system mtu 9198 ! 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 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>

VTEP 1	Border VTEP	VTEP 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 access vlan 101 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! </pre>	<pre> 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 ! interface TenGigabitEthernet1/0/2 no switchport ip address 172.16.26.6 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/5 switchport trunk allowed vlan 2001 switchport mode trunk ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode no autostate ! interface Vlan2001 vrf forwarding green ip address 10.2.1.1 255.255.255.0 ip mtu 1500 ip pim sparse-mode ip ospf network point-to-point ip ospf 2 area 0 ! 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 mcast-group 225.0.0.102 ! router ospf 2 vrf green redistribute bgp 65001 </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 access vlan 102 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! </pre>

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> address-family ipv4 mvpn 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 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 ! ip pim rp-address 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! end ! Leaf-01# </pre>	<pre> ! router ospf 1 router-id 172.16.255.6 ! 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 ipv4 mvpn 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 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 redistribute ospf 2 match internal external 1 external 2 exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! end ! Border# </pre>	<pre> address-family ipv4 mvpn 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 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 ! ip pim rp-address 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! end ! Leaf-02# </pre>

Table 38: Configuring Spine Switch 1 and Spine Switch 2 to Configure TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 GigabitEthernet1/0/4 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 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 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 ! interface GigabitEthernet1/0/4 no switchport ip address 172.16.26.2 255.255.255.0 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! </pre>

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-01# </pre>	<pre> address-family ipv4 exit-address-family ! address-family ipv4 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-02# </pre>

Table 39: Configuring Router 1, Router 2, and Router 3 to Configure TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Router 1	Router 2	Router 3
<pre> R1# show running-config hostname R1 ! ip multicast-routing distributed ! interface Loopback0 ip address 10.2.255.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet0/0/1.2001 encapsulation dot1Q 2001 ip address 10.2.1.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 router ospf 1 router-id 10.2.255.1 ! ip pim rp-address 10.2.255.255 ! end ! R1# </pre>	<pre> R2# show running-config hostname R2 ! ip multicast-routing distributed ! interface Loopback0 ip address 10.2.255.2 255.255.255.255 ip pim sparse-mode ip igmp join-group 226.1.1.1 ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 10.2.255.2 ! ip pim rp-address 10.2.255.255 ! end ! R2# </pre>	<pre> R3# show running-config hostname R3 ! ip multicast-routing distributed ! interface Loopback0 ip address 10.2.255.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface Loopback255 ip address 10.2.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! router ospf 1 router-id 10.2.255.3 ! ip pim rp-address 10.2.255.255 ! end ! R3# </pre>

Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

The following sections provide sample outputs for **show** commands to verify TRM with PIM-SM on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 378](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 383](#)
- [Outputs to Verify the Configuration on Border VTEP, on page 388](#)
- [Outputs to Verify the Configuration on Spine Switch 1, on page 394](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 397](#)
- [Outputs to Verify the Configuration on Router 3 \(RP Outside the BGP EVPN VXLAN Fabric\), on page 401](#)

Outputs to Verify the Configuration on VTEP 1

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

```
Leaf-01# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.6    0c75.bd67.ef48 50901     UP  A/-/4 16:44:02
nve1     50901    L3CP 172.16.254.4     7c21.0dbd.9548 50901     UP  A/-/4 16:41:00
nve1     50901    L3CP 172.16.254.6     0c75.bd67.ef48 50901     UP  A/M/6 16:44:02
nve1     50901    L3CP 172.16.254.4     7c21.0dbd.9548 50901     UP  A/M/6 16:41:00
nve1     10102    L2CP 172.16.254.4     7                10102     UP  N/A   16:23:05
nve1     10102    L2CP 172.16.254.6     5                10102     UP  N/A   16:44:02
Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time
-----
nve1     10102    172.16.254.4    7         10102    16:23:06
nve1     10102    172.16.254.6    5         10102    16:44:02
Leaf-01#
```

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on VTEP 1:

```
Leaf-01# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 58, main routing table version 58
6 network entries using 1824 bytes of memory
8 path entries using 1088 bytes of memory
3/3 BGP path/bestpath attribute entries using 936 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2372 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 6404 total bytes of memory
BGP activity 117/25 prefixes, 240/113 paths, scan interval 60 secs
6 networks peaked at 12:17:52 Aug 6 2020 UTC (16:27:28.286 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   1217   1126    58    0    0 16:44:58      2
172.16.255.2  4      65001   1213   1121    58    0    0 16:44:53      2
Leaf-01#
```

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

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

Group(s): 224.0.0.0/4, Static
          RP: 10.2.255.255 (?)
Leaf-01#
```

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

```
Leaf-01# show ip routing vrf green 10.2.255.255
Routing Table: green
Routing entry for 10.2.255.255/32
  Known via "bgp 65001", distance 200, metric 3, type internal
  Last update from 172.16.254.6 on Vlan901, 16:17:01 ago
Routing Descriptor Blocks:
  * 172.16.254.6 (default), from 172.16.255.1, 16:17:01 ago, via Vlan901
    opaque_ptr 0x7FBB8620D990
    Route metric is 3, traffic share count is 1
    AS Hops 0
    MPLS label: none
Leaf-01#
```

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

```
Leaf-01# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter  Group Accounted
226.1.1.1          Vlan101           13:03:08  00:02:13  10.1.101.11
224.0.1.40         Loopback901       16:45:17  00:02:50  10.1.255.1
Leaf-01#
```

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

```
Leaf-01# 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,
       e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.1.1.1), 13:03:08/stopped, RP 10.2.255.255, flags: SJCg
  Incoming interface: Vlan901, RPF nbr 172.16.254.6
  Outgoing interface list:
    Vlan101, Forward/Sparse, 13:03:08/00:02:13

(10.2.255.1, 226.1.1.1), 00:08:23/00:02:54, flags: TgQ
  Incoming interface: Vlan901, RPF nbr 172.16.254.6
  Outgoing interface list:
    Vlan101, Forward/Sparse, 00:08:23/00:02:13

(10.1.102.12, 226.1.1.1), 00:08:48/00:02:24, flags: TgQ
  Incoming interface: Vlan901, RPF nbr 172.16.254.4
  Outgoing interface list:
    Vlan101, Forward/Sparse, 00:08:48/00:02:13

(*, 224.0.1.40), 16:45:17/00:02:50, RP 10.2.255.255, flags: SJCLg
```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```
Incoming interface: Vlan901, RPF nbr 172.16.254.6
Leaf-01#
```

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

```
Leaf-01# 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,
             e - Encap helper tunnel flag.
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: 2/2/0
  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
  Vlan901, VXLAN Decap Flags: A NS
  Loopback901 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,226.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A NS
  Vlan101 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(10.1.102.12,226.1.1.1) Flags: HW
  SW Forwarding: 5/0/100/0, Other: 0/0/0
  HW Forwarding: 523/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
Leaf-01#
```

The following example shows the output for the **show bgp ipv4 mvpn all** command on VTEP 1:

```
Leaf-01# show bgp ipv4 mvpn all
BGP table version is 60, local router ID is 172.16.255.6
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: 1:1 (default for vrf green)
*>i  [5][1:1][10.1.102.12][226.1.1.1]/18
      172.16.255.4          0      100      0 ?
* i      172.16.255.4          0      100      0 ?
*>  [5][1:1][10.2.255.1][226.1.1.1]/18
      0.0.0.0                32768 ?
```



```

* i [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      172.16.255.3          0    100    0 ?
*>i      172.16.255.3          0    100    0 ?
* i [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      172.16.255.3          0    100    0 ?
*>i      172.16.255.3          0    100    0 ?
*>i [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      172.16.255.3          0    100    0 ?
* i      172.16.255.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:102
*> [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      0.0.0.0                32768 ?
Leaf-01#

```

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

```

Leaf-01# 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,
      e - encap-helper tunnel flag
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), 16:45:08/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 16:45:08/00:00:45

(172.16.254.4, 239.1.1.1), 00:08:47/00:01:59, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:08:47/00:00:12

(172.16.254.6, 239.1.1.1), 00:08:49/00:02:00, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:08:49/00:00:10

(*, 224.0.1.40), 16:45:17/00:02:46, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
Leaf-01#

```

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

```

Leaf-01# 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

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

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,
e - Encap helper tunnel flag.
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
  GigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  1/0/114/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 13/0/127/0, Other: 2/2/0
  HW Forwarding:  7870/0/164/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  2/0/172/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding:  5222/0/176/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding:  2137/0/163/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  11/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 4/0/150/0, Other: 0/0/0

```

```

HW Forwarding: 518/0/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0/4 Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 0/0/0
HW Forwarding: 498/1/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0/1 Rate: 0 pps
Leaf-01#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 377](#)

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI Type Peer-IP RMAC/Num_RTs eVNI state flags UP time
nve1 50901 L3CP 172.16.254.6 0c75.bd67.ef48 50901 UP A/-/4 16:56:53
nve1 50901 L3CP 172.16.254.3 10b3.d56a.8fc8 50901 UP A/-/4 16:56:53
nve1 50901 L3CP 172.16.254.6 0c75.bd67.ef48 50901 UP A/M/6 16:56:53
nve1 50901 L3CP 172.16.254.3 10b3.d56a.8fc8 50901 UP A/M/6 16:56:53
nve1 10101 L2CP 172.16.254.3 6 10101 UP N/A 16:56:53
nve1 10102 L2CP 172.16.254.6 5 10102 UP N/A 16:56:53
Leaf-02#

```

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

```

Leaf-02# show l2vpn evpn peers vxlan
Interface VNI Peer-IP Num routes eVNI UP time
-----
nve1 10101 172.16.254.3 6 10101 16:56:54
nve1 10102 172.16.254.6 5 10102 16:56:54
Leaf-02#

```

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on VTEP 2:

```

Leaf-02# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 62, main routing table version 62
7 network entries using 2128 bytes of memory
9 path entries using 1224 bytes of memory
4/4 BGP path/bestpath attribute entries using 1248 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2372 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 7156 total bytes of memory
BGP activity 121/28 prefixes, 202/77 paths, scan interval 60 secs
9 networks peaked at 12:22:24 Aug 6 2020 UTC (16:43:21.423 ago)

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
172.16.255.1 4 65001 1229 1151 62 0 0 16:57:50 2

```

```
172.16.255.2    4      65001    1227    1152      62     0     0 16:57:51      2
Leaf-02#
```

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

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

Group(s): 224.0.0.0/4, Static
          RP: 10.2.255.255 (?)
Leaf-02#
```

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

```
Leaf-02# show ip routing vrf green 10.2.255.255
Routing Table: green
Routing entry for 10.2.255.255/32
  Known via "bgp 65001", distance 200, metric 3, type internal
  Last update from 172.16.254.6 on Vlan901, 16:56:55 ago
  Routing Descriptor Blocks:
  * 172.16.254.6 (default), from 172.16.255.1, 16:56:55 ago, via Vlan901
    opaque_ptr 0x7F65B8B9E4B0
    Route metric is 3, traffic share count is 1
    AS Hops 0
    MPLS label: none
Leaf-02#
```

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

```
Leaf-02# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address      Interface      Uptime      Expires      Last Reporter      Group Accounted
226.1.1.1          Vlan102       16:58:00    00:02:11    10.1.102.12
224.0.1.40         Vlan901       16:58:37    00:02:33    172.16.254.4
Leaf-02#
```

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

```
Leaf-02# 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,
       e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
```

```

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.1.1.1), 16:58:00/stopped, RP 10.2.255.255, flags: SJCFg
  Incoming interface: Vlan901, RPF nbr 172.16.254.6
  Outgoing interface list:
    Vlan102, Forward/Sparse, 16:58:00/00:02:11

(10.2.255.1, 226.1.1.1), 00:24:16/00:02:40, flags: JTgQ
  Incoming interface: Vlan901, RPF nbr 172.16.254.6
  Outgoing interface list:
    Vlan102, Forward/Sparse, 00:24:16/00:02:11

(10.1.102.12, 226.1.1.1), 00:24:41/00:02:09, flags: FTGqX
  Incoming interface: Vlan102, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    Vlan901, Forward/Sparse, 00:24:41/stopped

(*, 224.0.1.40), 16:58:37/00:02:33, RP 10.2.255.255, flags: SJPLGx
  Incoming interface: Vlan901, RPF nbr 172.16.254.6
  Outgoing interface list: Null
Leaf-02#

```

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

```

Leaf-02# 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,
               e - Encap helper tunnel flag.
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
  Vlan901, VXLAN Decap Flags: A IC NS
(*,226.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 3/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A NS
  Vlan102 Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(10.1.102.12,226.1.1.1) Flags: HW
  SW Forwarding: 739/0/100/0, Other: 2/2/0
  HW Forwarding: 736/0/118/0, Other: 0/0/0
  Vlan102 Flags: A
  Tunnel5 Flags: F
    Pkts: 0/0/739   Rate: 0 pps
  Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```
Pkts: 0/0/739   Rate: 0 pps
Leaf-02#
```

The following example shows the output for the **show bgp ipv4 mvpn all** command on VTEP 2:

```
Leaf-02# show bgp ipv4 mvpn all
BGP table version is 62, local router ID is 172.16.255.4
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: 1:1 (default for vrf green)
*> [5][1:1][10.1.102.12][226.1.1.1]/18
      0.0.0.0
      32768 ?
*>i [5][1:1][10.2.255.1][226.1.1.1]/18
      172.16.255.6
      0 100 0 ?
* i 172.16.255.6
      0 100 0 ?
*> [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      0.0.0.0
      32768 ?
*> [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      0.0.0.0
      32768 ?
*>i [7][1:1][65001][10.1.102.12/32][226.1.1.1/32]/22
      172.16.255.3
      0 100 0 ?
*> [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      0.0.0.0
      32768 ?
Route Distinguisher: 172.16.254.4:102
*>i [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      172.16.255.3
      0 100 0 ?
* i 172.16.255.3
      0 100 0 ?
Leaf-02#
```

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

```
Leaf-02# 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,
       e - encap-helper tunnel flag
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), 16:58:28/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, 16:58:28/00:02:25
```

```
(172.16.254.6, 239.1.1.1), 00:24:42/00:00:58, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:24:42/00:02:17

(172.16.254.4, 239.1.1.1), 00:24:42/00:03:28, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:24:42/00:03:23, A

(*, 224.0.1.40), 16:58:37/00:02:26, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 16:58:36/00:02:26
Leaf-02#
```

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

```
Leaf-02# 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,
                  e - Encap helper tunnel flag.
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
  GigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2/0/170/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding: 7870/0/176/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/224/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

Pkts: 0/0/0    Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
SW Forwarding: 2/0/163/0, Other: 3/1/2
HW Forwarding: 5353/0/164/0, Other: 0/0/0
Null0 Flags: A
GigabitEthernet1/0/2 Flags: F NS
Pkts: 0/0/1    Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
SW Forwarding: 1/0/206/0, Other: 0/0/0
HW Forwarding: 2165/0/163/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0/1    Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 5/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0/0    Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 1495/1491/4
HW Forwarding: 742/0/156/0, Other: 0/0/0
Null0 Flags: A NS
GigabitEthernet1/0/2 Flags: F
Pkts: 0/0/1    Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 1/0/150/0, Other: 0/0/0
HW Forwarding: 1460/1/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0/1    Rate: 0 pps
Leaf-02#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 377](#)

Outputs to Verify the Configuration on Border VTEP

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

```

Border# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.3    10b3.d56a.8fc8 50901     UP  A/-/4 17:09:20
nve1     50901    L3CP 172.16.254.4    7c21.0dbd.9548 50901     UP  A/-/4 17:06:19
nve1     50901    L3CP 172.16.254.3    10b3.d56a.8fc8 50901     UP  A/M/6 17:09:20
nve1     50901    L3CP 172.16.254.4    7c21.0dbd.9548 50901     UP  A/M/6 17:06:19
nve1     10101    L2CP 172.16.254.3     6             10101     UP  N/A   17:09:20
nve1     10102    L2CP 172.16.254.4     7             10102     UP  N/A   16:48:24
Border#

```

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

Border VTEP

```

Border# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time

```



```

-----
nve1      10101    172.16.254.3          6          10101    17:09:21
nve1      10102    172.16.254.4          7          10102    16:48:24
Border#

```

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on Border VTEP:

```

Border# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.6, local AS number 65001
BGP table version is 60, main routing table version 60
6 network entries using 1824 bytes of memory
10 path entries using 1360 bytes of memory
4/4 BGP path/bestpath attribute entries using 1248 bytes of memory
4 BGP rinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
19 BGP extended community entries using 2682 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 7298 total bytes of memory
BGP activity 116/24 prefixes, 232/112 paths, scan interval 60 secs
8 networks peaked at 12:14:22 Aug 6 2020 UTC (16:52:46.174 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.255.1  4      65001  1246   1165    60     0   0 17:13:17      4
172.16.255.2  4      65001  1247   1161    60     0   0 17:13:14      4
Border#

```

The following example shows the output for the **show ip pim vrf vrf-name rp mapping** command on Border VTEP:

```

Border# show ip pim vrf green rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 10.2.255.255 (?)
Border#

```

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

```

Border# show ip routing vrf green 10.2.255.255
Routing Table: green
Routing entry for 10.2.255.255/32
  Known via "ospf 2", distance 110, metric 3, type intra area
  Redistributing via bgp 65001
  Advertised by bgp 65001 match internal external 1 & 2
  Last update from 10.2.1.2 on Vlan2001, 17:12:42 ago
  Routing Descriptor Blocks:
  * 10.2.1.2, from 10.2.255.3, 17:12:42 ago, via Vlan2001
    Route metric is 3, traffic share count is 1
Border#

```

The following example shows the output for the **show ip igmp vrf vrf-name groups** command on Border VTEP:

```

Border# show ip igmp vrf green groups
IGMP Connected Group Membership
Group Address      Interface      Uptime      Expires      Last Reporter      Group Accounted

```

```
224.0.1.40      Vlan901          17:14:13  00:02:51  172.16.254.6
Border#
```

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

```
Border# 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,
       e - encap-helper tunnel flag
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.1.1.1), 17:06:19/stopped, RP 10.2.255.255, flags: SJGx
  Incoming interface: Vlan2001, RPF nbr 10.2.1.2
  Outgoing interface list:
    Vlan901, Forward/Sparse, 17:06:19/stopped

(10.2.255.1, 226.1.1.1), 00:33:41/00:01:22, flags: TGqx
  Incoming interface: Vlan2001, RPF nbr 10.2.1.2
  Outgoing interface list:
    Vlan901, Forward/Sparse, 00:33:41/stopped

(10.1.102.12, 226.1.1.1), 00:34:06/00:03:14, flags: Tgx
  Incoming interface: Vlan901, RPF nbr 172.16.254.4
  Outgoing interface list:
    Vlan2001, Forward/Sparse, 00:34:06/00:02:52, A

(*, 224.0.1.40), 17:14:13/00:02:51, RP 10.2.255.255, flags: SJCLGx
  Incoming interface: Vlan2001, RPF nbr 10.2.1.2
  Outgoing interface list:
    Vlan901, Forward/Sparse, 17:14:12/00:02:51
Border#
```

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

```
Border# 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,
            e - Encap helper tunnel flag.
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
Vlan2001 Flags: A NS
Vlan901, VXLAN Decap Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,226.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 6/0/122/0, Other: 0/0/0
Vlan2001 Flags: A NS
Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
  Pkts: 0/0/0   Rate: 0 pps
(10.1.102.12,226.1.1.1) Flags: HW
  SW Forwarding: 4/0/100/0, Other: 1/1/0
  HW Forwarding: 2096/1/126/0, Other: 0/0/0
Vlan901, VXLAN Decap Flags: A
Vlan2001 Flags: F
  Pkts: 0/0/4   Rate: 0 pps
(10.2.255.1,226.1.1.1) Flags: HW
  SW Forwarding: 1/0/100/0, Other: 0/0/0
  HW Forwarding: 2072/1/122/0, Other: 0/0/0
Vlan2001 Flags: A
Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
  Pkts: 0/0/1   Rate: 0 pps
Border#

```

The following example shows the output for the **show bgp ipv4 mvpn all** command on Border VTEP:

```

Border# show bgp ipv4 mvpn all
BGP table version is 60, local router ID is 172.16.255.6
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: 1:1 (default for vrf green)
*>i  [5] [1:1] [10.1.102.12] [226.1.1.1] /18
      172.16.255.4          0      100      0 ?
* i   [5] [1:1] [10.1.102.12] [226.1.1.1] /18
      172.16.255.4          0      100      0 ?
*>   [5] [1:1] [10.2.255.1] [226.1.1.1] /18
      0.0.0.0                32768 ?
* i   [6] [1:1] [65001] [10.2.255.255/32] [224.0.1.40/32] /22
      172.16.255.3          0      100      0 ?
*>i  [6] [1:1] [65001] [10.2.255.255/32] [224.0.1.40/32] /22
      172.16.255.3          0      100      0 ?
* i   [6] [1:1] [65001] [10.2.255.255/32] [226.1.1.1/32] /22
      172.16.255.3          0      100      0 ?
*>i  [6] [1:1] [65001] [10.2.255.255/32] [226.1.1.1/32] /22
      172.16.255.3          0      100      0 ?
*>i  [7] [1:1] [65001] [10.2.255.1/32] [226.1.1.1/32] /22
      172.16.255.3          0      100      0 ?

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

* i          172.16.255.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:102
*>  [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      0.0.0.0                      32768 ?
Border#

```

The following example shows the output for the **show ip mroute** command on Border VTEP:

```

Border# 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,
       e - encap-helper tunnel flag
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), 17:14:04/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 17:14:04/00:01:48

(172.16.254.4, 239.1.1.1), 00:34:05/00:02:44, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:34:05/00:01:54

(172.16.254.6, 239.1.1.1), 00:34:07/00:03:12, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/2, Forward/Sparse, 00:34:07/00:02:52, A

(*, 224.0.1.40), 17:14:13/00:02:47, RP 172.16.255.255, flags: SJCL
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 17:14:12/00:02:47
Border#

```

The following example shows the output for the **show ip mfib** command on Border VTEP:

```

Border# 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,
              e - Encap helper tunnel flag.
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
  TenGigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  8/0/146/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 7/0/125/0, Other: 0/0/0
  HW Forwarding: 8010/0/176/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/7   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  1/0/172/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 5353/0/176/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 11/0/167/0, Other: 7/6/1
  HW Forwarding: 2207/0/151/0, Other: 0/0/0
  Null0 Flags: A
  TenGigabitEthernet1/0/2 Flags: F
  Pkts: 0/0/10  Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  8/0/168/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 4/0/150/0, Other: 0/0/0
  HW Forwarding: 2032/1/168/1, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 2/0/150/0, Other: 4/4/0

```

```

HW Forwarding: 2015/1/156/1, Other: 0/0/0
Null0 Flags: A
TenGigabitEthernet1/0/2 Flags: F
Pkts: 0/0/1 Rate: 0 pps
Border#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 377](#)

Outputs to Verify the Configuration on Spine Switch 1

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on Spine Switch 1:

```

Spine-01# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 169, main routing table version 169
6 network entries using 1824 bytes of memory
16 path entries using 2176 bytes of memory
3/3 BGP path/bestpath attribute entries using 912 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
16 BGP extended community entries using 2332 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 7388 total bytes of memory
BGP activity 250/203 prefixes, 2984/2883 paths, scan interval 60 secs
8 networks peaked at 12:20:11 Aug 6 2020 UTC (16:59:40.011 ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.2	4	65001	1632	1581	169	0	0	20:28:37	6
172.16.255.3	4	65001	1161	1252	169	0	0	17:17:09	4
172.16.255.4	4	65001	1169	1247	169	0	0	17:14:09	4
172.16.255.6	4	65001	1172	1253	169	0	0	17:20:10	2

```

Spine-01#

```

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 (?)
Spine-01#

```

The following example shows the output for the **show bgp ipv4 mvpn all** command on Spine Switch 1:

```

Spine-01# show bgp ipv4 mvpn all
BGP table version is 169, local router ID is 172.16.255.1
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: 1:1
* i [5][1:1][10.1.102.12][226.1.1.1]/18
    172.16.255.4 0 100 0 ?
*>i 172.16.255.4 0 100 0 ?
* i [5][1:1][10.2.255.1][226.1.1.1]/18
    172.16.255.6 0 100 0 ?
*>i 172.16.255.6 0 100 0 ?
* i [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
    172.16.255.4 0 100 0 ?
*>i 172.16.255.3 0 100 0 ?
* i 172.16.255.3 0 100 0 ?
*>i [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
    172.16.255.3 0 100 0 ?
* i 172.16.255.3 0 100 0 ?
* i 172.16.255.4 0 100 0 ?
*>i [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
    172.16.255.3 0 100 0 ?
* i 172.16.255.3 0 100 0 ?
* i 172.16.255.4 0 100 0 ?
Route Distinguisher: 172.16.254.4:102
*>i [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
    172.16.255.3 0 100 0 ?
* i 172.16.255.3 0 100 0 ?
* i 172.16.255.6 0 100 0 ?
Spine-01#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 1:

```

Spine-01# 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), 00:42:45/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(172.16.254.6, 239.1.1.1), 00:42:22/00:02:37, flags: PTA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null

(172.16.254.4, 239.1.1.1), 00:42:45/00:02:28, flags: PTA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null

(*, 224.0.1.40), 1w0d/00:02:18, RP 172.16.255.255, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback2, Forward/Sparse, 1w0d/00:02:18

```

```

(*, 225.0.0.102), 6d19h/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(172.16.254.6, 225.0.0.102), 05:29:52/00:02:22, flags: PA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null

(172.16.254.4, 225.0.0.102), 17:12:35/00:02:03, flags: PA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null
Spine-01#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 1:

```

Spine-01# 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: 82/82/0
  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
  Tunnell Flags: A
  Loopback2 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 1/0/206/0, Other: 279/0/279
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
  GigabitEthernet1/0/2 Flags: NS
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
  GigabitEthernet1/0/4 Flags: NS
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 1/0/1
  HW Forwarding: 0/0/0/0, Other: 0/0/0

```



```

Tunnell Flags: A
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 1224/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A NS
Spine-01#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 377](#)

Outputs to Verify the Configuration on Spine Switch 2

The following example shows the output for the **show bgp ipv4 mvpn all summary** command on Spine Switch 2:

```

Spine-02# show bgp ipv4 mvpn all summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 131, main routing table version 131
6 network entries using 1824 bytes of memory
16 path entries using 2176 bytes of memory
3/3 BGP path/bestpath attribute entries using 912 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
16 BGP extended community entries using 2332 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 7388 total bytes of memory
BGP activity 283/236 prefixes, 3089/2988 paths, scan interval 60 secs
8 networks peaked at 12:20:59 Aug 6 2020 UTC (17:02:43.558 ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.1	4	65001	1584	1635	131	0	0	20:31:41	6
172.16.255.3	4	65001	1160	1252	131	0	0	17:20:09	4
172.16.255.4	4	65001	1173	1249	131	0	0	17:17:14	4
172.16.255.6	4	65001	1172	1258	131	0	0	17:23:12	2

```

Spine-02#

```

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 (?)
Spine-02#

```

The following example shows the output for the **show bgp ipv4 mvpn all** command on Spine Switch 2:

```

Spine-02# show bgp ipv4 mvpn all
BGP table version is 131, local router ID is 172.16.255.2
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,

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

        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: 1:1
* i [5][1:1][10.1.102.12][226.1.1.1]/18
      172.16.255.4          0 100 0 ?
*>i 172.16.255.4          0 100 0 ?
* i [5][1:1][10.2.255.1][226.1.1.1]/18
      172.16.255.6          0 100 0 ?
*>i 172.16.255.6          0 100 0 ?
* i [6][1:1][65001][10.2.255.255/32][224.0.1.40/32]/22
      172.16.255.4          0 100 0 ?
*>i 172.16.255.3          0 100 0 ?
* i 172.16.255.3          0 100 0 ?
*>i [6][1:1][65001][10.2.255.255/32][226.1.1.1/32]/22
      172.16.255.3          0 100 0 ?
* i 172.16.255.3          0 100 0 ?
* i 172.16.255.4          0 100 0 ?
*>i [7][1:1][65001][10.2.255.1/32][226.1.1.1/32]/22
      172.16.255.3          0 100 0 ?
* i 172.16.255.3          0 100 0 ?
* i 172.16.255.4          0 100 0 ?
Route Distinguisher: 172.16.254.4:102
*>i [7][172.16.254.4:102][65001][10.1.102.12/32][226.1.1.1/32]/22
      172.16.255.3          0 100 0 ?
* i 172.16.255.3          0 100 0 ?
* i 172.16.255.6          0 100 0 ?
Spine-02#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 2:

```

Spine-02# 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), 3d00h/00:03:23, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 17:17:14/00:03:23
    GigabitEthernet1/0/1, Forward/Sparse, 17:20:16/00:03:17
    GigabitEthernet1/0/4, Forward/Sparse, 17:23:12/00:02:52

(172.16.254.4, 239.1.1.1), 00:44:04/00:01:34, flags: T
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
  Outgoing interface list:
    GigabitEthernet1/0/4, Forward/Sparse, 00:44:04/00:02:52

```

```

GigabitEthernet1/0/1, Forward/Sparse, 00:44:04/00:03:17
(172.16.254.6, 239.1.1.1), 00:44:04/00:01:32, flags: T
Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
Outgoing interface list:
  GigabitEthernet1/0/1, Forward/Sparse, 00:44:04/00:03:17
  GigabitEthernet1/0/2, Forward/Sparse, 00:44:04/00:03:23
(*, 224.0.1.40), 1w0d/00:03:22, RP 172.16.255.255, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 17:17:14/00:02:46
  GigabitEthernet1/0/1, Forward/Sparse, 17:20:16/00:03:22
  GigabitEthernet1/0/4, Forward/Sparse, 17:23:12/00:03:13
  Loopback2, Forward/Sparse, 1w0d/00:02:33
(*, 225.0.0.102), 1w0d/00:03:29, RP 172.16.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 17:17:14/00:03:04
  GigabitEthernet1/0/1, Forward/Sparse, 17:20:16/00:03:29
  GigabitEthernet1/0/4, Forward/Sparse, 17:23:12/00:02:36
(172.16.254.4, 225.0.0.102), 17:17:08/00:02:44, flags: MT
Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
Outgoing interface list:
  GigabitEthernet1/0/4, Forward/Sparse, 17:17:08/00:02:37
  GigabitEthernet1/0/1, Forward/Sparse, 17:17:08/00:03:29
(172.16.254.6, 225.0.0.102), 17:23:14/00:03:21, flags: MT
Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 17:17:14/00:03:04
  GigabitEthernet1/0/1, Forward/Sparse, 17:20:16/00:03:29
(*, 225.0.0.101), 3d00h/00:03:10, RP 172.16.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 17:17:14/00:03:01
  GigabitEthernet1/0/1, Forward/Sparse, 17:20:16/00:03:10
  GigabitEthernet1/0/4, Forward/Sparse, 17:23:12/00:02:40
(172.16.254.3, 225.0.0.101), 17:19:56/00:02:53, flags: TA
Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
Outgoing interface list:
  GigabitEthernet1/0/2, Forward/Sparse, 17:17:14/00:03:01
  GigabitEthernet1/0/4, Forward/Sparse, 17:19:56/00:03:02
Spine-02#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 2:

```

Spine-02# 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,

```

Example: Configuring TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

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
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
Loopback2 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 9/0/112/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/2   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 4/0/132/0, Other: 0/0/0
  HW Forwarding: 8067/0/176/0, Other: 0/0/0
GigabitEthernet1/0/1 Flags: A
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 27/0/101/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 5404/0/176/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2214/0/163/0, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A NS
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS

```

```

    Pkts: 0/0/0    Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 9/0/150/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/3    Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/3    Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/3    Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2629/1/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0    Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
  Pkts: 0/0/0    Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2607/1/168/1, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A
GigabitEthernet1/0/1 Flags: F NS
  Pkts: 0/0/0    Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/0    Rate: 0 pps
Spine-02#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 377](#)

Outputs to Verify the Configuration on Router 3 (RP Outside the BGP EVPN VXLAN Fabric)

The following example shows the output for the **show ip pim rp mapping** command on Router 3:

```

R3# show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
          RP: 10.2.255.255 (?)
R3#

```

The following example shows the output for the **show ip mroute** command on Router 3:

```

R3# 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

(*, 226.1.1.1), 2d19h/00:03:17, RP 10.2.255.255, flags: S
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  TenGigabitEthernet0/0/0, Forward/Sparse, 17:10:20/00:03:17
  TenGigabitEthernet0/0/1, Forward/Sparse, 2d16h/00:03:11

(10.2.255.1, 226.1.1.1), 00:37:40/00:02:14, flags: PJT
Incoming interface: TenGigabitEthernet0/0/0, RPF nbr 10.2.13.1
Outgoing interface list: Null

(10.1.102.12, 226.1.1.1), 00:38:05/00:02:58, flags: P
Incoming interface: TenGigabitEthernet0/0/0, RPF nbr 10.2.13.1
Outgoing interface list: Null

(*, 224.0.1.40), 2d23h/00:03:27, RP 10.2.255.255, flags: SJCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Loopback0, Forward/Sparse, 2d23h/00:02:11
  TenGigabitEthernet0/0/1, Forward/Sparse, 2d19h/00:03:26
  TenGigabitEthernet0/0/0, Forward/Sparse, 2d19h/00:03:27
R3#

```

Return to [Verifying TRM with PIM-SM for IPv4 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric](#), on page 377

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

This example shows how to configure and verify Layer 3 TRM with PIM-SM for IPv4 and IPv6 multicast traffic when the RP is outside the BGP EVPN VXLAN fabric. The example uses the topology in the [Figure 25: TRM with PIM-SM when the RP is Outside the BGP EVPN VXLAN Fabric](#), on page 371 figure.

The topology shows an EVPN VXLAN network, with two spine switches and three VTEPs, connected to an external network with three routers. Router 3 in the external network acts as the RP in this topology and Border VTEP connects the fabric to the external network through Router 1. The IPv4 multicast group is 226.1.1.1 and the IPv6 multicast group is FF06:1::1 in this topology. The following tables provide sample configurations for the devices in this topology:

Table 40: Configuring VTEP 1, Border VTEP, and VTEP 2 to Configure TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre>Leaf-01# show running-config hostname Leaf-01 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901</pre>	<pre>Border# show running-config hostname Border ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 !</pre>	<pre>Leaf-02# show running-config hostname Leaf-02 ! vrf definition green rd 1:1 ! address-family ipv4 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 mdt auto-discovery vxlan mdt default vxlan 239.1.1.1 mdt overlay use-bgp 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 ip multicast-routing vrf green ! ipv6 unicast-routing ipv6 multicast-routing vrf green ! 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 ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 901 member vni 50901 !</pre>

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> ! interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip pim sparse-mode 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 Loopback901 vrf forwarding green ip address 10.1.255.1 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::1/128 ipv6 enable ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode ipv6 enable no autostate ! </pre>	<pre> vlan 2001 ! 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 Loopback901 vrf forwarding green ip address 10.1.255.4 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::4/128 ipv6 enable ! 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 ! interface TenGigabitEthernet1/0/2 no switchport ip address 172.16.26.6 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface TenGigabitEthernet1/0/5 switchport trunk allowed vlan 2001 switchport mode trunk ! interface TenGigabitEthernet1/0/10 switchport access vlan 102 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! </pre>	<pre> 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 Loopback901 vrf forwarding green ip address 10.1.255.2 255.255.255.255 ip pim sparse-mode ipv6 address FC00:1:255::2/128 ipv6 enable ! 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 GigabitEthernet1/0/2 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 access vlan 101 switchport mode access ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:101::1/64 ipv6 enable ! interface Vlan102 vrf forwarding green ip address 10.1.102.1 255.255.255.0 ip pim sparse-mode ipv6 address FC00:1:102::1/64 ipv6 enable ! interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode ipv6 enable no autostate ! </pre>

VTEP 1	Border VTEP	VTEP 2
<pre> 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 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 ! </pre>	<pre> interface Vlan901 vrf forwarding green ip unnumbered Loopback1 ip pim sparse-mode ipv6 enable no autostate ! interface Vlan2001 vrf forwarding green ip address 10.2.1.1 255.255.255.0 ip mtu 1500 ip pim sparse-mode ip ospf network point-to-point ip ospf 2 area 0 ipv6 address FC00:2::1/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! 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 mcast-group 225.0.0.102 ! router ospfv3 1 ! address-family ipv6 unicast vrf green redistribute bgp 65001 exit-address-family ! router ospf 2 vrf green redistribute bgp 65001 ! router ospf 1 router-id 172.16.255.6 ! 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 </pre>	<pre> 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 mcast-group 225.0.0.102 ! 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 redistribute connected redistribute static exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 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 ! </pre>

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

VTEP 1	Border VTEP	VTEP 2
<pre> 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 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end ! Leaf-01# </pre>	<pre> ! address-family ipv4 exit-address-family ! address-family ipv4 mvpn 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 ipv6 mvpn 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 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 redistribute ospf 2 match internal external 1 external 2 exit-address-family ! address-family ipv6 vrf green redistribute connected redistribute ospf 1 include-connected redistribute static advertise l2vpn evpn exit-address-family ! ip pim rp-address 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ! end ! Border# </pre>	<pre> 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 172.16.255.255 ip pim ssm default ip pim vrf green rp-address 10.2.255.255 ! ipv6 pim vrf green rp-address FC00:2:255::255 ipv6 pim vrf green register-source Loopback901 ! end ! Leaf-02# </pre>

Table 41: Configuring Spine Switch 1 and Spine Switch 2 to Configure TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
----------------	----------------

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 GigabitEthernet1/0/4 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 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 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 ! interface GigabitEthernet1/0/4 no switchport ip address 172.16.26.2 255.255.255.0 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 172.16.255.6 remote-as 65001 neighbor 172.16.255.6 update-source Loopback0 ! address-family ipv4 exit-address-family ! </pre>

Spine Switch 1	Spine Switch 2
<pre> address-family ipv4 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family ipv6 mvpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-01# </pre>	<pre> address-family ipv4 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client exit-address-family ! address-family ipv6 mvpn neighbor 172.16.255.1 activate neighbor 172.16.255.1 send-community both neighbor 172.16.255.1 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 route-reflector-client 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 172.16.255.6 activate neighbor 172.16.255.6 send-community both neighbor 172.16.255.6 route-reflector-client 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 ! end ! Spine-02# </pre>

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric
Table 42: Configuring Router 1, Router 2, and Router 3 to Configure TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

Router 1	Router 2	Router 3
<pre> R1# show running-config hostname R1 ! ip multicast-routing distributed ! ipv6 unicast-routing ipv6 multicast-routing ! interface Loopback0 ip address 10.2.255.1 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ipv6 address FC00:2:255::1/128 ipv6 enable ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:12::1/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.13.1 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:13::1/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface GigabitEthernet0/0/1.2001 encapsulation dot1Q 2001 ip address 10.2.1.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ipv6 address FC00:2:1::2/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 </pre>	<pre> R2# show running-config hostname R2 ! ip multicast-routing distributed ! ipv6 unicast-routing ipv6 multicast-routing ! interface Loopback0 ip address 10.2.255.2 255.255.255.255 ip pim sparse-mode ip igmp join-group 226.1.1.1 ip ospf 1 area 0 ipv6 address FC00:2:255::2/128 ipv6 enable ipv6 mld join-group FF06:1::1 ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.12.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:12::2/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.2 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:23::2/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! router ospfv3 1 ! address-family ipv6 unicast exit-address-family ! router ospf 1 router-id 10.2.255.2 </pre>	<pre> R3# show running-config hostname R3 ! ip multicast-routing distributed ! ipv6 unicast-routing ipv6 multicast-routing ! interface Loopback0 ip address 10.2.255.3 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ipv6 address FC00:2:255::3/128 ipv6 enable ospfv3 1 ipv6 area 0 ! interface Loopback255 ip address 10.2.255.255 255.255.255.255 ip pim sparse-mode ip ospf 1 area 0 ipv6 address FC00:2:255::255/128 ipv6 enable ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/0 ip address 10.2.13.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:13::3/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! interface TenGigabitEthernet0/0/1 ip address 10.2.23.3 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 cdp enable ipv6 address FC00:2:23::3/64 ipv6 enable ospfv3 network point-to-point ospfv3 1 ipv6 area 0 ! router ospfv3 1 ! </pre>

Router 1	Router 2	Router 3
<pre> ! router ospfv3 1 ! address-family ipv6 unicast exit-address-family ! router ospf 1 router-id 10.2.255.1 ! ip pim rp-address 10.2.255.255 ! ipv6 pim rp-address FC00:2:255::255 ! end ! R1# </pre>	<pre> ! ip pim rp-address 10.2.255.255 ! ipv6 pim rp-address FC00:2:255::255 ! end ! R2# </pre>	<pre> address-family ipv6 unicast exit-address-family ! router ospf 1 router-id 10.2.255.3 ! ip pim rp-address 10.2.255.255 ! ipv6 pim rp-address FC00:2:255::255 ! end ! R3# </pre>

Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

The following sections provide sample outputs for **show** commands to verify TRM with PIM-SM on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 411](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 419](#)
- [Outputs to Verify the Configuration on Border VTEP, on page 426](#)
- [Outputs to Verify the Configuration on Spine Switch 1, on page 434](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 437](#)
- [Outputs to Verify the Configuration on Router 3 \(RP Outside the BGP EVPN VXLAN Fabric\), on page 441](#)

Outputs to Verify the Configuration on VTEP 1

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

```

Leaf-01# show nve peers
Interface VNI      Type Peer-IP           RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.6     0c75.bd67.ef48 50901     UP  A/-/4 1d01h
nve1     50901    L3CP 172.16.254.4     7c21.0dbd.9548 50901     UP  A/-/4 1d01h
nve1     50901    L3CP 172.16.254.6     0c75.bd67.ef48 50901     UP  A/M/6 1d01h
nve1     50901    L3CP 172.16.254.4     7c21.0dbd.9548 50901     UP  A/M/6 1d01h
nve1     10102    L2CP 172.16.254.4       7             10102     UP   N/A 1d00h
nve1     10102    L2CP 172.16.254.6       5             10102     UP   N/A 1d01h
Leaf-01#
                    
```

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

```

Leaf-01# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP           Num routes eVNI      UP time
                    
```

```

-----
nve1      10102      172.16.254.4          7          10102      1d00h
nve1      10102      172.16.254.6          5          10102      1d01h
Leaf-01#

```

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on VTEP 1:

```

Leaf-01# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.3, local AS number 65001
BGP table version is 43, main routing table version 43
5 network entries using 1960 bytes of memory
7 path entries using 1120 bytes of memory
3/3 BGP path/bestpath attribute entries using 936 bytes of memory
4 BGP rinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2372 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 6572 total bytes of memory
BGP activity 124/30 prefixes, 253/123 paths, scan interval 60 secs
5 networks peaked at 15:46:09 Aug 6 2020 UTC (21:27:07.275 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   1796   1688    43    0    0  1d01h    2
172.16.255.2  4      65001   1795   1685    43    0    0  1d01h    2
Leaf-01#

```

The following example shows the output for the **show ipv6 pim vrf vrf-name group-map** command on VTEP 1:

```

Leaf-01# show ip pim vrf green group-map ff06:::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: V1901,::FFFF:172.16.254.6
  Info source: Static
  Uptime: 21:43:02, Groups: 1
Leaf-01#

```

The following example shows the output for the **show ipv6 routing vrf** command on VTEP 1:

```

Leaf-01# show ipv6 routing vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "bgp 65001", distance 200, metric 2, type internal
  Route count is 1/1, share count 0
  Routing paths:
    172.16.254.6%default, Vlan901%default
      From AC10:FF01::
      opaque_ptr 0x7FBB863DE268
      Last updated 1d00h ago
Leaf-01#

```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on VTEP 1:


```
Leaf-01# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
      Uptime    Expires
FF06:1::1                                   Vlan101
      21:30:55  00:03:57
Leaf-01#
```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on VTEP 1:

```
Leaf-01# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
      C - Connected, L - Local, I - Received Source Specific Host Report,
      P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
      J - Join SPT, Y - Joined MDT-data group,
      y - Sending to MDT-data group
      g - BGP signal originated, G - BGP Signal received,
      N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
      q - BGP Src-Active originated, Q - BGP Src-Active received
      E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 21:30:55/00:03:08, RP FC00:2:255::255, flags: SCJg
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.6
  Immediate Outgoing interface list:
    Vlan101, Forward, 21:30:55/00:03:08

(FC00:1:102::12, FF06:1::1), 00:01:55/00:01:34, flags: SJTgQ
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.4
  Immediate Outgoing interface list:
    Vlan101, Forward, 00:01:55/00:02:38

(FC00:2:255::1, FF06:1::1), 00:01:14/00:02:15, flags: SJTgQ
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.6
  Immediate Outgoing interface list:
    Vlan101, Forward, 00:01:14/00:03:18
Leaf-01#
```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on VTEP 1:

```
Leaf-01# show ipv6 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,
              e - Encap helper tunnel flag.
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
```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

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
(*,FF00::/8) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 412/412/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF00::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF02::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 4/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A NS
  Vlan101 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(FC00:1:102::12,FF06:1::1) Flags: HW
  SW Forwarding: 3/0/100/0, Other: 0/0/0
  HW Forwarding: 58/0/125/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan101 Flags: F NS
  Pkts: 0/0/3   Rate: 0 pps
(FC00:2:255::1,FF06:1::1) Flags: HW
  SW Forwarding: 1/0/100/0, Other: 0/0/0
  HW Forwarding: 36/0/126/0, Other: 0/0/0
  Vlan901, VXLAN Decap Flags: A
  Vlan101 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,FF10::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF12::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF20::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF22::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF30::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF32::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF33::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF34::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF35::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF36::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF37::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF38::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF39::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3A::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3B::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3C::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3D::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3E::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3F::/32) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF40::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF42::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF50::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF52::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF60::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF62::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF70::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF72::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF80::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF82::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF90::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF92::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC0::/15) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC2::/16) Flags:
SW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

(*,FFD0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
Leaf-01#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on VTEP 1:

```

Leaf-01# show bgp ipv6 mvpn all
BGP table version is 43, local router ID is 172.16.255.3
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: 1:1 (default for vrf green)
*>i  [5][1:1][FC00:1:102::12][FF06:1::1]/42
      172.16.255.4          0      100      0 ?
* i      172.16.255.4          0      100      0 ?
*>i  [5][1:1][FC00:2:255::1][FF06:1::1]/42
      172.16.255.6          0      100      0 ?
* i      172.16.255.6          0      100      0 ?
*>  [6][1:1][65001][FC00:2:255::255][FF06:1::1]/46
      ::                      32768 ?
*>  [7][1:1][65001][FC00:2:255::1][FF06:1::1]/46
      ::                      32768 ?
Route Distinguisher: 172.16.254.4:102
*>  [7][172.16.254.4:102][65001][FC00:1:102::12][FF06:1::1]/46
      ::                      32768 ?
Leaf-01#

```

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

```

Leaf-01# 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-stargy configured on rpf intf,

```

```

    e - encap-helper tunnel flag
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), 1d01h/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:49

(172.16.254.4, 239.1.1.1), 00:01:54/00:01:05, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:01:54/00:01:05

(172.16.254.6, 239.1.1.1), 00:01:56/00:01:03, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:01:56/00:01:03

(*, 224.0.1.40), 1d01h/00:02:53, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 1d01h/00:02:53

(*, 225.0.0.102), 1d01h/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:49

(172.16.254.4, 225.0.0.102), 1d01h/00:02:01, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:10

(172.16.254.6, 225.0.0.102), 1d01h/00:02:20, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.23.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:49

(*, 225.0.0.101), 1d01h/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, 1d01h/00:02:49

(172.16.254.3, 225.0.0.101), 1d01h/00:01:58, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:03:08
Leaf-01#

```

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

```

Leaf-01# 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,
              e - Encap helper tunnel flag.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

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
  GigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  1/0/114/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 13/0/127/0, Other: 2/2/0
  HW Forwarding:  7870/0/164/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  2/0/172/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding:  5222/0/176/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding:  2137/0/163/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding:  11/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 4/0/150/0, Other: 0/0/0
  HW Forwarding:  518/0/168/1, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0

```

```

HW Forwarding: 498/1/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0/1 Rate: 0 pps
Leaf-01#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 411](#)

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peers
Interface VNI Type Peer-IP RMAC/Num_RTs eVNI state flags UP time
nve1 50901 L3CP 172.16.254.6 0c75.bd67.ef48 50901 UP A/-/4 1d01h
nve1 50901 L3CP 172.16.254.3 10b3.d56a.8fc8 50901 UP A/-/4 1d01h
nve1 50901 L3CP 172.16.254.6 0c75.bd67.ef48 50901 UP A/M/6 1d01h
nve1 50901 L3CP 172.16.254.3 10b3.d56a.8fc8 50901 UP A/M/6 1d01h
nve1 10101 L2CP 172.16.254.3 6 10101 UP N/A 1d01h
nve1 10102 L2CP 172.16.254.6 5 10102 UP N/A 1d01h
Leaf-02#

```

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

```

Leaf-02# show l2vpn evpn peers vxlan
Interface VNI Peer-IP Num routes eVNI UP time
-----
nve1 10101 172.16.254.3 6 10101 1d01h
nve1 10102 172.16.254.6 5 10102 1d01h
Leaf-02#

```

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on VTEP 2:

```

Leaf-02# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.4, local AS number 65001
BGP table version is 63, main routing table version 63
6 network entries using 2352 bytes of memory
8 path entries using 1280 bytes of memory
4/4 BGP path/bestpath attribute entries using 1248 bytes of memory
4 BGP rrinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
17 BGP extended community entries using 2372 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 7436 total bytes of memory
BGP activity 128/33 prefixes, 221/93 paths, scan interval 60 secs
6 networks peaked at 15:50:41 Aug 6 2020 UTC (21:30:56.871 ago)

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
172.16.255.1 4 65001 1797 1698 63 0 0 1d01h 2
172.16.255.2 4 65001 1792 1701 63 0 0 1d01h 2
Leaf-02#

```

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

```
Leaf-02# show ip pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF06:1::/8*
  SM, RP: FC00:2:255::255
  RPF: V1901,::FFFF:172.16.254.6
  Info source: Static
  Uptime: 1d01h, Groups: 1
Leaf-02#
```

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

```
Leaf-02# show ip routing vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "bgp 65001", distance 200, metric 2, type internal
  Route count is 1/1, share count 0
  Routing paths:
    172.16.254.6%default, Vlan901%default
      From AC10:FF01::
        opaque_ptr 0x7F65BA333AD0
        Last updated 1d01h ago
Leaf-02#
```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on VTEP 2:

```
Leaf-02# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
  Uptime    Expires
FF06:1::1                                     Vlan102
  1d00h     00:02:25
Leaf-02#
```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on VTEP 2:

```
Leaf-02# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 1d00h/never, RP FC00:2:255::255, flags: SCJg
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.6
  Immediate Outgoing interface list:
    Vlan102, Forward, 1d00h/never

(FC00:1:102::12, FF06:1::1), 00:05:45/00:01:27, flags: SFJTGq
  Incoming interface: Vlan102
  RPF nbr: FE80::46D3:CAFF:FE28:6CC5
```



```

Immediate Outgoing interface list:
  Vlan901, Forward, 00:05:45/never

(FC00:2:255::1, FF06:1::1), 00:05:04/00:02:07, flags: SJTgQ
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.6
  Inherited Outgoing interface list:
    Vlan102, Forward, 1d00h/never
Leaf-02#

```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on VTEP 2:

```

Leaf-02# show ipv6 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,
             e - Encap helper tunnel flag.
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
(*,FF00::/8) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF00::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF02::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 9/9/0
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 3/0/126/0, Other: 0/0/0
Vlan901, VXLAN Decap Flags: A NS
Vlan102 Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(FC00:1:102::12,FF06:1::1) Flags: HW
  SW Forwarding: 3/0/100/0, Other: 2/2/0
  HW Forwarding: 168/0/118/0, Other: 0/0/0
Vlan102 Flags: A F
  Pkts: 0/0/0   Rate: 0 pps
Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
  Pkts: 0/0/2   Rate: 0 pps
(FC00:2:255::1,FF06:1::1) Flags: HW
  SW Forwarding: 1/0/100/0, Other: 0/0/0
  HW Forwarding: 148/0/126/0, Other: 0/0/0
Vlan901, VXLAN Decap Flags: A
Vlan102 Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,FF10::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF12::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

(*,FF20::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF22::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF30::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF32::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF33::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF34::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF35::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF36::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF37::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF38::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF39::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3A::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3B::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3C::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3D::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3E::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3F::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF40::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF42::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF50::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF52::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF60::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF62::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

(*,FF70::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF72::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF80::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF82::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF90::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF92::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
Leaf-02#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on VTEP 2:

```

Leaf-02# show bgp ipv6 mvpn all
BGP table version is 63, local router ID is 172.16.255.4
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: 1:1 (default for vrf green)
*>   [5][1:1][FC00:1:102::12][FF06:1::1]/42
      ::
      32768 ?
* i   [5][1:1][FC00:2:255::1][FF06:1::1]/42

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

          172.16.255.6          0    100    0 ?
*>i          172.16.255.6          0    100    0 ?
*>  [6][1:1][65001][FC00:2:255::255][FF06:1::1]/46
      ::                          32768 ?
*>i  [7][1:1][65001][FC00:1:102::12][FF06:1::1]/46
          172.16.255.3          0    100    0 ?
*>  [7][1:1][65001][FC00:2:255::1][FF06:1::1]/46
      ::                          32768 ?
Route Distinguisher: 172.16.254.4:102
* i  [7][172.16.254.4:102][65001][FC00:1:102::12][FF06:1::1]/46
          172.16.255.3          0    100    0 ?
*>i          172.16.255.3          0    100    0 ?
Leaf-02#

```

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

```

Leaf-02# 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,
       e - encap-helper tunnel flag
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), 1d01h/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, 1d01h/00:01:32

(172.16.254.4, 239.1.1.1), 00:05:43/00:01:46, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 00:05:43/00:02:43

(172.16.254.6, 239.1.1.1), 00:05:45/00:01:06, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:05:45/00:00:14

(*, 224.0.1.40), 1d01h/00:02:31, RP 172.16.255.255, flags: SJCL
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 1d01h/00:02:31

(*, 225.0.0.102), 1d01h/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, 1d01h/00:01:32

(172.16.254.6, 225.0.0.102), 1d01h/00:00:55, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2

```

```

Outgoing interface list:
  Tunnel0, Forward/Sparse, 1d01h/00:01:32

(172.16.254.4, 225.0.0.102), 1d01h/00:01:49, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:03:26

(*, 225.0.0.101), 1d01h/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:01:32

(172.16.254.3, 225.0.0.101), 1d01h/00:01:46, flags: JTx
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:01:32
Leaf-02#

```

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

```

Leaf-02# 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,
               e - Encap helper tunnel flag.
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
GigabitEthernet1/0/2 Flags: A NS
Loopback0 Flags: F IC NS
  Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2/0/170/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding: 7870/0/176/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/1   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

HW Forwarding: 1/0/224/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 2/0/163/0, Other: 3/1/2
  HW Forwarding: 5353/0/164/0, Other: 0/0/0
  Null0 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/206/0, Other: 0/0/0
  HW Forwarding: 2165/0/163/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 5/0/168/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0   Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 1495/1491/4
  HW Forwarding: 742/0/156/0, Other: 0/0/0
  Null0 Flags: A NS
  GigabitEthernet1/0/2 Flags: F
    Pkts: 0/0/1   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 1/0/150/0, Other: 0/0/0
  HW Forwarding: 1460/1/168/1, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1   Rate: 0 pps
Leaf-02#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric](#), on page 411

Outputs to Verify the Configuration on Border VTEP

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

```

Border# show nve peers
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.3    10b3.d56a.8fc8 50901      UP  A/-/4 1d01h
nve1     50901    L3CP 172.16.254.4    7c21.0dbd.9548 50901      UP  A/-/4 1d01h
nve1     50901    L3CP 172.16.254.3    10b3.d56a.8fc8 50901      UP  A/M/6 1d01h
nve1     50901    L3CP 172.16.254.4    7c21.0dbd.9548 50901      UP  A/M/6 1d01h
nve1     10101    L2CP 172.16.254.3      6              10101      UP  N/A   1d01h
nve1     10102    L2CP 172.16.254.4      7              10102      UP  N/A   1d00h
Border#

```

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

```

Border# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes eVNI      UP time

```

```

-----
nve1      10101    172.16.254.3          6           10101    1d01h
nve1      10102    172.16.254.4          7           10102    1d00h
Border#

```

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Border VTEP:

```

Border# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.6, local AS number 65001
BGP table version is 62, main routing table version 62
5 network entries using 1960 bytes of memory
8 path entries using 1280 bytes of memory
4/4 BGP path/bestpath attribute entries using 1248 bytes of memory
4 BGP rinfo entries using 160 bytes of memory
1 BGP community entries using 24 bytes of memory
19 BGP extended community entries using 2682 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 7354 total bytes of memory
BGP activity 122/28 prefixes, 244/122 paths, scan interval 60 secs
5 networks peaked at 15:42:39 Aug 6 2020 UTC (21:35:36.535 ago)

Neighbor      V          AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.255.1  4          65001   1810   1710     62     0   0 1d01h    3
172.16.255.2  4          65001   1810   1704     62     0   0 1d01h    3
Border#

```

The following example shows the output for the **show ip pim vrf vrf-name group-map** command on Border VTEP:

```

Border# show ip pim vrf green group-map ff06:1::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: V12001,FE80::A2B4:39FF:FE21:9183
  Info source: Static
  Uptime: 1d01h, Groups: 1
Border#

```

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

```

Border# show ip routing vrf green FC00:2:255::255
Routing entry for FC00:2:255::255/128
  Known via "ospf 1", distance 110, metric 2, type intra area
  Redistributing via bgp 65001
  Route count is 1/1, share count 0
  Routing paths:
    FE80::A2B4:39FF:FE21:9183, Vlan2001
      From FE80::A2B4:39FF:FE21:9183
      Last updated 1d01h ago
Border#

```

The following example shows the output for the **show ipv6 mld vrf vrf-name groups** command on Border VTEP:

```

Border# show ipv6 mld vrf green groups
MLD Connected Group Membership
Group Address                               Interface
  Uptime    Expires
FF06:1::1                                     Vlan102
  1d00h    00:04:02
Border#

```

The following example shows the output for the **show ipv6 mroute vrf vrf-name** command on Border VTEP:

```

Border# show ipv6 mroute vrf green
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 1d00h/never, RP FC00:2:255::255, flags: SCG
  Incoming interface: Vlan2001
  RPF nbr: FE80::A2B4:39FF:FE21:9183
  Immediate Outgoing interface list:
    Vlan102, Null, 1d00h/never
    Vlan901, Forward, 1d00h/never

(FC00:1:102::12, FF06:1::1), 00:10:24/now, flags: STg
  Incoming interface: Vlan901
  RPF nbr: ::FFFF:172.16.254.4
  Immediate Outgoing interface list:
    Vlan2001, Forward, 00:10:24/00:03:05
  Inherited Outgoing interface list:
    Vlan102, Null, 1d00h/never

(FC00:2:255::1, FF06:1::1), 00:09:43/never, flags: STGq
  Incoming interface: Vlan2001
  RPF nbr: FE80::A2B4:39FF:FE21:9183
  Immediate Outgoing interface list:
    Vlan901, Forward, 00:09:43/never
  Inherited Outgoing interface list:
    Vlan102, Null, 1d00h/never
Border#

```

The following example shows the output for the **show ipv6 mfib vrf vrf-name** command on Border VTEP:

```

Border# show ipv6 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,
              e - Encap helper tunnel flag.
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

```
(*,FF00::/8) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF00::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF02::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 9/9/0
(*,FF06:1::1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 3/3/0
  HW Forwarding: 7/0/122/0, Other: 0/0/0
Vlan2001 Flags: A
Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
  Pkts: 0/0/0 Rate: 0 pps
(FC00:1:102::12,FF06:1::1) Flags: HW
  SW Forwarding: 2/0/100/0, Other: 1/0/1
  HW Forwarding: 311/0/125/0, Other: 0/0/0
Vlan901, VXLAN Decap Flags: A
Vlan2001 Flags: F NS
  Pkts: 0/0/2 Rate: 0 pps
(FC00:2:255::1,FF06:1::1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 291/0/122/0, Other: 0/0/0
Vlan2001 Flags: A
Vlan901, VXLAN v4 Encap (50901, 239.1.1.1) Flags: F
  Pkts: 0/0/0 Rate: 0 pps
(*,FF10::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF12::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF20::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF22::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF30::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF32::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF33::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF34::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF35::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF36::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF37::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
```

```

(*,FF38::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF39::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3A::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3B::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3C::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3D::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3E::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF3F::/32) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF40::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF42::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF50::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF52::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF60::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF62::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF70::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF72::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF80::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF82::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF90::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF92::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFA2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFB2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0

```

```

(*,FFC0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFC2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFD2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFE2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF0::/15) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,FFF2::/16) Flags:
  SW Forwarding: 0/0/0/0, Other: 0/0/0
Border#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Border VTEP:

```

Border# show bgp ipv4 mvpn all
BGP table version is 62, local router ID is 172.16.255.6
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: 1:1 (default for vrf green)
*>i  [5][1:1][FC00:1:102::12][FF06:1::1]/42
      172.16.255.4          0    100    0 ?
* i   [5][1:1][FC00:2:255::1][FF06:1::1]/42
      172.16.255.4          0    100    0 ?
*>   [5][1:1][FC00:2:255::1][FF06:1::1]/42
      ::                      32768 ?
* i   [6][1:1][65001][FC00:2:255::255][FF06:1::1]/46
      172.16.255.3          0    100    0 ?
*>i   [6][1:1][65001][FC00:2:255::1][FF06:1::1]/46
      172.16.255.3          0    100    0 ?
* i   [7][1:1][65001][FC00:2:255::1][FF06:1::1]/46
      172.16.255.3          0    100    0 ?
*>i   [7][1:1][65001][FC00:2:255::1][FF06:1::1]/46
      172.16.255.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:102
*>   [7][172.16.254.4:102][65001][FC00:1:102::12][FF06:1::1]/46
      ::                      32768 ?
Border#

```

The following example shows the output for the **show ip mroute** command on Border VTEP:

```

Border# 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,

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

    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,
    e - encap-helper tunnel flag
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), 1d01h/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:41

(172.16.254.4, 239.1.1.1), 00:10:23/00:02:45, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:10:23/00:01:36

(172.16.254.6, 239.1.1.1), 00:10:25/00:03:25, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/2, Forward/Sparse, 00:10:25/00:02:56

(*, 224.0.1.40), 1d01h/00:02:45, RP 172.16.255.255, flags: SJCL
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Loopback0, Forward/Sparse, 1d01h/00:02:45

(*, 225.0.0.102), 1d01h/stopped, RP 172.16.255.255, flags: SJCFx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:41

(172.16.254.4, 225.0.0.102), 1d01h/00:02:35, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:41

(172.16.254.6, 225.0.0.102), 1d01h/00:03:27, flags: FTx
  Incoming interface: Loopback1, RPF nbr 0.0.0.0
  Outgoing interface list:
    TenGigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:02:49, A

(*, 225.0.0.101), 1d01h/stopped, RP 172.16.255.255, flags: SJCx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:41

(172.16.254.3, 225.0.0.101), 1d01h/00:01:12, flags: JTx
  Incoming interface: TenGigabitEthernet1/0/2, RPF nbr 172.16.26.2
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 1d01h/00:02:53
Border#

```

The following example shows the output for the **show ip mfib** command on Border VTEP:

```

Border# 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,
 e - Encap helper tunnel flag.
 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
  TenGigabitEthernet1/0/2 Flags: A NS
  Loopback0 Flags: F IC NS
    Pkts: 0/0/0 Rate: 0 pps
(* ,225.0.0.101) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 8/0/146/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 7/0/125/0, Other: 0/0/0
  HW Forwarding: 8010/0/176/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/7 Rate: 0 pps
(* ,225.0.0.102) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1/0/172/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 1/0/154/0, Other: 0/0/0
  HW Forwarding: 5353/0/176/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/1 Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 11/0/167/0, Other: 7/6/1
  HW Forwarding: 2207/0/151/0, Other: 0/0/0
  Null0 Flags: A
  TenGigabitEthernet1/0/2 Flags: F
    Pkts: 0/0/10 Rate: 0 pps
(* ,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
(* ,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 8/0/168/0, Other: 0/0/0
  TenGigabitEthernet1/0/2 Flags: A NS
  Tunnel0, VXLAN Decap Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
```

```

SW Forwarding: 4/0/150/0, Other: 0/0/0
HW Forwarding: 2032/1/168/1, Other: 0/0/0
TenGigabitEthernet1/0/2 Flags: A
Tunnel0, VXLAN Decap Flags: F NS
  Pkts: 0/0/4   Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 2/0/150/0, Other: 4/4/0
HW Forwarding: 2015/1/156/1, Other: 0/0/0
Null0 Flags: A
TenGigabitEthernet1/0/2 Flags: F
  Pkts: 0/0/1   Rate: 0 pps
Border#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric](#), on page 411

Outputs to Verify the Configuration on Spine Switch 1

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Spine Switch 1:

```

Spine-01# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.1, local AS number 65001
BGP table version is 61, main routing table version 61
5 network entries using 1960 bytes of memory
13 path entries using 2080 bytes of memory
3/3 BGP path/bestpath attribute entries using 912 bytes of memory
3 BGP rinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
16 BGP extended community entries using 2332 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 7428 total bytes of memory
BGP activity 257/209 prefixes, 3003/2900 paths, scan interval 60 secs
5 networks peaked at 15:48:28 Aug 6 2020 UTC (21:38:24.468 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2  4      65001  2190   2137    61    0    0 1d04h    5
172.16.255.3  4      65001  1700   1808    61    0    0 1d01h    3
172.16.255.4  4      65001  1706   1805    61    0    0 1d01h    3
172.16.255.6  4      65001  1713   1813    61    0    0 1d01h    2
Spine-01#

```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Spine Switch 1:

```

Spine-01# show bgp ipv6 mvpn all
BGP table version is 61, local router ID is 172.16.255.1
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: 1:1
* i   [5][1:1][FC00:1:102::12][FF06:1::1]/42
      172.16.255.4           0      100      0 ?
*>i   172.16.255.4           0      100      0 ?

```

```

* i [5] [1:1] [FC00:2:255::1] [FF06:1::1]/42
    172.16.255.6          0    100    0 ?
*>i          172.16.255.6          0    100    0 ?
*>i [6] [1:1] [65001] [FC00:2:255::255] [FF06:1::1]/46
    172.16.255.3          0    100    0 ?
* i          172.16.255.3          0    100    0 ?
* i          172.16.255.4          0    100    0 ?
* i [7] [1:1] [65001] [FC00:2:255::1] [FF06:1::1]/46
    172.16.255.4          0    100    0 ?
* i          172.16.255.3          0    100    0 ?
*>i          172.16.255.3          0    100    0 ?
Route Distinguisher: 172.16.254.4:102
*>i [7] [172.16.254.4:102] [65001] [FC00:1:102::12] [FF06:1::1]/46
    172.16.255.3          0    100    0 ?
* i          172.16.255.3          0    100    0 ?
* i          172.16.255.6          0    100    0 ?
Spine-01#

```

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 (?)
Spine-01#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 1:

```

Spine-01# 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), 00:13:12/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, 239.1.1.1), 00:11:10/00:01:49, flags: PA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null

(172.16.254.6, 239.1.1.1), 00:13:12/00:02:08, flags: PA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null

(*, 224.0.1.40), 1w0d/00:02:04, RP 172.16.255.255, flags: SJCL

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
  Loopback2, Forward/Sparse, 1w0d/00:02:04

(*, 225.0.0.102), 1w0d/stopped, RP 172.16.255.255, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null

(172.16.254.6, 225.0.0.102), 00:19:31/00:02:22, flags: PA
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.16.6
  Outgoing interface list: Null

(172.16.254.4, 225.0.0.102), 1d01h/00:01:52, flags: PA
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.14.4
  Outgoing interface list: Null
Spine-01#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 1:

```

Spine-01# 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: 82/82/0
  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
  Tunnel1 Flags: A
  Loopback2 Flags: F IC NS
    Pkts: 0/0/0   Rate: 0 pps
(*,225.0.0.102) Flags: C HW
  SW Forwarding: 1/0/206/0, Other: 279/0/279
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel1 Flags: A
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel1 Flags: A
  GigabitEthernet1/0/2 Flags: NS
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel1 Flags: A
  GigabitEthernet1/0/4 Flags: NS
(*,232.0.0.0/8) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0

```



```
(*,239.1.1.1) Flags: C HW
  SW Forwarding: 0/0/0/0, Other: 1/0/1
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
(172.16.254.4,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 1224/0/168/0, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
(172.16.254.6,239.1.1.1) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A NS
Spine-01#
```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 411](#)

Outputs to Verify the Configuration on Spine Switch 2

The following example shows the output for the **show bgp ipv6 mvpn all summary** command on Spine Switch 2:

```
Spine-02# show bgp ipv6 mvpn all summary
BGP router identifier 172.16.255.2, local AS number 65001
BGP table version is 61, main routing table version 61
5 network entries using 1960 bytes of memory
13 path entries using 2080 bytes of memory
3/3 BGP path/bestpath attribute entries using 912 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
1 BGP community entries using 24 bytes of memory
16 BGP extended community entries using 2332 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 7428 total bytes of memory
BGP activity 288/240 prefixes, 3108/3005 paths, scan interval 60 secs
5 networks peaked at 15:49:16 Aug 6 2020 UTC (21:40:40.843 ago)

Neighbor          V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1     4      65001  2139   2193     61    0    0 1d04h      5
172.16.255.3     4      65001  1700   1810     61    0    0 1d01h      3
172.16.255.4     4      65001  1711   1803     61    0    0 1d01h      3
172.16.255.5     4      65001    0      0        1    0    0 08:41:01 Idle
172.16.255.6     4      65001  1710   1815     61    0    0 1d01h      2
172.16.255.7     4      65001    0      0        1    0    0 08:40:29 Idle
Spine-02#
```

The following example shows the output for the **show bgp ipv6 mvpn all** command on Spine Switch 2:

```
Spine-02# show bgp ipv6 mvpn all
BGP table version is 61, local router ID is 172.16.255.2
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: 1:1
```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

* i [5][1:1][FC00:1:102::12][FF06:1::1]/42
    172.16.255.4      0 100 0 ?
*>i 172.16.255.4      0 100 0 ?
* i [5][1:1][FC00:2:255::1][FF06:1::1]/42
    172.16.255.6      0 100 0 ?
*>i 172.16.255.6      0 100 0 ?
*>i [6][1:1][65001][FC00:2:255::255][FF06:1::1]/46
    172.16.255.3      0 100 0 ?
* i 172.16.255.3      0 100 0 ?
* i 172.16.255.4      0 100 0 ?
* i [7][1:1][65001][FC00:2:255::1][FF06:1::1]/46
    172.16.255.4      0 100 0 ?
* i 172.16.255.3      0 100 0 ?
*>i 172.16.255.3      0 100 0 ?
Route Distinguisher: 172.16.254.4:102
*>i [7][172.16.254.4:102][65001][FC00:1:102::12][FF06:1::1]/46
    172.16.255.3      0 100 0 ?
* i 172.16.255.3      0 100 0 ?
* i 172.16.255.6      0 100 0 ?
Spine-02#

```

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 (?)
Spine-02#

```

The following example shows the output for the **show ip mroute** command on Spine Switch 2:

```

Spine-02# 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), 3d08h/00:03:24, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:03:24
    GigabitEthernet1/0/1, Forward/Sparse, 1d01h/00:03:06
    GigabitEthernet1/0/4, Forward/Sparse, 1d01h/00:03:02

(172.16.254.4, 239.1.1.1), 00:15:27/00:02:45, flags: T
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4

```

```

Outgoing interface list:
  GigabitEthernet1/0/4, Forward/Sparse, 00:15:27/00:03:02
  GigabitEthernet1/0/1, Forward/Sparse, 00:15:27/00:03:06

(172.16.254.6, 239.1.1.1), 00:15:29/00:02:38, flags: MT
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
  Outgoing interface list:
    GigabitEthernet1/0/1, Forward/Sparse, 00:15:29/00:03:06
    GigabitEthernet1/0/2, Forward/Sparse, 00:15:29/00:03:24

(*, 224.0.1.40), 1w0d/00:03:27, RP 172.16.255.255, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:02:31
    GigabitEthernet1/0/1, Forward/Sparse, 1d01h/00:03:27
    GigabitEthernet1/0/4, Forward/Sparse, 1d01h/00:03:08
    Loopback2, Forward/Sparse, 1w0d/00:02:17

(*, 225.0.0.102), 1w0d/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, 1d01h/00:03:02
    GigabitEthernet1/0/1, Forward/Sparse, 1d01h/00:02:50
    GigabitEthernet1/0/4, Forward/Sparse, 1d01h/00:03:21

(172.16.254.4, 225.0.0.102), 1d01h/00:01:55, flags: MT
  Incoming interface: GigabitEthernet1/0/2, RPF nbr 172.16.24.4
  Outgoing interface list:
    GigabitEthernet1/0/4, Forward/Sparse, 1d01h/00:03:21
    GigabitEthernet1/0/1, Forward/Sparse, 1d01h/00:03:02

(172.16.254.6, 225.0.0.102), 1d01h/00:02:03, flags: MT
  Incoming interface: GigabitEthernet1/0/4, RPF nbr 172.16.26.6
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:03:02
    GigabitEthernet1/0/1, Forward/Sparse, 1d01h/00:03:13

(*, 225.0.0.101), 3d08h/00:03:29, RP 172.16.255.255, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:03:05
    GigabitEthernet1/0/1, Forward/Sparse, 1d01h/00:03:07
    GigabitEthernet1/0/4, Forward/Sparse, 1d01h/00:03:29

(172.16.254.3, 225.0.0.101), 1d01h/00:02:39, flags: TA
  Incoming interface: GigabitEthernet1/0/1, RPF nbr 172.16.23.3
  Outgoing interface list:
    GigabitEthernet1/0/2, Forward/Sparse, 1d01h/00:03:05
    GigabitEthernet1/0/4, Forward/Sparse, 1d01h/00:03:29
Spine-02#

```

The following example shows the output for the **show ip mfib** command on Spine Switch 2:

```

Spine-02# 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,

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

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
  Tunnell Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  Loopback2 Flags: F IC NS
    Pkts: 0/0/0 Rate: 0 pps
(* ,225.0.0.101) Flags: C HW
  SW Forwarding: 9/0/112/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/2 Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/2 Rate: 0 pps
(172.16.254.3,225.0.0.101) Flags: HW
  SW Forwarding: 4/0/132/0, Other: 0/0/0
  HW Forwarding: 8067/0/176/0, Other: 0/0/0
  GigabitEthernet1/0/1 Flags: A
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/4 Rate: 0 pps
(* ,225.0.0.102) Flags: C HW
  SW Forwarding: 27/0/101/0, Other: 0/0/0
  HW Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnell Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/2 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.4,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 5404/0/176/0, Other: 0/0/0
  GigabitEthernet1/0/2 Flags: A
  GigabitEthernet1/0/1 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
  GigabitEthernet1/0/4 Flags: F NS
    Pkts: 0/0/0 Rate: 0 pps
(172.16.254.6,225.0.0.102) Flags: HW
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  HW Forwarding: 2214/0/163/0, Other: 0/0/0
  GigabitEthernet1/0/4 Flags: A NS
  GigabitEthernet1/0/1 Flags: F NS
```

```

      Pkts: 0/0/0      Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
      Pkts: 0/0/0      Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,239.1.1.1) Flags: C HW
SW Forwarding: 9/0/150/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Tunnell Flags: A
GigabitEthernet1/0/1 Flags: F NS
      Pkts: 0/0/3      Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
      Pkts: 0/0/3      Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
      Pkts: 0/0/3      Rate: 0 pps
(172.16.254.4,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 2629/1/168/1, Other: 0/0/0
GigabitEthernet1/0/2 Flags: A
GigabitEthernet1/0/1 Flags: F NS
      Pkts: 0/0/0      Rate: 0 pps
GigabitEthernet1/0/4 Flags: F NS
      Pkts: 0/0/0      Rate: 0 pps
(172.16.254.6,239.1.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 2607/1/168/1, Other: 0/0/0
GigabitEthernet1/0/4 Flags: A
GigabitEthernet1/0/1 Flags: F NS
      Pkts: 0/0/0      Rate: 0 pps
GigabitEthernet1/0/2 Flags: F NS
      Pkts: 0/0/0      Rate: 0 pps
Spine-02#

```

Return to [Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 411](#)

Outputs to Verify the Configuration on Router 3 (RP Outside the BGP EVPN VXLAN Fabric)

The following example shows the output for the **show ip pim group-map** command on Router 3:

```

R3# show ipv6 pim group-map ff06:::1
IP PIM Group Mapping Table
(* indicates group mappings being used)

FF00::/8*
  SM, RP: FC00:2:255::255
  RPF: Tu4,FC00:2:255::255 (us)
  Info source: Static
  Uptime: 1d04h, Groups: 1
R3#

```

The following example shows the output for the **show ipv6 mroute** command on Router 3:

```

R3# show ipv6 mroute
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group

```

Example: Configuring TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric

```

g - BGP signal originated, G - BGP Signal received,
N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
q - BGP Src-Active originated, Q - BGP Src-Active received
E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:1::1), 1d04h/00:03:12, RP FC00:2:255::255, flags: S
  Incoming interface: Tunnel4
  RPF nbr: FC00:2:255::255
  Immediate Outgoing interface list:
    TenGigabitEthernet0/0/1, Forward, 1d04h/00:03:12
    TenGigabitEthernet0/0/0, Forward, 1d01h/00:02:45

(FC00:1:102::12, FF06:1::1), 00:18:43/00:03:12, RP FC00:2:255::255, flags: SPR
  Incoming interface: Tunnel4
  RPF nbr: FC00:2:255::255
  Immediate Outgoing interface list:
    TenGigabitEthernet0/0/0, Null, 00:18:43/00:02:45
    TenGigabitEthernet0/0/1, Null, 00:18:43/00:03:12

(FC00:1:102::12, FF06:1::1), 00:18:45/00:03:12, flags: S
  Incoming interface: TenGigabitEthernet0/0/0
  RPF nbr: FE80::A2B4:39FF:FE21:9181
  Inherited Outgoing interface list:
    TenGigabitEthernet0/0/1, Forward, 1d04h/00:03:12

(FC00:2:255::1, FF06:1::1), 00:18:08/00:02:55, RP FC00:2:255::255, flags: SPR
  Incoming interface: Tunnel4
  RPF nbr: FC00:2:255::255
  Immediate Outgoing interface list:
    TenGigabitEthernet0/0/0, Null, 00:18:08/00:02:45
    TenGigabitEthernet0/0/1, Null, 00:18:04/00:03:12

(FC00:2:255::1, FF06:1::1), 00:18:06/00:02:55, flags: S
  Incoming interface: TenGigabitEthernet0/0/0
  RPF nbr: FE80::A2B4:39FF:FE21:9181
  Inherited Outgoing interface list:
    TenGigabitEthernet0/0/1, Forward, 1d04h/00:03:12
R3#

```

[Return to Verifying TRM with PIM-SM for IPv4 and IPv6 Multicast Traffic when the RP is Outside the BGP EVPN VXLAN Fabric, on page 411](#)



CHAPTER 9

Configuring EVPN VXLAN External Connectivity

- [Restrictions for EVPN VXLAN External Connectivity, on page 443](#)
- [Information About EVPN VXLAN External Connectivity, on page 443](#)
- [How to Configure EVPN VXLAN External Connectivity, on page 447](#)
- [Configuration Examples for EVPN VXLAN External Connectivity, on page 459](#)

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

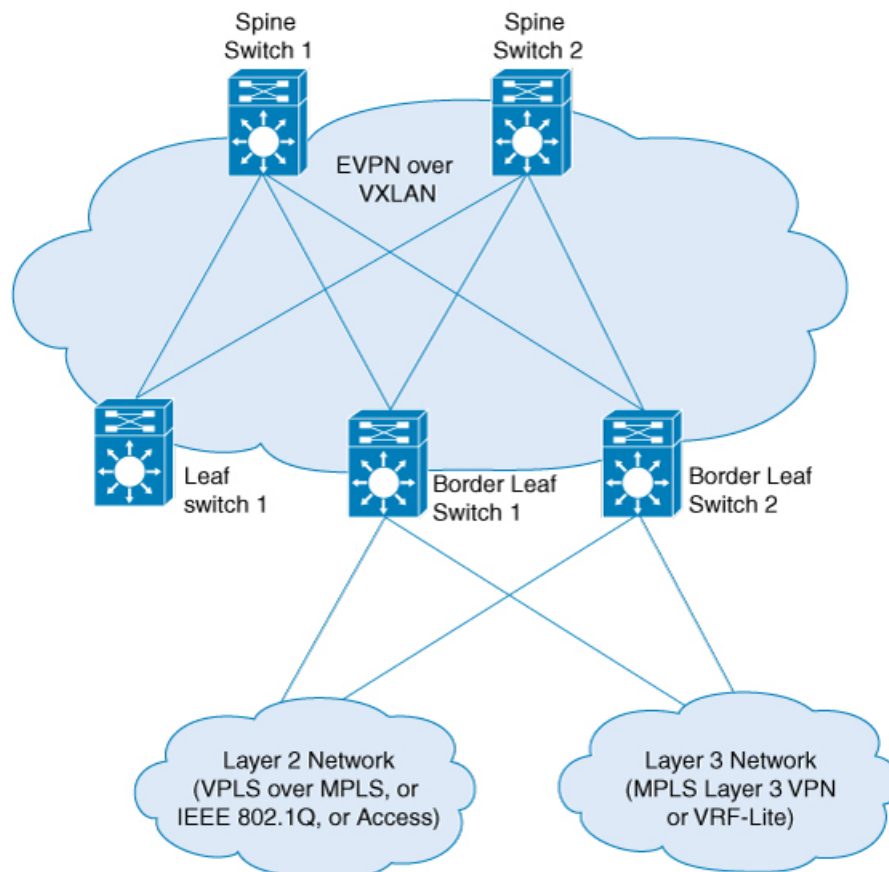
Leaf switches deployed as border nodes support the required control plane and data plane functionalities. Border leaf deployment ensures that the configuration on the spine switches is much simpler. Border leaf switches only allow communication between the external network and the VXLAN network, also 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 26: EVPN VXLAN External Connectivity Through a Border Leaf Switch



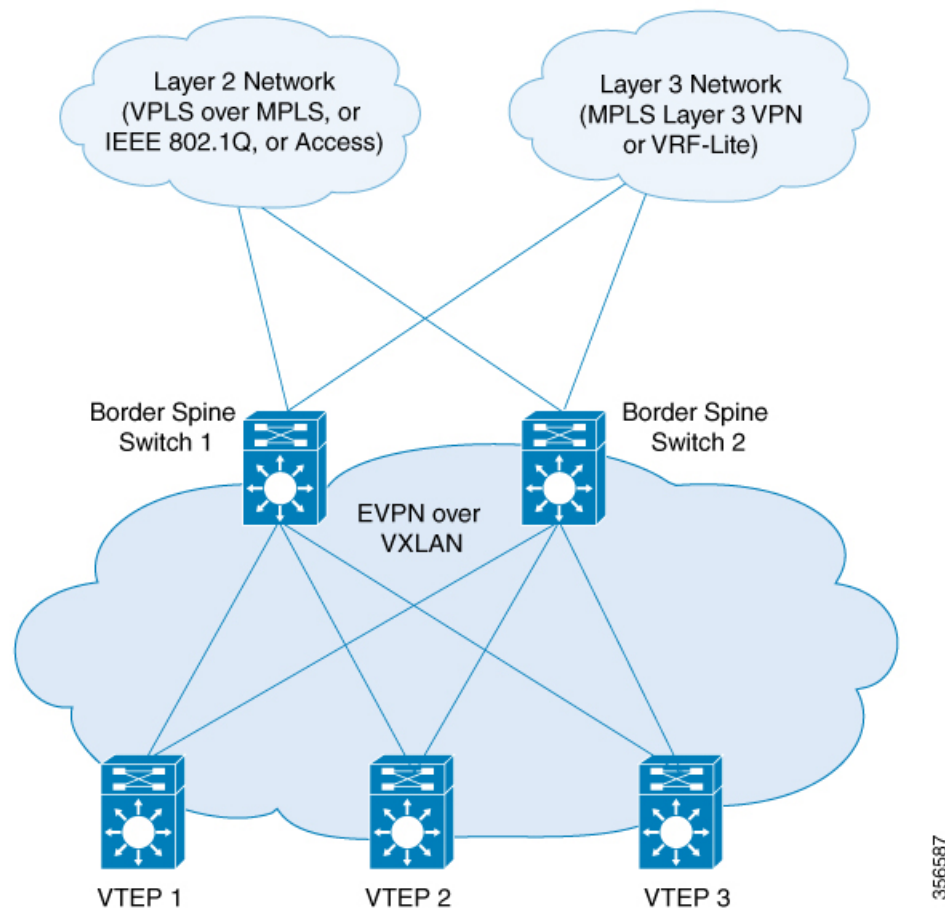
356586

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 27: EVPN VXLAN External Connectivity Through a Border Spine Switch



356587

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.

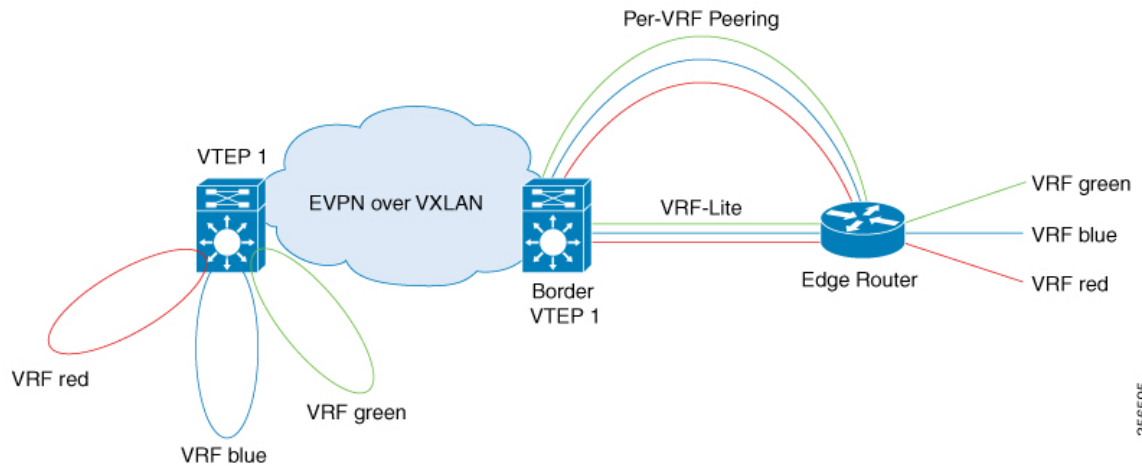


Note You must configure EVPN VXLAN Layer 2 and Layer 3 overlay networks before you configure external connectivity. See [How to Configure EVPN VXLAN Integrated Routing and Bridging, on page 99](#) for detailed steps.

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 28: Layer 3 External Connectivity with VRF-Lite



3566595

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 104 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

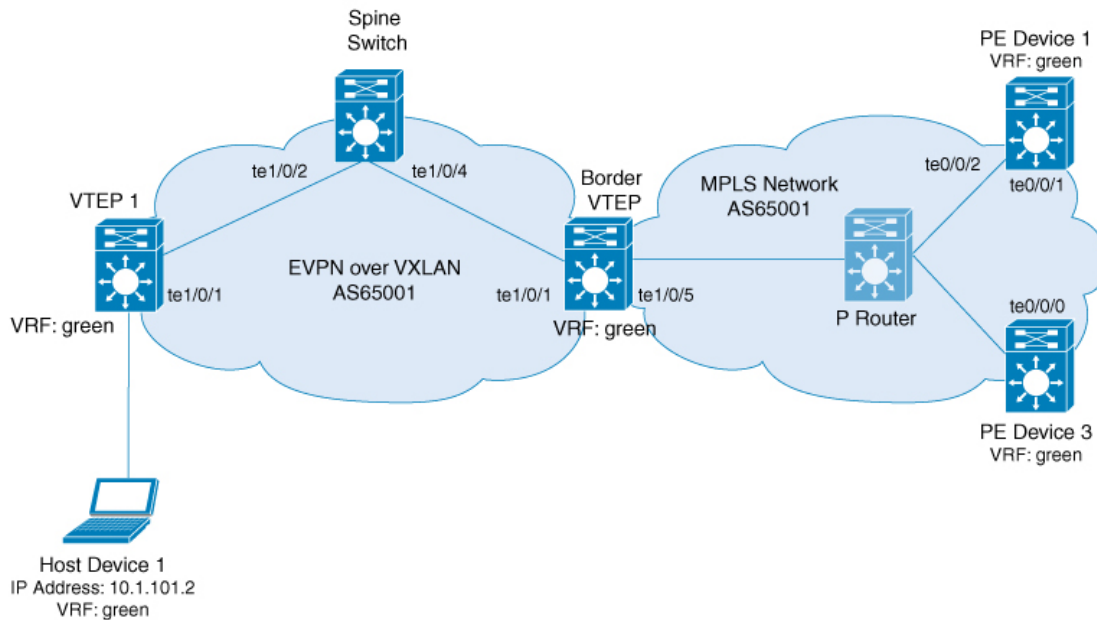
	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>interface-id</i> Example: Device(config)# interface GigabitEthernet1/0/30	Enters the interface configuration mode for the specified interface.
Step 4	vrf forwarding <i>vrf-name</i> Example: Device(config-if)# vrf forwarding green	Associates the VRF with the interface. Note The interface must be associated with the same VRF for which the Layer 3 VNI has been configured for the EVPN VXLAN network.
Step 5	ip address <i>ip-address</i> Example: Device(config-if)# ip address 192.168.3.203 255.255.255.0	Configures the IP address for the interface.
Step 6	end Example: Device(config-if)# end	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 29: 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
Step 1	<code>enable</code> Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	<code>configure terminal</code> Example: Device# <code>configure terminal</code>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 1	Enables a BGP routing process, assigns it an autonomous system number, and enters router configuration mode.
Step 4	bgp log-neighbor-changes Example: Device(config-router)# bgp log-neighbor-changes	(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> .
Step 5	bgp update-delay <i>time-period</i> Example: Device(config-router)# bgp update-delay 1	(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> .
Step 6	bgp graceful-restart Example: Device(config-router)# bgp graceful-restart	(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> .
Step 7	no bgp default ipv4-unicast Example: Device(config-router)# no bgp default ipv4-unicast	(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> .
Step 8	neighbor <i>spine-ip-address</i> remote-as <i>number</i> Example: Device(config-router)# neighbor 172.16.255.1 remote-as 1	Defines multiprotocol-BGP neighbors in the EVPN network. Use the IP address of the spine switch as the neighbor IP address. This configures the spine switch as a BGP neighbor.
Step 9	neighbor <i>mpls-peer-ip-address</i> remote-as <i>number</i> Example: Device(config-router)# neighbor 172.16.255.103 remote-as 1	Defines multiprotocol-BGP neighbors in the external MPLS network. 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.
Step 10	neighbor { <i>ip-address</i> <i>group-name</i> } update-source <i>interface</i> Example: Device(config-router)# neighbor 172.16.255.1 update-source 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.

	Command or Action	Purpose
Step 11	address-family l2vpn evpn Example: Device(config-router)# address-family l2vpn evpn	Specifies the L2VPN address family and enters address family configuration mode.
Step 12	import vpnv4 unicast re-originate Example: Device(config-router-af)# import vpnv4 unicast re-originate	Reoriginates the VPNv4 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.
Step 13	import vpnv6 unicast re-originate Example: Device(config-router-af)# import vpnv6 unicast re-originate	Reoriginates the VPNv6 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.
Step 14	neighbor ip-address activate Example: Device(config-router-af)# neighbor 10.11.11.11 activate	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 15	neighbor ip-address send-community [both extended standard] Example: Device(config-router-af)# neighbor 10.11.11.11 send-community both	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address. Note Use either of extended or both keywords. External connectivity cannot be established when you use the standard keyword.
Step 16	neighbor {ip-address peer-group-name} next-hop-self [all] Example: Device(config-router-af)# neighbor ip-address next-hop-self all	Configures the router as the next hop for a BGP-speaking neighbor or peer group. The all 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. The all keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers.
Step 17	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 18	address-family vpnv4 Example: Device(config-router)# address-family vpnv4	Specifies the VPNv4 address family and enters address family configuration mode.
Step 19	import l2vpn evpn re-originate Example: Device(config-router-af)# import l2vpn evpn re-originate	Reoriginates the EVPN routes imported from the EVPN fabric into the VPNv4 address family as VPNv4 routes and distributes them to the external network.
Step 20	neighbor ip-address activate Example: Device(config-router-af)# neighbor 172.16.255.103 activate	Enables the exchange information from a BGP neighbor. Use the IP address of the external MPLS network router as the neighbor IP address.
Step 21	neighbor ip-address send-community [both extended standard] Example: Device(config-router-af)# neighbor 172.16.255.103 send-community both	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the external MPLS network router as the neighbor IP address. Note Use either of extended or both keywords. External connectivity cannot be established when you use the standard keyword.
Step 22	neighbor {ip-address peer-group-name} next-hop-self [all] Example: Device(config-router-af)# neighbor ip-address next-hop-self all	Configures the router as the next hop for a BGP-speaking neighbor or peer group. The all 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. The all keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers.
Step 23	exit-address-family Example: Device(config-router-af)# exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 24	address-family vpnv6 Example: Device(config-router)# address-family vpnv6	Specifies the VPNv6 address family and enters address family configuration mode.

	Command or Action	Purpose
Step 25	import l2vpn evpn re-originate Example: Device(config-router-af) # import l2vpn evpn re-originate	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	neighbor ip-address activate Example: Device(config-router-af) # neighbor 172.16.255.103 activate	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 27	neighbor ip-address send-community [both extended standard] Example: Device(config-router-af) # neighbor 172.16.255.103 send-community both	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address. Note Use either of extended or both keywords. External connectivity cannot be established when you use the standard keyword.
Step 28	neighbor {ip-address peer-group-name} next-hop-self [all] Example: Device(config-router-af) # neighbor ip-address next-hop-self all	Configures the router as the next hop for a BGP-speaking neighbor or peer group. The all 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. The all keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers.
Step 29	exit-address-family Example: Device(config-router-af) # exit-address-family	Exits address family configuration mode and returns to router configuration mode.
Step 30	end Example: Device(config-router) # end	Returns to privileged EXEC mode.

Enabling EVPN VXLAN Layer 3 TRM Interworking with MVPN Networks

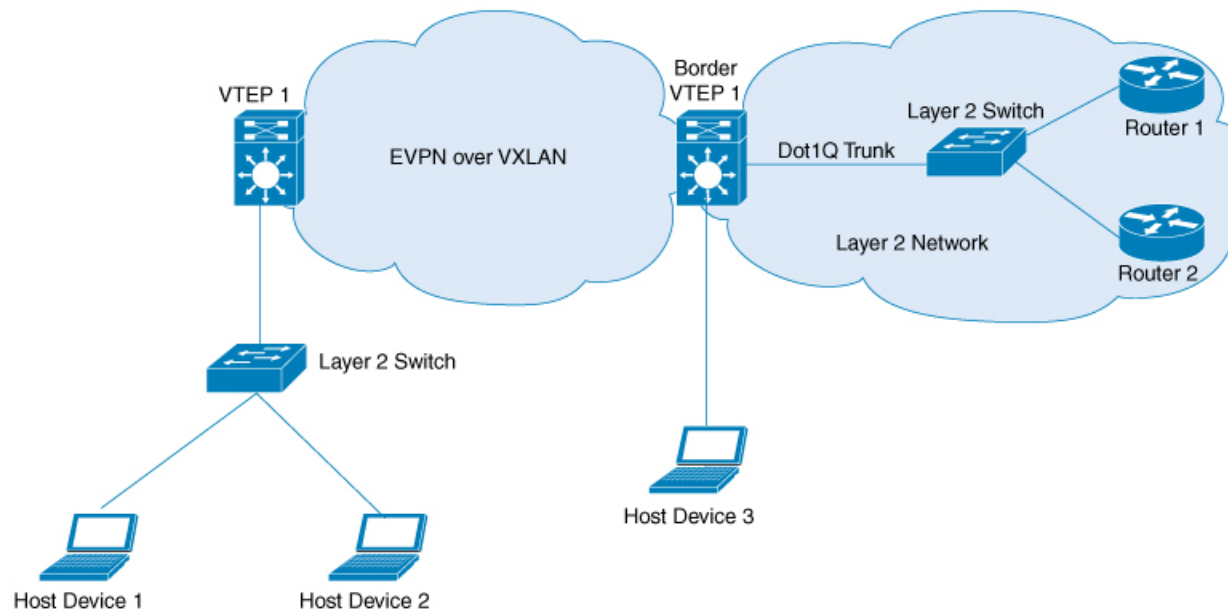
To configure interworking of Layer 3 TRM with MVPN networks, perform the following steps:

- Configure Layer 3 TRM in the BGP EVPN VXLAN fabric before you enable Layer 3 TRM interworking with MVPN. See [How to Configure Tenant Routed Multicast, on page 250](#) for detailed steps.
- Configure the MVPN network for the VPNv4 address family. See *Configuring Multicast Virtual Private Network* module of the *IP Multicast Routing Configuration Guide*.
- If internal Border Gateway Protocol (iBGP) is used for peering between the two networks, run the **mdt auto-discovery interworking vxlan-pim** in VRF configuration mode on the border VTEP.
If external Border Gateway Protocol (eBGP) is used for peering between the two networks, run the **mdt auto-discovery interworking vxlan-pim inter-as** in VRF configuration mode on the border VTEP.

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 30: Layer 2 External Connectivity with IEEE 802.1Q Networks



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

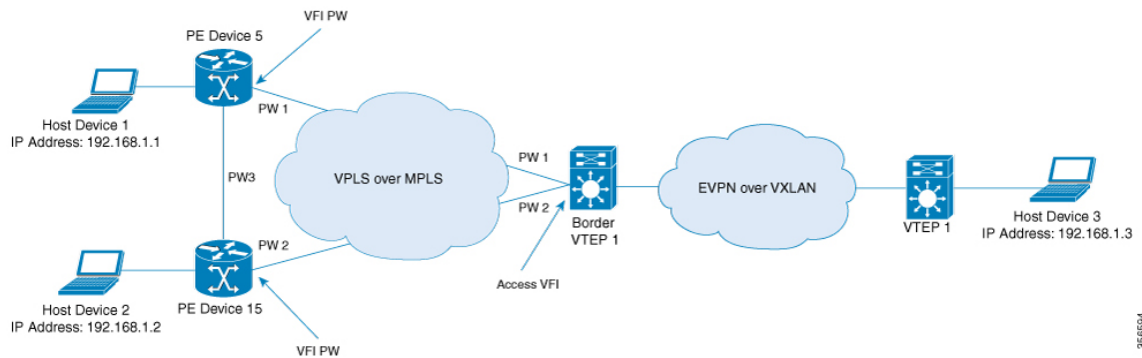
	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# <code>interface GigabitEthernet4/0/1</code>	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.
Step 4	switchport mode trunk Example: Device(config-if)# <code>switchport mode trunk</code>	Configures the interface as a trunking VLAN Layer 2 interface.
Step 5	switchport trunk allowed vlan <i>vlan-list</i> Example: Device(config-if)# <code>switchport trunk allowed vlan 201,202</code>	Sets the list of VLANs that are allowed to transmit traffic from this interface in tagged format when the interface is in trunking mode.
Step 6	end Example: Device(config-if)# <code>end</code>	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 31: Layer 2 External Connectivity with a VPLS Network Through an Access VFI



356594



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
Step 1	enable Example: Device> enable	Enters privileged EXEC mode. Enter password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	l2vpn vfi context vfi-name Example: Device(config)# l2vpn vfi context myVFI	Establishes an Layer 2 VPN VFI between two or more separate networks, and enters VFI configuration mode.
Step 4	vpn id vpn-id Example: Device(config-vfi)# vpn id 1	Configures the VPN ID for the VFI.
Step 5	member ip-address encapsulation mpls Example: Device(config-vfi)# member 10.12.12.5 encapsulation mpls	Specifies the device that forms a point-to-point Layer 2 VPN VFI connection.

	Command or Action	Purpose
Step 6	Repeat step 5 for all devices that form a point-to-point Layer 2 VPN VFI connection.	
Step 7	end Example: Device(config-vfi) # end	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
Step 1	enable Example: Device> enable	Enters privileged EXEC mode. Enter password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vlan configuration <i>vlan-number</i> Example: Device(config) # vlan configuration 11	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.
Step 4	member access-vfi <i>vfi-name</i> Example: Device(config-vlan) # member access-vfi myVFI	Adds the access VFI as a member of the VLAN configuration.
Step 5	member evpn-instance <i>evpn-instance-number</i> vni <i>l2-vni-number</i> Example: Device(config-vlan) # member evpn-instance 1 vni 6000	Adds the EVPN instance as a member of the VLAN configuration.
Step 6	end Example: Device(config-vlan) # end	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*.

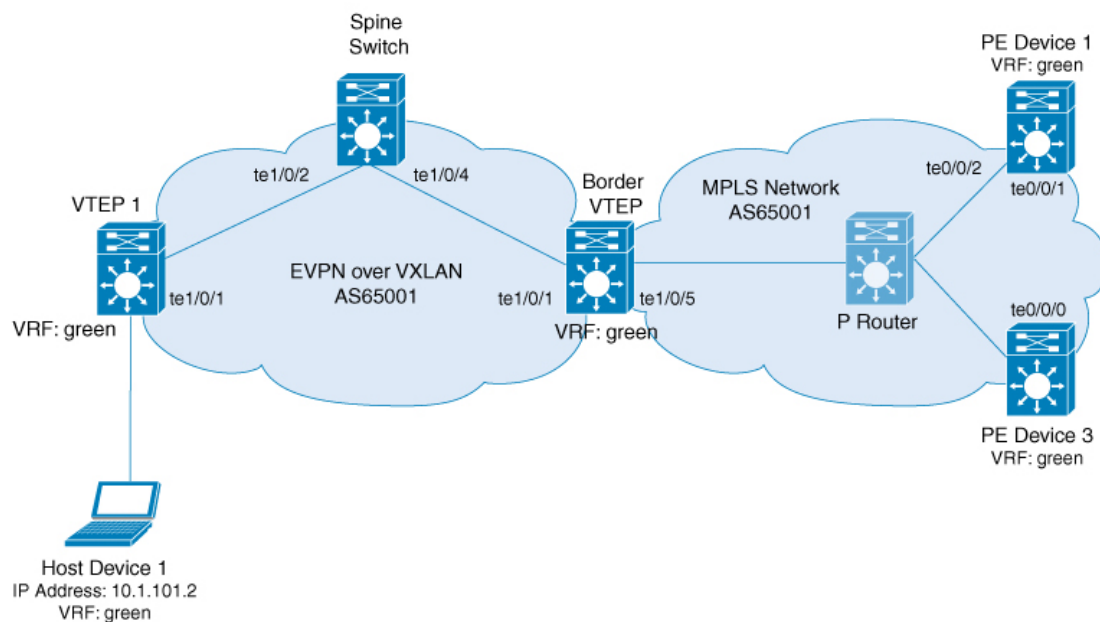
Configuration Examples for EVPN VXLAN External Connectivity

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 32: 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 43: 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 44: 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
```



```

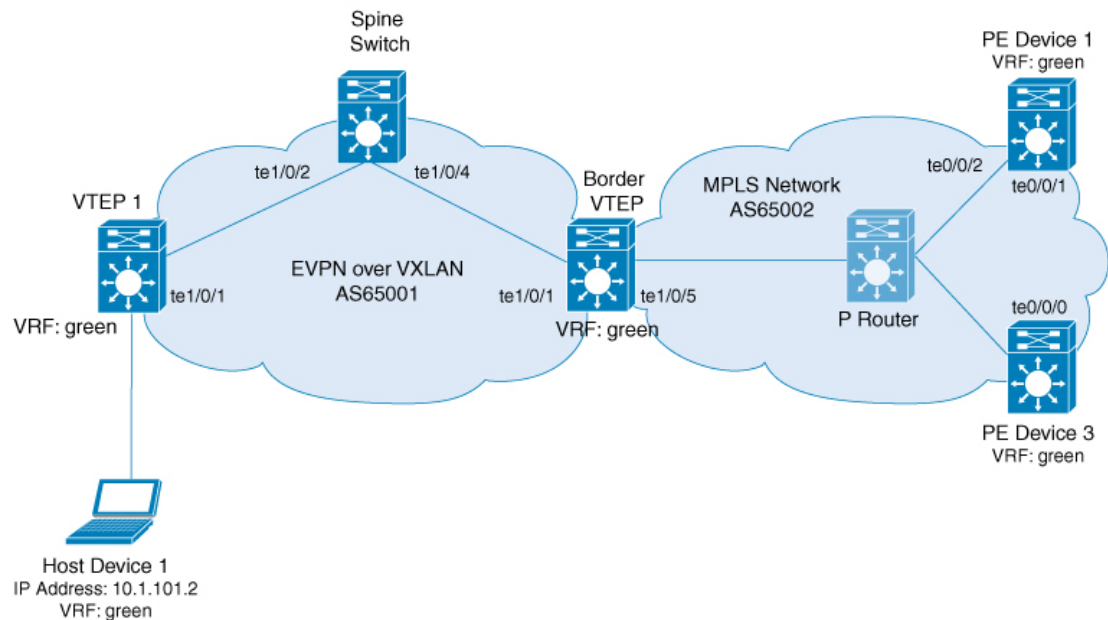
B      10.1.101.0/24 [200/0] via 172.16.255.6, 00:28:12
B      10.1.101.1/32 [200/0] via 172.16.255.6, 00:28:10
B      10.1.101.2/32 [200/0] via 172.16.255.6, 00:27:48
B      10.1.102.0/24 [200/0] via 172.16.255.6, 00:28:12
B      10.1.102.1/32 [200/0] via 172.16.255.6, 00:28:10
B      10.1.255.101/32 [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 33: 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 45: 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 46: 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 10

Configuring Multi-Homing in a BGP EVPN VXLAN Fabric

- [Restrictions for Multi-Homing in a BGP EVPN VXLAN Fabric, on page 481](#)
- [Information About Multi-Homing in a BGP EVPN VXLAN Fabric, on page 481](#)
- [How to Configure Multi-Homing in a BGP EVPN VXLAN Fabric, on page 487](#)
- [Configuration Examples for Multi-Homing in a BGP EVPN VXLAN Fabric, on page 492](#)

Restrictions for Multi-Homing in a BGP EVPN VXLAN Fabric

- Multi-homing in all-active redundancy mode is not supported.
- Multi-homing in single-active redundancy mode supports only dual-homing, which allows two nodes within a redundancy group.
- Cross-linking between host or access devices and VTEPs is not supported for a dual-homed network.
- A dual-homed network needs internal redundancy to avoid a network split.
- Provision and operational state of EVPN instances must be consistent on both dual-homed VTEPs. Inconsistencies in configuration or operational state of EVPN instances between the VTEPs leads to traffic blackholing.
- Do not configure EVPN-enabled VLAN and non-EVPN-enabled VLAN on an ethernet segment enabled trunk interface. This is because spanning tree protocol (STP) is disabled at the interface level when an ethernet segment is enabled, and may cause Layer 2 loops in non-EVPN-enabled VLANs.

Information About Multi-Homing in a BGP EVPN VXLAN Fabric

Multi-homing feature in a BGP EVPN VXLAN fabric provides redundancy in the connection between a host or Layer 2 switch and the EVPN VXLAN network.

In a BGP EVPN VXLAN fabric, you connect a host or Layer 2 switch to the EVPN VXLAN network either through single-homing or through multi-homing.

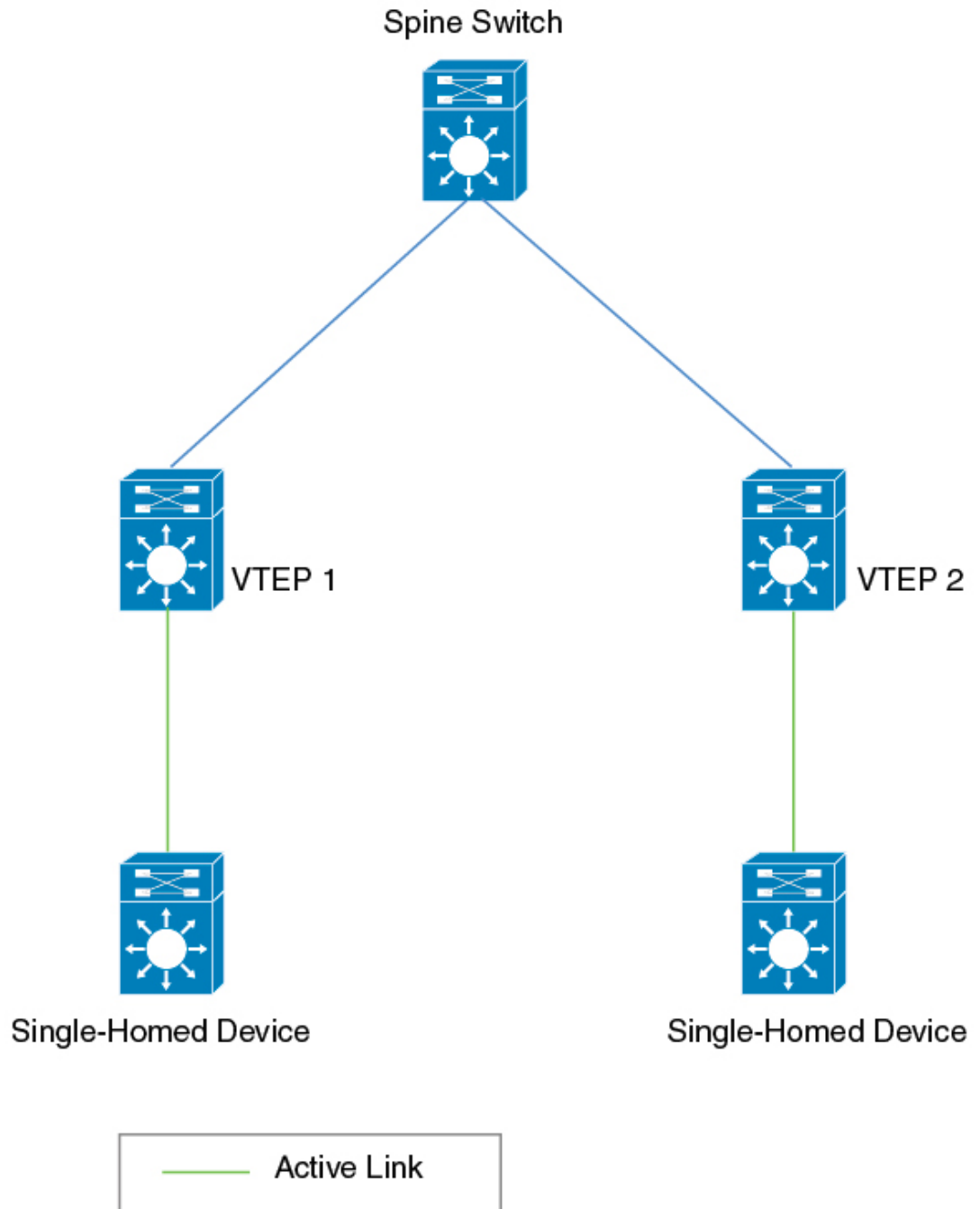
Cisco Catalyst 9000 Series switches support [RFC 7432](#) and [RFC 8365](#) for VXLAN encapsulation-based EVPN multi-homing capabilities.

Single-Homing

Single-homing allows you to connect a host or Layer 2 switch to a single VTEP in the EVPN VXLAN network. Single-homing does not support redundancy in the connection between the host or access device and the VTEP. When the active link breaks down, the connection between the host (or Layer 2 switch) and the VTEP is lost. As a result, single-homed topologies are not always reliable and efficient.

The following figure shows a single-homed topology:

Figure 34: Single-Homed Topology



357247

Multi-Homing

Multi-homing allows you to connect a host or Layer 2 switch to more than one VTEP in the EVPN VXLAN network. This connection provides redundancy and allows network optimization. Redundancy in the connection

with the VTEPs ensures that there's no traffic disruption when there's a network failure. Multi-homed topologies are more reliant, secure, and efficient than single-homed topologies.

Multi-homing operates in single-active and all-active redundancy modes. In both modes, the connected host or access device is represented by an ethernet segment ID. This ethernet segment ID must also be part of the configuration on the VTEP's interface that connects the multihomed host or network device. All traffic forwarded between the VTEPs and the host (or Layer 2 switch) passes through this ethernet segment.

Single-Active Redundancy Mode

In single-active redundancy mode, only a single VTEP, among a group of VTEPs that are attached to a particular Ethernet-segment, is allowed to forward traffic to and from the Ethernet segment. It results in a single-active access link between the VTEPs and the host (or Layer 2 switch) that passes through the Ethernet segment. The single access link can either be a physical link or an ether-channel.

Multi-homing in single-active redundancy mode is supported only in the form of dual-homing. Dual-homing allows a host or access device to be connected to only two VTEPs. A dual-homed topology with single-active redundancy can be deployed in one of the following ways:

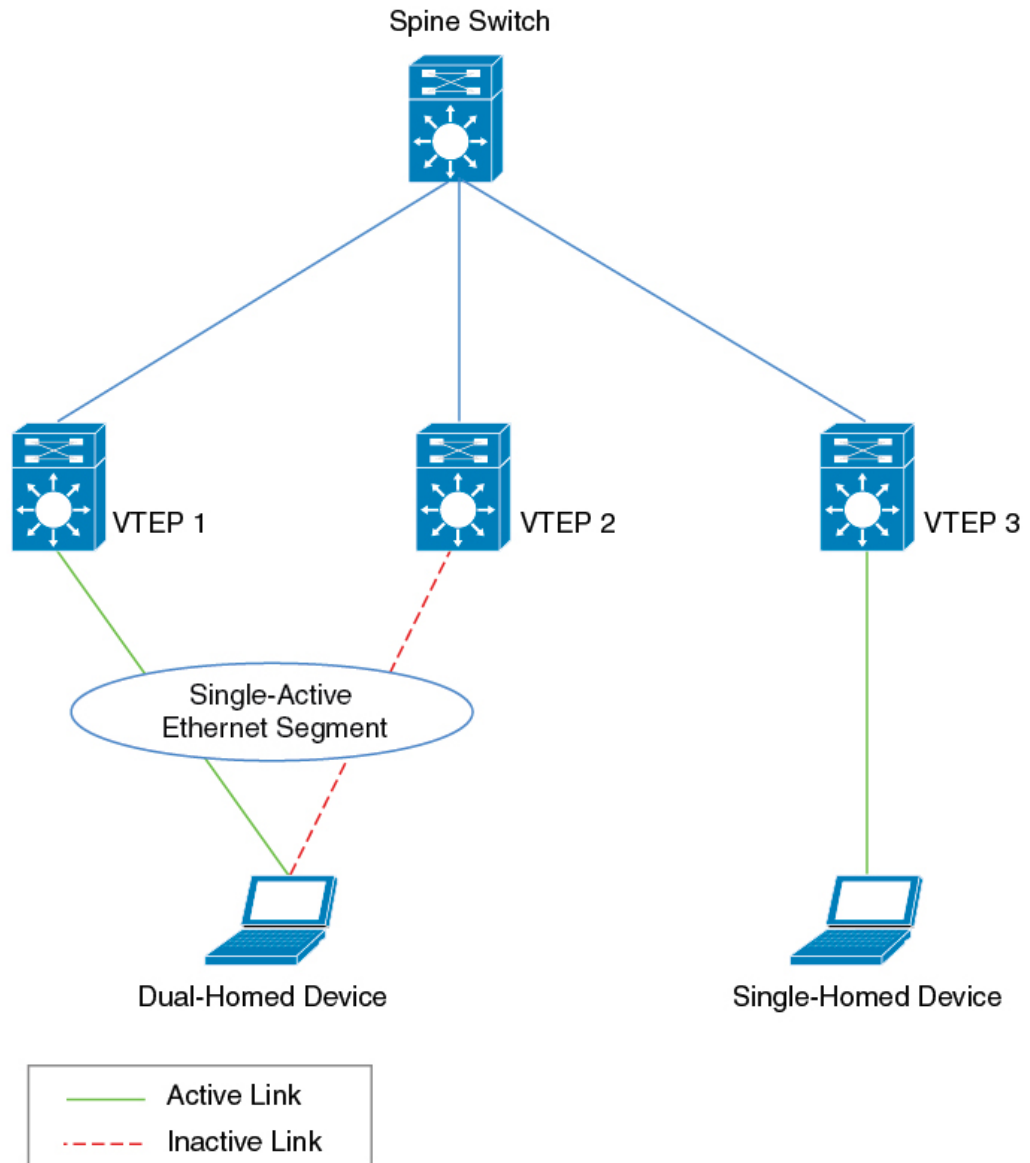
- Dual-homed device
- Dual-homed network

Dual-Homed Device

In single-active dual-homed device topology, a single host or access device is connected to two VTEPs with two links that pass through a single-active Ethernet segment. The Ethernet segment contains two separate links connecting the host or access device to each VTEP, but only one link remains active at any given time. For each VLAN interface on a dual-homed host or access device, only one link remains active. When the active link breaks down, the back-up link takes over and ensures constant connectivity.

The following figure shows a dual-homed device topology:

Figure 35: Dual-Homed Device Topology



357245

Dual-Homed Network

In single-active dual-homed network topology, two host or access devices from the same network are connected to two separate VTEPs through links that pass through a single-active Ethernet segment. At any given time, only one of these links remains active. When the active link breaks down, the back-up link takes over and ensures constant connectivity. The two host or access devices are part of a dual-homed network.

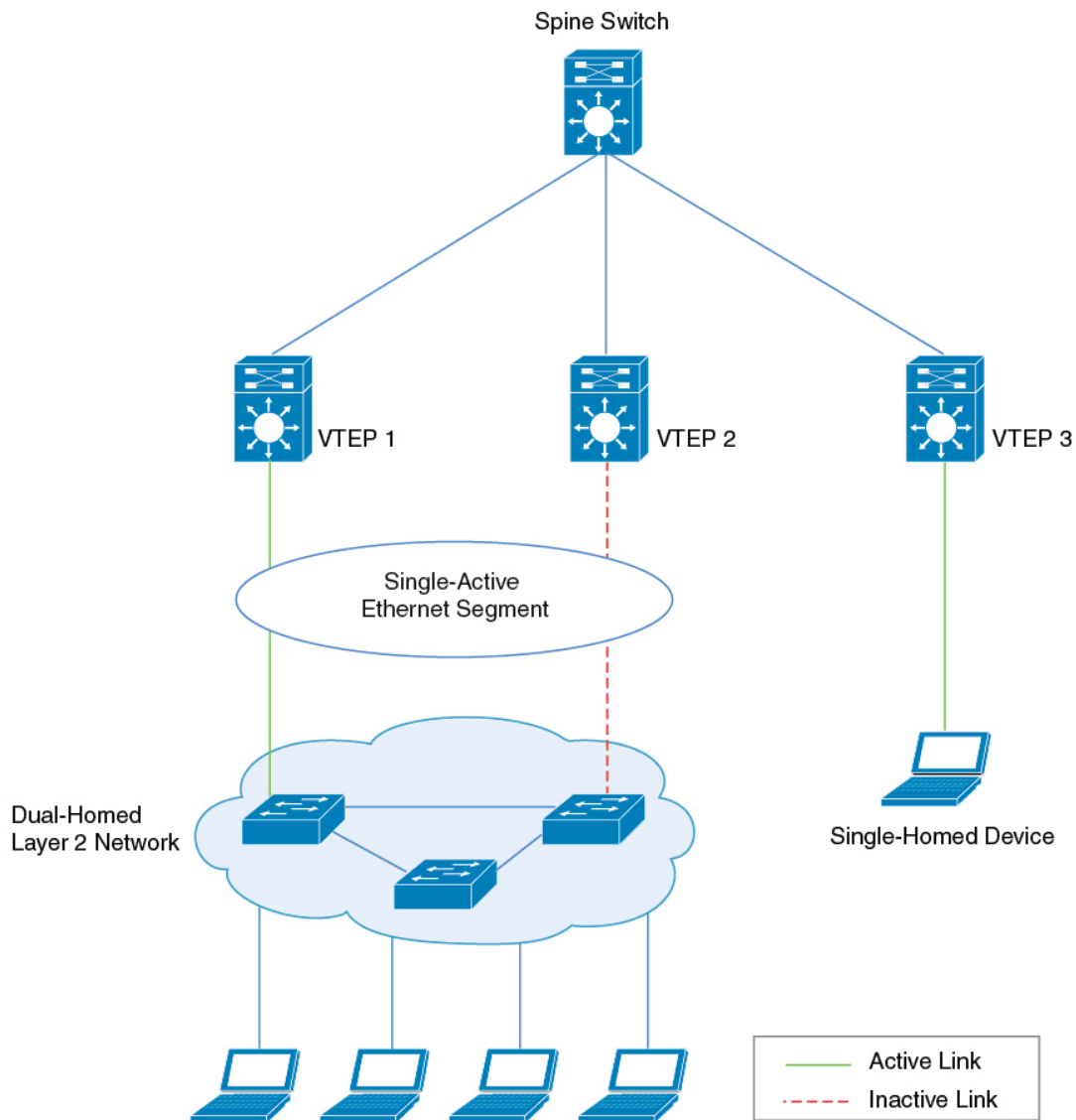
A dual-homed network topology results in a situation where the network splits into two different networks if the connectivity between the host or access devices is lost. To avoid this scenario, redundancy must also be enabled within the dual-homed network.

The following figure shows a dual-homed network topology:



Note Ensure that you enable a spanning tree within the dual-homed Layer 2 network.

Figure 36: Dual-Homed Network Topology



DF Election and Load Balance

A dual-homed network with a single-active Ethernet segment uses a Designated Forwarder (DF) election mechanism to load balance the traffic. The DF election is made at the Layer 2 VNI level, when the access interface from the VTEP is a trunk interface and an Ethernet segment is configured.

In the above topology, some Layer 2 VNIs use the interface connected to VTEP 1 as the active link and the others use the interface connected to VTEP 2 as the active link. This allows effective utilization of bandwidth

357246

on both the interfaces in a steady network state. Traffic in each Layer 2 VNI is load balanced for the downstream dual-homed Layer 2 network. If any of the physical interface link to the downstream Layer 2 device goes down and is not operational, the DF election algorithm recalculates the active link interface. After the link is reestablished and both links are operational again, the DF election algorithm restores the load balancing operation to utilize the bandwidth of both the links effectively.

Migration Between Single-Homed and Multi-Homed Network Topologies

BGP EVPN VXLAN allows you to migrate your network topology from one redundancy mode to another. You can transition from a single-homed topology to a multi-homed topology. Likewise, you can also remove the redundancy from a multi-homed topology to move back to a single-homed topology.



Note When you migrate from one topology to another, ensure you make corresponding changes to the Ethernet segment configuration. If you change either of the two without making corresponding changes to the other, it results in traffic loops and traffic blackholing.

For a sample illustration and detailed steps about how to migrate from a single-homed topology to a single-active dual-homed topology, see [Migrating from a Single-Homed Topology to a Single-Active Dual-Homed Topology, on page 489](#).

For a sample illustration and detailed steps about how to migrate from a single-active dual-homed topology to a single-homed topology, see [Migrating from a Single-Active Dual-Homed Topology to a Single-Homed Topology, on page 491](#).

How to Configure Multi-Homing in a BGP EVPN VXLAN Fabric

Before you configure multi-homing in a BGP EVPN VXLAN fabric, ensure that you configure EVPN VXLAN Layer 2 and Layer 3 overlay networks. See [How to Configure EVPN VXLAN Integrated Routing and Bridging, on page 99](#) for detailed steps.

Configuring Dual-Homing with Single Active Redundancy in a BGP EVPN VXLAN Fabric

To configure dual-homing with single-active redundancy in a BGP EVPN VXLAN fabric, perform the following set of procedures:

Configuring Redundancy on an Ethernet Segment

To configure redundancy on an ethernet segment, perform the following steps :

Procedure

	Command or Action	Purpose
Step 1	enable Example:	Enters privileged EXEC mode. Enter password, if prompted.

	Command or Action	Purpose
	Device> enable	
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	l2vpn evpn ethernet-segment <i>ethernet-segment-id</i> Example: Device(config)# l2vpn evpn ethernet-segment 1	Enters Layer 2 VPN EVPN ethernet segment configuration mode.
Step 4	identifier type { 0 esi-value 3 system-mac mac-address } Example: Device(config-evpn-es)# identifier type 0 0.0.0.0.0.0.0.0.1	Configures the ethernet segment identifier type (ESI) and value for the ethernet segment. The following ESI types are supported: <ul style="list-style-type: none"> • Type 0: This type indicates an arbitrary 9-octet ESI value. The format is 00 + 9-octets of ESI value. • Type 3: This type indicates a MAC-based ESI Value. The format is 03 + system-mac (6 bytes) + value of MAC address (3 bytes).
Step 5	redundancy <i>redundancy-type</i> Example: Device(config-evpn-es)# redundancy single-active	Configures the redundancy type for the ethernet segment.
Step 6	df-election wait-time <i>time-period</i> Example: Device(config-evpn-es)# df-election wait-time 1	(Optional) Configures the designated forwarder (DF) election wait time for the ethernet segment. The range is 1 to 10 seconds. The default wait time is 3 seconds.
Step 7	end Example: Device(config-evpn-es)# end	Exits Layer 2 VPN EVPN ethernet segment configuration mode and enters privileged EXEC mode.

Associating an Ethernet Segment with an Interface on a VTEP

To associate the ethernet segment with an interface on a VTEP, perform the following steps :

Procedure

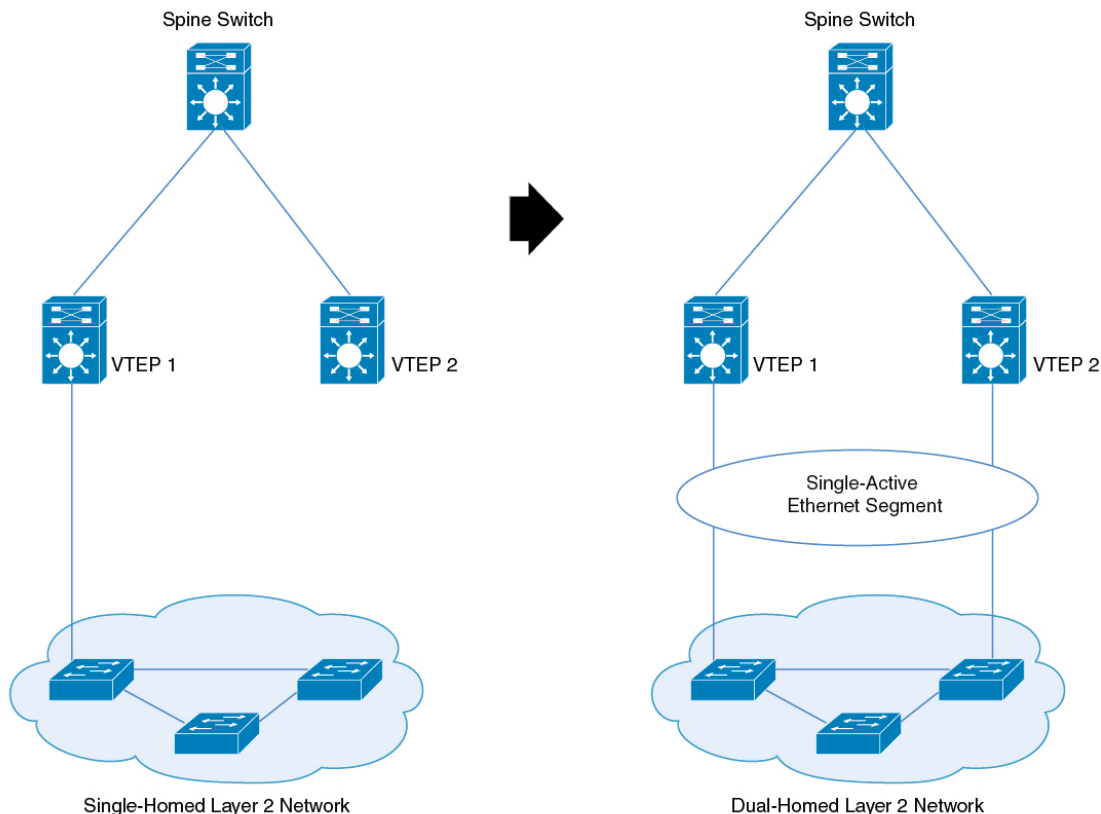
	Command or Action	Purpose
Step 1	enable	Enters privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	Enter password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface GigabitEthernet1/0/10	Specifies the interface, and enters interface configuration mode.
Step 4	evpn ethernet-segment <i>ethernet-segment-id</i> Example: Device(config-if)# evpn ethernet-segment 1	Associates the specified Ethernet segment with the interface. Each Ethernet segment is represented by a unique Ethernet segment ID. Note Ensure that you configure a unique Ethernet segment ID on any interface. Ensure that you configure the same segment ID on the link that connects the second VTEP and the dual-homed device (the second link through the Ethernet segment).
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode and enters privileged EXEC mode.

Migrating from a Single-Homed Topology to a Single-Active Dual-Homed Topology

The following figures illustrate the migration from a single-homed topology to a single-active dual-homed topology:

Figure 37: Migration from a Single-Homed Network to a Single-Active Dual-Homed Network



To migrate from a single-homed network to a single-active dual-homed network, perform the following steps:



Note When you migrate from one topology to another, ensure you make corresponding changes to the Ethernet segment configuration. If you change either of the two without making corresponding changes to the other, it results in traffic loops and traffic blackholing.

1. Before you migrate, we recommend that you do not configure the VTEP as the root bridge of the spanning tree, as the provision of Ethernet segment on the interface of a VTEP excludes it from spanning-tree. If the VTEP is the root bridge, its exclusion from the spanning-tree triggers an immediate spanning tree re-convergence.



Note Do not activate a link between VTEP 2 and a switch in the single-homed network yet. Activate the second only once you configure the Ethernet segment. In case a link is already activated, ensure that you deactivate the link.

2. Provision the Ethernet segment on the interface of the VTEP that has the active link. Provision of the Ethernet segment updates all the MAC addresses that are locally learned on that interface with the Ethernet segment ID of the interface.

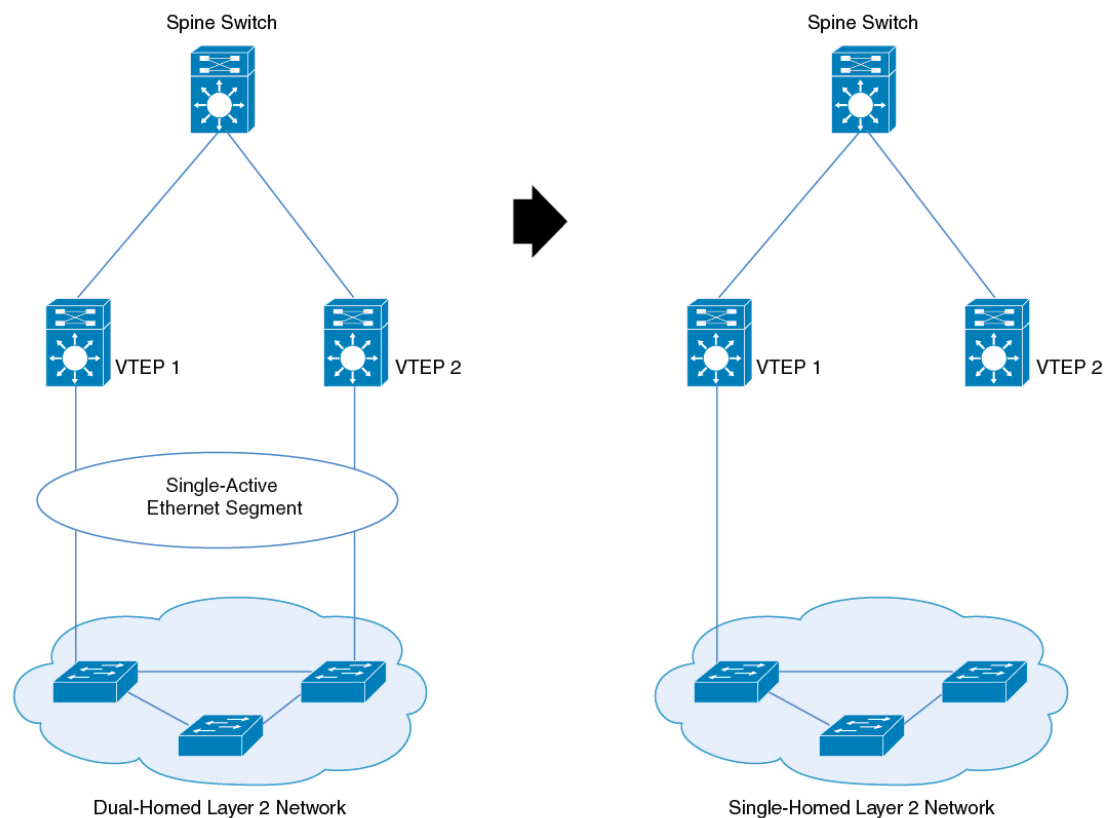
3. Now provision the Ethernet segment on the interface of the second VTEP that needs to be connected to the switch in the single-homed network.
4. Connect the link and bring up the interface of the second VTEP. By doing this, you trigger fast convergence, Ethernet segment auto-discovery, and DF reelection.

The single-homed network has now migrated to a dual-homed network.

Migrating from a Single-Active Dual-Homed Topology to a Single-Homed Topology

The following figures illustrate the migration from a single-active dual-homed topology to a single-homed topology:

Figure 38: Migration from a Single-Active Dual-Homed Network to a Single-Homed Network



To migrate from a single-active dual-homed network to a single-homed network, perform the following steps:



Note When you migrate from one topology to another, ensure you make corresponding changes to the Ethernet segment configuration. If you change either of the two without making corresponding changes to the other, it results in traffic loops and traffic blackholing.



Note Ensure that the Ethernet segment remains configured on the dual-homed links as long as the links are up. If the Ethernet segment is removed from an active link, it causes traffic loops.

1. Before you migrate, we recommended that you configure portfast on the link you activate. Removal of Ethernet segment from an interface on the VTEP puts it back into the spanning-tree. If the interface is not configured with portfast, the port goes through block-learn-forward states and causes extensive traffic loss.
2. Shut down the interface that needs to be decommissioned. When you shut down the interface, you trigger fast convergence, Ethernet segment auto-discovery, and DF reelection. As a result, all the traffic converges into the active link.
3. Remove the Ethernet segment from the decommissioned interface. Ensure that the interface is down before you disconnect the link.

The dual-homed network has now migrated to a single-homed network with an Ethernet segment.

4. (Optional) Remove the Ethernet segment from the interface with the active link on the VTEP.

Removal of the Ethernet segment updates all the MAC addresses that are locally learned on that interface without the Ethernet segment ID.

The Ethernet segment is now removed from the single-homed network homed network.

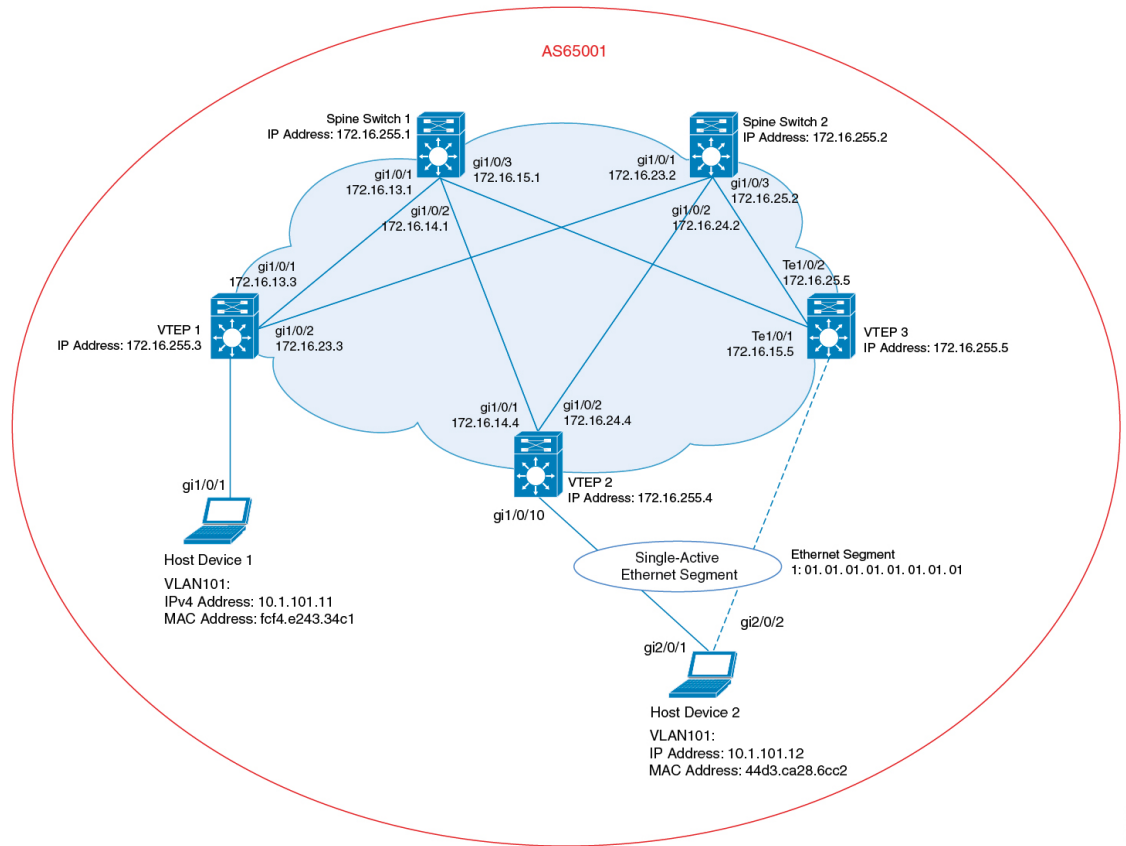
Configuration Examples for Multi-Homing in a BGP EVPN VXLAN Fabric

This section provides configuration examples for multi-homing in a BGP EVPN VXLAN fabric:

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

This example shows how to configure and verify dual-homing with single-active redundancy in a BGP EVPN VXLAN fabric for the following topology:

Figure 39: Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric



The topology shows an EVPN VXLAN network with two spine switches (Spine Switch 1 and Spine Switch 2) and three VTEPs (VTEP 1, VTEP 2, and VTEP 3). Host Device 1 is connected to VTEP 1. Host Device 2 is connected to VTEP 2 and VTEP 3 as a dual-homed single-active connection that passes through Ethernet Segment 1.



Note Ensure that you configure a unique Ethernet segment ID on any interface in the fabric. If an Ethernet segment ID is associated with one of the connecting links passing through the segment, associate the same Ethernet segment ID with the second link.



Note Do not configure a unique Ethernet segment ID per EVPN instance or VLAN or virtual network instance (VNI). For example purpose, EVPN instance 101 is used in the [Verifying Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric, on page 499](#) section.

Table 47: Configuring Dual-Homing with Single-Active Redundancy using VTEP 2 and VTEP 3

VTEP 2	VTEP 3
<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 ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn ethernet-segment 1 identifier type 0 01.01.01.01.01.01.01.01 redundancy single-active ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type ingress ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.4 255.255.255.255 ip ospf 1 area 0 ! </pre>	<pre> LEaf-03# show running-config hostname Leaf-03 ! 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 ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn ethernet-segment 1 identifier type 0 01.01.01.01.01.01.01.01 redundancy single-active ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type ingress ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 901 member vni 50901 ! interface Loopback0 ip address 172.16.255.5 255.255.255.255 ip ospf 1 area 0 ! </pre>

VTEP 2	VTEP 3
<pre> interface Loopback1 ip address 172.16.254.4 255.255.255.255 ip ospf 1 area 0 ! interface GigabitEthernet0/0 vrf forwarding Mgmt-vrf ip address 10.62.149.182 255.255.255.0 negotiation auto ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.14.4 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.24.4 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access evpn ethernet-segment 1 spanning-tree portfast ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 no autostate ! 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 ingress-replication member vni 50901 vrf green </pre>	<pre> interface Loopback1 ip address 172.16.254.5 255.255.255.255 ip ospf 1 area 0 ! interface GigabitEthernet0/0 vrf forwarding Mgmt-vrf ip address 10.62.149.183 255.255.255.0 negotiation auto ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.15.5 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.25.5 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access evpn ethernet-segment 1 spanning-tree portfast ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! 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 ingress-replication member vni 50901 vrf green </pre>

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

VTEP 2	VTEP 3
<pre> ! 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 ! end ! Leaf-02# </pre>	<pre> ! 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 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 ! end ! Leaf-03# </pre>

Table 48: Configuring Spine Switch 1, Spine Switch 2, and VTEP 1 to Configure Dual-Homing with Single-Active Redundancy

Spine Switch 1	Spine Switch 2	VTEP 1
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! system mtu 9198 ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ! interface GigabitEthernet0/0 vrf forwarding Mgmt-vrf ip address 10.62.149.180 255.255.255.0 negotiation auto ! 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 ! </pre>	<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! system mtu 9198 ! interface Loopback0 ip address 172.16.255.1 255.255.255.255 ip ospf 1 area 0 ! interface GigabitEthernet0/0 vrf forwarding Mgmt-vrf ip address 10.62.149.180 255.255.255.0 negotiation auto ! 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 ! </pre>	<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 ! l2vpn evpn replication-type static router-id Loopback1 default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan replication-type ingress ! ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 ! 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 ospf 1 area 0 ! ! </pre>

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2	VTEP 1
<pre> router bgp 65001 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 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 ! 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 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 ! end ! Spine-01# </pre>	<pre> router bgp 65001 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 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 ! 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 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 ! end ! Spine-02# </pre>	<pre> interface GigabitEthernet0/0 vrf forwarding Mgmt-vrf ip address 10.62.149.179 255.255.255.0 negotiation auto ! interface GigabitEthernet1/0/1 no switchport ip address 172.16.13.3 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/2 no switchport ip address 172.16.23.3 255.255.255.0 ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/10 switchport access vlan 101 switchport mode access spanning-tree portfast ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 ! 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 ingress-replication member vni 50901 vrf green ! router ospf 1 router-id 172.16.255.3 ! </pre>

Spine Switch 1	Spine Switch 2	VTEP 1
		<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 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 ! end ! Leaf-01# </pre>

Verifying Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

The following sections provide sample outputs for **show** commands to verify dual-homing with single-active redundancy on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 499](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 502](#)
- [Outputs to Verify the Configuration on VTEP 3, on page 505](#)
- [Outputs to Verify the Configuration on Spine Switch 1, on page 509](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 511](#)

Outputs to Verify the Configuration on VTEP 1

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

```

Leaf-01# show nve peer
Interface  VNI      Type Peer-IP      RMAC/Num_RTs  eVNI      state flags UP time
nve1      50901    L3CP 172.16.254.5  7c21.0dbd.2748 50901      UP   A/M/4 01:17:04
nve1      50901    L3CP 172.16.254.4  7c21.0dbd.9548 50901      UP   A/M/4 03:26:09
nve1      10101    L2CP 172.16.254.4   8           10101      UP   N/A   03:52:15

```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

```
nve1      10101    L2CP 172.16.254.5    10                10101    UP    N/A    05:25:28

Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn evi 101 detail
EVPN instance:      101 (VLAN Based)
RD:                 172.16.254.3:101 (auto)
Import-RTs:        65001:101
Export-RTs:        65001:101
Per-EVI Label:     none
State:              Established
Replication Type:  Ingress
Encapsulation:     vxlan
IP Local Learn:    Enabled (global)
Adv. Def. Gateway: Enabled (global)
Vlan:              101
  Ethernet-Tag:    0
  State:           Established
  Core If:         Vlan901
  Access If:       Vlan101
  NVE If:          nve1
  RMAC:            10b3.d56a.8fc8
  Core Vlan:       901
  L2 VNI:          10101
  L3 VNI:          50901
  VTEP IP:         172.16.254.3
  VRF:             green
  IPv4 IRB:        Enabled
  IPv6 IRB:        Disabled
Pseudoports:
  GigabitEthernet1/0/10 service instance 101
    Routes: 1 MAC, 1 MAC/IP
Peers:
  172.16.254.4
    Routes: 4 MAC, 2 MAC/IP, 1 IMET, 1 EAD
  172.16.254.5
    Routes: 6 MAC, 2 MAC/IP, 1 IMET, 1 EAD

Leaf-01#
```

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

```
Leaf-01# show bgp l2vpn evpn evi 101
BGP table version is 6958, local router ID is 172.16.255.3
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: 172.16.254.3:101
*>i  [1][172.16.254.3:101][000101010101010101][0]/23
      172.16.254.5          0      100      0 ?
*mi  172.16.254.4          0      100      0 ?
*>  [2][172.16.254.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
      ::                      32768 ?
```



```
*>i [2] [172.16.254.3:101] [0] [48] [44D3CA286C82] [0] [*]/20
      172.16.254.5          0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [44D3CA286CC2] [0] [*]/20
      172.16.254.5          0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [7C210DBD2741] [32] [10.1.101.1]/24
      172.16.254.5          0      100      0 ?
*>i [2] [172.16.254.3:101] [0] [48] [7C210DBD9541] [32] [10.1.101.1]/24
      172.16.254.4          0      100      0 ?
*> [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [0] [*]/20
      ::                      32768 ?
*> [2] [172.16.254.3:101] [0] [48] [F4CFE24334C1] [32] [10.1.101.11]/24
      ::                      32768 ?
*> [3] [172.16.254.3:101] [0] [32] [172.16.254.3]/17
      ::                      32768 ?
*>i [3] [172.16.254.3:101] [0] [32] [172.16.254.4]/17
      172.16.254.4          0      100      0 ?
*>i [3] [172.16.254.3:101] [0] [32] [172.16.254.5]/17
      172.16.254.5          0      100      0 ?
```

Leaf-01#

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

```
Leaf-01# show l2route evpn mac
  EVI      ETag  Prod   Mac Address                Next Hop(s)  Seq Number
-----
  101      0    L2VPN 10b3.d56a.8fc1            V1101:0      0
  101      0    BGP   44d3.ca28.6c82            V:10101 172.16.254.5  0
  101      0    BGP   44d3.ca28.6cc2            V:10101 172.16.254.5  0
  101      0    BGP   7c21.0dbd.2741            V:10101 172.16.254.5  0
  101      0    BGP   7c21.0dbd.9541            V:10101 172.16.254.4  0
  101      0    L2VPN f4cf.e243.34c1            Gi1/0/10:101 0
```

Leaf-01#

The following example shows the output for the **show l2route evpn mac esi ethernet-segment-id** command on VTEP 1:

```
Leaf-01# show l2route evpn mac esi 0001.0101.0101.0101.0101
  EVI      ETag  Prod   Mac Address                Next Hop(s)  Seq Number
-----
  101      0    BGP   44d3.ca28.6c82            V:10101 172.16.254.5  0
  101      0    BGP   44d3.ca28.6cc2            V:10101 172.16.254.5  0
```

Leaf-01#

The following example shows the output for the **show l2route evpn mac esi ethernet-segment-id detail** command on VTEP 1:

```
Leaf-01# show l2route evpn mac esi 0001.0101.0101.0101.0101 detail
EVPN Instance:          101
Ethernet Tag:           0
Producer Name:          BGP
MAC Address:            44d3.ca28.6c82
Num of MAC IP Route(s): 0
Sequence Number:        0
ESI:                    0001.0101.0101.0101.0101
Flags:                  B()
Next Hop(s):            V:10101 172.16.254.5
Resolved Next Hops:     V:10101 172.16.254.5, V:10101 172.16.254.4
Resolved Redundancy Mode: Single-Active
```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

```

EVPN Instance:          101
Ethernet Tag:           0
Producer Name:          BGP
MAC Address:            44d3.ca28.6cc2
Num of MAC IP Route(s): 0
Sequence Number:       0
ESI:                   0001.0101.0101.0101.0101
Flags:                  B()
Next Hop(s):            V:10101 172.16.254.5
Resolved Next Hops:     V:10101 172.16.254.5, V:10101 172.16.254.4
Resolved Redundancy Mode: Single-Active

```

```
Leaf-01#
```

Return to [Verifying Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric](#), on page 499.

Outputs to Verify the Configuration on VTEP 2

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

```

Leaf-02# show nve peer
Interface  VNI      Type Peer-IP           RMAC/Num_RTs  eVNI      state flags UP time
nve1      50901    L3CP 172.16.254.3     10b3.d56a.8fc8 50901      UP   A/M/4 03:24:45
nve1      50901    L3CP 172.16.254.5     7c21.0dbd.2748 50901      UP   A/M/4 01:15:39
nve1      10101    L2CP 172.16.254.3     5              10101     UP   N/A   03:24:45
nve1      10101    L2CP 172.16.254.5     6              10101     UP   N/A   03:24:45

```

```
Leaf-02#
```

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

```

Leaf-02# show l2vpn evpn ethernet-segment detail
EVPN Ethernet Segment ID: 0001.0101.0101.0101.0101
Interface:                 Gi1/0/10
Redundancy mode:           single-active
DF election wait time:     3 seconds
Split Horizon label:       0
State:                     Ready
Encapsulation:             vxlan
Ordinal:                   0
RD:                        172.16.254.4:7
Export-RTs:                65001:101
Forwarder List:            172.16.254.4 172.16.254.5

```

```
Leaf-02#
```

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

```

Leaf-02# show l2vpn evpn evi 101 detail
EVPN instance:            101 (VLAN Based)
RD:                       172.16.254.4:101 (auto)
Import-RTs:               65001:101
Export-RTs:               65001:101
Per-EVI Label:            none
State:                    Established
Replication Type:         Ingress

```

```

Encapsulation:      vxlan
IP Local Learn:     Enabled (global)
Adv. Def. Gateway: Enabled (global)
Vlan:               101
  Ethernet-Tag:     0
  State:            Established
  Core If:          Vlan901
  Access If:        Vlan101
  NVE If:           nve1
  RMAC:             7c21.0dbd.9548
  Core Vlan:        901
  L2 VNI:           10101
  L3 VNI:           50901
  VTEP IP:          172.16.254.4
  VRF:              green
  IPv4 IRB:         Enabled
  IPv6 IRB:         Disabled
Pseudoports:
  GigabitEthernet1/0/10 service instance 101 (DF state: blocked)
    Routes: 0 MAC, 0 MAC/IP
Peers:
  172.16.254.3
    Routes: 2 MAC, 2 MAC/IP, 1 IMET, 0 EAD
  172.16.254.5
    Routes: 3 MAC, 1 MAC/IP, 1 IMET, 1 EAD
Leaf-02#

```

The following example shows the output for the **show bgp l2vpn evpn route-type 4** command for route type 4 on VTEP 2:

```

Leaf-02# show bgp l2vpn evpn route-type 4
BGP routing table entry for [4][172.16.255.4:257][00010101010101010101][32][172.16.254.4]/23,
version 601
Paths: (1 available, best #1, table EVPN-BGP-Table)
  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
      Local vtep: 172.16.254.4
      Extended Community: ENCAP:8 EVPN ES-IMPORT:0x101:0x101:0x101
      rx pathid: 0, tx pathid: 0x0
      Updated on Jan 26 2021 19:41:40 UTC
BGP routing table entry for [4][172.16.255.5:257][00010101010101010101][32][172.16.254.5]/23,
version 658
Paths: (2 available, best #2, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 6
  Local
    172.16.254.5 (metric 3) (via default) from 172.16.255.2 (172.16.255.2)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: ENCAP:8 EVPN ES-IMPORT:0x101:0x101:0x101
      Originator: 172.16.255.5, Cluster list: 172.16.255.2
      rx pathid: 0, tx pathid: 0
      Updated on Jan 26 2021 19:43:19 UTC
  Refresh Epoch 6
  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
      Extended Community: ENCAP:8 EVPN ES-IMPORT:0x101:0x101:0x101
      Originator: 172.16.255.5, Cluster list: 172.16.255.1

```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

```
rx pathid: 0, tx pathid: 0x0
Updated on Jan 26 2021 19:43:19 UTC
```

```
Leaf-02#
```

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

```
Leaf-02# show bgp l2vpn evpn evi 101
BGP table version is 845, local router ID is 172.16.255.4
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: 172.16.254.4:101					
*mi [1][172.16.254.4:101][00010101010101010101][0]/23	172.16.254.5	0	100	0	?
>	::			32768	?
*>i [2][172.16.254.4:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24	172.16.254.3	0	100	0	?
>>i [2][172.16.254.4:101][0][48][44D3CA286C82][0][*]/20	172.16.254.5	0	100	0	?
>i [2][172.16.254.4:101][0][48][44D3CA286CC2][0][]/20	172.16.254.5	0	100	0	?
*>i [2][172.16.254.4:101][0][48][7C210DBD2741][32][10.1.101.1]/24	172.16.254.5	0	100	0	?
*> [2][172.16.254.4:101][0][48][7C210DBD9541][32][10.1.101.1]/24	::			32768	?
>i [2][172.16.254.4:101][0][48][F4CFE24334C1][0][]/20	172.16.254.3	0	100	0	?
*>i [2][172.16.254.4:101][0][48][F4CFE24334C1][32][10.1.101.11]/24	172.16.254.3	0	100	0	?
*>i [3][172.16.254.4:101][0][32][172.16.254.3]/17	172.16.254.3	0	100	0	?
*> [3][172.16.254.4:101][0][32][172.16.254.4]/17	::			32768	?
*>i [3][172.16.254.4:101][0][32][172.16.254.5]/17	172.16.254.5	0	100	0	?

```
Leaf-02#
```

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

EVI	Etag	Prod	Mac Address	Next Hop(s)	Seq Number
101	0	BGP	10b3.d56a.8fc1	V:10101 172.16.254.3	0
101	0	BGP	44d3.ca28.6c82	V:10101 172.16.254.5	0
101	0	BGP	44d3.ca28.6cc2	V:10101 172.16.254.5	0
101	0	BGP	7c21.0dbd.2741	V:10101 172.16.254.5	0
101	0	L2VPN	7c21.0dbd.9541	V1101:0	0
101	0	BGP	f4cf.e243.34c1	V:10101 172.16.254.3	0

```
Leaf-02#
```

The following example shows the output for the **show l2route evpn mac esi ethernet-segment-id** command on VTEP 2:

```
Leaf-02# show l2route evpn mac esi 0001.0101.0101.0101.0101
  EVI      ETag  Prod   Mac Address                Next Hop(s) Seq Number
-----
  101      0    BGP 44d3.ca28.6c82           V:10101 172.16.254.5      0
  101      0    BGP 44d3.ca28.6cc2           V:10101 172.16.254.5      0

Leaf-02#
```

The following example shows the output for the **show l2route evpn mac esi ethernet-segment-id detail** command on VTEP 2:

```
Leaf-02# show l2route evpn mac esi 0001.0101.0101.0101.0101 detail
EVPN Instance:          101
Ethernet Tag:           0
Producer Name:          BGP
MAC Address:             44d3.ca28.6c82
Num of MAC IP Route(s): 0
Sequence Number:        0
ESI:                    0001.0101.0101.0101.0101
Flags:                  B()
Next Hop(s):            V:10101 172.16.254.5
Resolved Next Hops:     V:10101 172.16.254.5
Resolved Redundancy Mode: Single-Active

EVPN Instance:          101
Ethernet Tag:           0
Producer Name:          BGP
MAC Address:             44d3.ca28.6cc2
Num of MAC IP Route(s): 0
Sequence Number:        0
ESI:                    0001.0101.0101.0101.0101
Flags:                  B()
Next Hop(s):            V:10101 172.16.254.5
Resolved Next Hops:     V:10101 172.16.254.5
Resolved Redundancy Mode: Single-Active

Leaf-02#
```

Return to [Verifying Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric](#), on page 499.

Outputs to Verify the Configuration on VTEP 3

The following example shows the output for the **show nve peer** command on VTEP 3:

```
Leaf-03# show nve peer
Interface VNI    Type Peer-IP           RMAC/Num_RTs  eVNI    state flags UP time
nve1     50901 L3CP 172.16.254.3       10b3.d56a.8fc8 50901    UP   A/M/4 04:23:46
nve1     50901 L3CP 172.16.254.4       7c21.0dbd.9548 50901    UP   A/M/4 03:24:57
nve1     10101 L2CP 172.16.254.3       5             10101    UP   N/A   04:23:46
nve1     10101 L2CP 172.16.254.4       4             10101    UP   N/A   03:24:57

Leaf-03#
```

The following example shows the output for the **show l2vpn evpn ethernet-segment detail** command on VTEP 3:

```
Leaf-03# show l2vpn evpn ethernet-segment detail
EVPN Ethernet Segment ID: 0001.0101.0101.0101.0101
  Interface:                Gi1/0/10
```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

```

Redundancy mode:      single-active
DF election wait time: 3 seconds
Split Horizon label:  0
State:               Ready
Encapsulation:       vxlan
Ordinal:             1
RD:                 172.16.254.5:9
  Export-RTs:       65001:101
Forwarder List:     172.16.254.4 172.16.254.5

```

Leaf-03#

The following example shows the output for the **show l2vpn evpn evi evpn-instance detail** command on VTEP 3:

```

Leaf-03# show l2vpn evpn evi 101 detail
EVPN instance:      101 (VLAN Based)
RD:                172.16.254.5:101 (auto)
Import-RTs:        65001:101
Export-RTs:        65001:101
Per-EVI Label:     none
State:             Established
Replication Type:  Ingress
Encapsulation:     vxlan
IP Local Learn:    Enabled (global)
Adv. Def. Gateway: Enabled (global)
Vlan:              101
  Ethernet-Tag:    0
  State:           Established
  Core If:         Vlan901
  Access If:       Vlan101
  NVE If:          nve1
  RMAC:            7c21.0dbd.2748
  Core Vlan:       901
  L2 VNI:          10101
  L3 VNI:          50901
  VTEP IP:         172.16.254.5
  VRF:             green
  IPv4 IRB:        Enabled
  IPv6 IRB:        Disabled
Pseudoports:
  GigabitEthernet1/0/10 service instance 101 (DF state: forwarding)
    Routes: 2 MAC, 0 MAC/IP
Peers:
  172.16.254.3
    Routes: 2 MAC, 2 MAC/IP, 1 IMET, 0 EAD
  172.16.254.4
    Routes: 1 MAC, 1 MAC/IP, 1 IMET, 1 EAD

```

Leaf-03#

The following example shows the output for the **show bgp l2vpn evpn route-type 4** command for route type 4 on VTEP 3:

```

Leaf-03# show bgp l2vpn evpn route-type 4
BGP routing table entry for [4][172.16.255.4:257][00010101010101010101][32][172.16.254.4]/23,
version 337
Paths: (2 available, best #2, table EVPN-BGP-Table)
Not advertised to any peer
Refresh Epoch 5
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
Extended Community: ENCAP:8 EVPN ES-IMPORT:0x101:0x101:0x101
Originator: 172.16.255.4, Cluster list: 172.16.255.2
rx pathid: 0, tx pathid: 0
Updated on Jan 26 2021 19:38:35 UTC
Refresh Epoch 5
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
  Extended Community: ENCAP:8 EVPN ES-IMPORT:0x101:0x101:0x101
  Originator: 172.16.255.4, Cluster list: 172.16.255.1
  rx pathid: 0, tx pathid: 0x0
  Updated on Jan 26 2021 19:38:35 UTC
BGP routing table entry for [4][172.16.255.5:257][00010101010101010101][32][172.16.254.5]/23,
version 1269
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
    2
  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
  Local vtep: 172.16.254.5
  Extended Community: ENCAP:8 EVPN ES-IMPORT:0x101:0x101:0x101
  rx pathid: 0, tx pathid: 0x0
  Updated on Jan 26 2021 19:40:14 UTC

Leaf-03#

```

The following example shows the output for the **show bgp l2vpn evpn evi evpn-instance** command on VTEP 3:

```

Leaf-03# show bgp l2vpn evpn evi 101
BGP table version is 1284, local router ID is 172.16.255.5
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: 172.16.254.5:101
*> [1][172.16.254.5:101][00010101010101010101][0]/23
   ::                                32768 ?
*mi 172.16.254.4          0      100    0 ?
*>i [2][172.16.254.5:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
   172.16.254.3          0      100    0 ?
*> [2][172.16.254.5:101][0][48][44D3CA286C82][0][*]/20
   ::                                32768 ?
*> [2][172.16.254.5:101][0][48][44D3CA286CC2][0][*]/20
   ::                                32768 ?
*> [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24
   ::                                32768 ?
*>i [2][172.16.254.5:101][0][48][7C210DBD9541][32][10.1.101.1]/24
   172.16.254.4          0      100    0 ?
*>i [2][172.16.254.5:101][0][48][F4CFE24334C1][0][*]/20
   172.16.254.3          0      100    0 ?
*>i [2][172.16.254.5:101][0][48][F4CFE24334C1][32][10.1.101.11]/24
   172.16.254.3          0      100    0 ?
*>i [3][172.16.254.5:101][0][32][172.16.254.3]/17
   172.16.254.3          0      100    0 ?
*>i [3][172.16.254.5:101][0][32][172.16.254.4]/17

```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

```

172.16.254.4          0    100    0 ?
*> [3][172.16.254.5:101][0][32][172.16.254.5]/17
::                               32768 ?
Leaf-03#

```

The following example shows the output for the **show l2route evpn mac** command on VTEP 3:

```

Leaf-03# show l2route evpn mac
EVI      ETag  Prod  Mac Address          Next Hop(s)  Seq Number
-----
101      0     BGP  10b3.d56a.8fc1      V:10101 172.16.254.3  0
101      0     L2VPN 44d3.ca28.6c82      Gi1/0/10:101 0
101      0     L2VPN 44d3.ca28.6cc2      Gi1/0/10:101 0
101      0     L2VPN 7c21.0dbd.2741      V1101:0      0
101      0     BGP  7c21.0dbd.9541      V:10101 172.16.254.4  0
101      0     BGP  f4cf.e243.34c1      V:10101 172.16.254.3  0
Leaf-03#

```

The following example shows the output for the **show l2route evpn mac esi ethernet-segment-id** command on VTEP 3:

```

Leaf-03# show l2route evpn mac esi 0001.0101.0101.0101.0101
EVI      ETag  Prod  Mac Address          Next Hop(s)  Seq Number
-----
101      0     L2VPN 44d3.ca28.6c82      Gi1/0/10:101 0
101      0     L2VPN 44d3.ca28.6cc2      Gi1/0/10:101 0
Leaf-03#

```

The following example shows the output for the **show l2route evpn mac esi ethernet-segment-id detail** command on VTEP 3:

```

Leaf-03# show l2route evpn mac esi 0001.0101.0101.0101.0101 detail
EVPN Instance:          101
Ethernet Tag:           0
Producer Name:          L2VPN
MAC Address:            44d3.ca28.6c82
Num of MAC IP Route(s): 0
Sequence Number:        0
ESI:                    0001.0101.0101.0101.0101
Flags:                   B()
Next Hop(s):            Gi1/0/10:101

EVPN Instance:          101
Ethernet Tag:           0
Producer Name:          L2VPN
MAC Address:            44d3.ca28.6cc2
Num of MAC IP Route(s): 0
Sequence Number:        0
ESI:                    0001.0101.0101.0101.0101
Flags:                   B()
Next Hop(s):            Gi1/0/10:101
Leaf-03#

```

Return to [Verifying Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric](#), on page 499.

Outputs to Verify the Configuration on Spine Switch 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 5443, main routing table version 5443
17 network entries using 5848 bytes of memory
34 path entries using 7072 bytes of memory
13/11 BGP path/bestpath attribute entries using 3744 bytes of memory
3 BGP rinfo entries using 120 bytes of memory
10 BGP extended community entries using 480 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 17264 total bytes of memory
BGP activity 101/84 prefixes, 2825/2791 paths, scan interval 60 secs
25 networks peaked at 14:54:41 Jan 26 2021 UTC (05:39:56.356 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.2  4      65001   5664   5668    5443   0    0 05:40:29    15
172.16.255.3  4      65001    378   5690    5443   0    0 05:35:23     5
172.16.255.4  4      65001    440   1633    5443   0    0 03:36:33     6
172.16.255.5  4      65001    594   5296    5443   0    0 04:34:27     8

Spine-01#
```

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

```
Spine-01# show bgp l2vpn evpn
BGP table version is 5443, local router ID is 172.16.255.1
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: 172.16.254.4:7
 *>i [1][172.16.254.4:7][00010101010101010101][4294967295]/23
      172.16.254.4      0      100      0 ?
Route Distinguisher: 172.16.254.4:101
 *>i [1][172.16.254.4:101][00010101010101010101][0]/23
      172.16.254.4      0      100      0 ?
Route Distinguisher: 172.16.254.5:9
 *>i [1][172.16.254.5:9][00010101010101010101][4294967295]/23
      172.16.254.5      0      100      0 ?
 * i      172.16.254.5      0      100      0 ?
Route Distinguisher: 172.16.254.5:101
 *>i [1][172.16.254.5:101][00010101010101010101][0]/23
      172.16.254.5      0      100      0 ?
 * i      172.16.254.5      0      100      0 ?
Route Distinguisher: 172.16.254.3:101
 * i [2][172.16.254.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
      172.16.254.3      0      100      0 ?
 *>i      172.16.254.3      0      100      0 ?
 * i [2][172.16.254.3:101][0][48][F4CFE24334C1][0][*]/20
      172.16.254.3      0      100      0 ?
 *>i      172.16.254.3      0      100      0 ?
 * i [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24
```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

```

172.16.254.3          0 100 0 ?
*>i                 172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.254.4:101
* i [2][172.16.254.4:101][0][48][7C210DBD9541][32][10.1.101.1]/24
172.16.254.4          0 100 0 ?
*>i                 172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.254.5:101
* i [2][172.16.254.5:101][0][48][44D3CA286C82][0][*]/20
172.16.254.5          0 100 0 ?
*>i                 172.16.254.5          0 100 0 ?
* i [2][172.16.254.5:101][0][48][44D3CA286CC2][0][*]/20
172.16.254.5          0 100 0 ?
*>i                 172.16.254.5          0 100 0 ?
* i [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24
172.16.254.5          0 100 0 ?
*>i                 172.16.254.5          0 100 0 ?
Route Distinguisher: 172.16.254.3:101
* i [3][172.16.254.3:101][0][32][172.16.254.3]/17
172.16.254.3          0 100 0 ?
*>i                 172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.254.4:101
* i [3][172.16.254.4:101][0][32][172.16.254.4]/17
172.16.254.4          0 100 0 ?
*>i                 172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.254.5:101
* i [3][172.16.254.5:101][0][32][172.16.254.5]/17
172.16.254.5          0 100 0 ?
*>i                 172.16.254.5          0 100 0 ?
Route Distinguisher: 172.16.255.4:257
* i [4][172.16.255.4:257][00010101010101010101][32][172.16.254.4]/23
172.16.254.4          0 100 0 ?
*>i                 172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.255.5:257
* i [4][172.16.255.5:257][00010101010101010101][32][172.16.254.5]/23
172.16.254.5          0 100 0 ?
*>i                 172.16.254.5          0 100 0 ?
Route Distinguisher: 1:1
* i [5][1:1][0][24][10.1.101.0]/17
172.16.254.5          0 100 0 ?
* i                 172.16.254.4          0 100 0 ?
*>i                 172.16.254.3          0 100 0 ?
* i                 172.16.254.3          0 100 0 ?

```

Spine-01#

The following example shows the output for the **show ip route** command on Spine Switch 1:

```

Spine-01# show ip route
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

172.16.0.0/16 is variably subnetted, 17 subnets, 2 masks

```

C      172.16.13.0/24 is directly connected, GigabitEthernet1/0/1
L      172.16.13.1/32 is directly connected, GigabitEthernet1/0/1
C      172.16.14.0/24 is directly connected, GigabitEthernet1/0/2
L      172.16.14.1/32 is directly connected, GigabitEthernet1/0/2
C      172.16.15.0/24 is directly connected, GigabitEthernet1/0/3
L      172.16.15.1/32 is directly connected, GigabitEthernet1/0/3
O      172.16.23.0/24
        [110/2] via 172.16.13.3, 05:35:46, GigabitEthernet1/0/1
O      172.16.24.0/24
        [110/2] via 172.16.14.4, 03:37:00, GigabitEthernet1/0/2
O      172.16.25.0/24
        [110/2] via 172.16.15.5, 03:38:33, GigabitEthernet1/0/3
O      172.16.254.3/32
        [110/2] via 172.16.13.3, 05:35:46, GigabitEthernet1/0/1
O      172.16.254.4/32
        [110/2] via 172.16.14.4, 03:36:50, GigabitEthernet1/0/2
O      172.16.254.5/32
        [110/2] via 172.16.15.5, 03:38:33, GigabitEthernet1/0/3
C      172.16.255.1/32 is directly connected, Loopback0
O      172.16.255.2/32
        [110/3] via 172.16.15.5, 03:38:33, GigabitEthernet1/0/3
        [110/3] via 172.16.14.4, 03:37:00, GigabitEthernet1/0/2
        [110/3] via 172.16.13.3, 05:35:46, GigabitEthernet1/0/1
O      172.16.255.3/32
        [110/2] via 172.16.13.3, 05:35:46, GigabitEthernet1/0/1
O      172.16.255.4/32
        [110/2] via 172.16.14.4, 03:36:56, GigabitEthernet1/0/2
O      172.16.255.5/32
        [110/2] via 172.16.15.5, 03:38:33, GigabitEthernet1/0/3
Spine-01#

```

Return to [Verifying Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric](#), on page 499.

Outputs to Verify the Configuration on Spine Switch 2

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 5499, main routing table version 5499
17 network entries using 5848 bytes of memory
34 path entries using 7072 bytes of memory
13/11 BGP path/bestpath attribute entries using 3744 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
10 BGP extended community entries using 480 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 17264 total bytes of memory
BGP activity 101/84 prefixes, 2823/2789 paths, scan interval 60 secs
25 networks peaked at 14:56:03 Jan 26 2021 UTC (05:40:54.652 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   5669   5665    5499   0    0 05:41:28    15
172.16.255.3  4      65001    381   5691    5499   0    0 05:36:22     5
172.16.255.4  4      65001    440  1632    5499   0    0 03:37:31     6
172.16.255.5  4      65001    594   5291    5499   0    0 04:35:26     8

Spine-02#

```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

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

```
Spine-02# show bgp l2vpn evpn
BGP table version is 5499, local router ID is 172.16.255.2
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: 172.16.254.4:7
*>i [1][172.16.254.4:7][00010101010101010101][4294967295]/23
      172.16.254.4          0      100      0 ?
* i      172.16.254.4          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
*>i [1][172.16.254.4:101][00010101010101010101][0]/23
      172.16.254.4          0      100      0 ?
* i      172.16.254.4          0      100      0 ?
Route Distinguisher: 172.16.254.5:9
*>i [1][172.16.254.5:9][00010101010101010101][4294967295]/23
      172.16.254.5          0      100      0 ?
Route Distinguisher: 172.16.254.5:101
*>i [1][172.16.254.5:101][00010101010101010101][0]/23
      172.16.254.5          0      100      0 ?
Route Distinguisher: 172.16.254.3:101
* i [2][172.16.254.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
      172.16.254.3          0      100      0 ?
*>i      172.16.254.3          0      100      0 ?
* i [2][172.16.254.3:101][0][48][F4CFE24334C1][0][*]/20
      172.16.254.3          0      100      0 ?
*>i      172.16.254.3          0      100      0 ?
* i [2][172.16.254.3:101][0][48][F4CFE24334C1][32][10.1.101.11]/24
      172.16.254.3          0      100      0 ?
*>i      172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
* i [2][172.16.254.4:101][0][48][7C210DBD9541][32][10.1.101.1]/24
      172.16.254.4          0      100      0 ?
*>i      172.16.254.4          0      100      0 ?
Route Distinguisher: 172.16.254.5:101
* i [2][172.16.254.5:101][0][48][44D3CA286C82][0][*]/20
      172.16.254.5          0      100      0 ?
*>i      172.16.254.5          0      100      0 ?
* i [2][172.16.254.5:101][0][48][44D3CA286CC2][0][*]/20
      172.16.254.5          0      100      0 ?
*>i      172.16.254.5          0      100      0 ?
* i [2][172.16.254.5:101][0][48][7C210DBD2741][32][10.1.101.1]/24
      172.16.254.5          0      100      0 ?
*>i      172.16.254.5          0      100      0 ?
Route Distinguisher: 172.16.254.3:101
* i [3][172.16.254.3:101][0][32][172.16.254.3]/17
      172.16.254.3          0      100      0 ?
*>i      172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.254.4:101
* i [3][172.16.254.4:101][0][32][172.16.254.4]/17
      172.16.254.4          0      100      0 ?
*>i      172.16.254.4          0      100      0 ?
Route Distinguisher: 172.16.254.5:101
* i [3][172.16.254.5:101][0][32][172.16.254.5]/17
      172.16.254.5          0      100      0 ?
*>i      172.16.254.5          0      100      0 ?
```

```

Route Distinguisher: 172.16.255.4:257
 * i  [4][172.16.255.4:257][00010101010101010101][32][172.16.254.4]/23
      172.16.254.4          0      100      0 ?
 *>i      172.16.254.4          0      100      0 ?
Route Distinguisher: 172.16.255.5:257
 * i  [4][172.16.255.5:257][00010101010101010101][32][172.16.254.5]/23
      172.16.254.5          0      100      0 ?
 *>i      172.16.254.5          0      100      0 ?
Route Distinguisher: 1:1
 * i  [5][1:1][0][24][10.1.101.0]/17
      172.16.254.5          0      100      0 ?
 * i      172.16.254.4          0      100      0 ?
 *>i      172.16.254.3          0      100      0 ?
 * i      172.16.254.3          0      100      0 ?

```

Spine-02#

The following example shows the output for the **show ip route** command on Spine Switch 2:

```

Spine-02# show ip route
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

```

      172.16.0.0/16 is variably subnetted, 17 subnets, 2 masks
O      172.16.13.0/24
      [110/2] via 172.16.23.3, 05:36:24, GigabitEthernet1/0/1
O      172.16.14.0/24
      [110/2] via 172.16.24.4, 03:37:38, GigabitEthernet1/0/2
O      172.16.15.0/24
      [110/2] via 172.16.25.5, 03:39:11, GigabitEthernet1/0/3
C      172.16.23.0/24 is directly connected, GigabitEthernet1/0/1
L      172.16.23.2/32 is directly connected, GigabitEthernet1/0/1
C      172.16.24.0/24 is directly connected, GigabitEthernet1/0/2
L      172.16.24.2/32 is directly connected, GigabitEthernet1/0/2
C      172.16.25.0/24 is directly connected, GigabitEthernet1/0/3
L      172.16.25.2/32 is directly connected, GigabitEthernet1/0/3
O      172.16.254.3/32
      [110/2] via 172.16.23.3, 05:36:24, GigabitEthernet1/0/1
O      172.16.254.4/32
      [110/2] via 172.16.24.4, 03:37:28, GigabitEthernet1/0/2
O      172.16.254.5/32
      [110/2] via 172.16.25.5, 03:39:11, GigabitEthernet1/0/3
O      172.16.255.1/32
      [110/3] via 172.16.25.5, 03:39:11, GigabitEthernet1/0/3
      [110/3] via 172.16.24.4, 03:37:38, GigabitEthernet1/0/2
      [110/3] via 172.16.23.3, 05:36:24, GigabitEthernet1/0/1
C      172.16.255.2/32 is directly connected, Loopback0
O      172.16.255.3/32
      [110/2] via 172.16.23.3, 05:36:24, GigabitEthernet1/0/1
O      172.16.255.4/32
      [110/2] via 172.16.24.4, 03:37:34, GigabitEthernet1/0/2
O      172.16.255.5/32

```

Example: Configuring Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric

```
[110/2] via 172.16.25.5, 03:39:11, GigabitEthernet1/0/3
```

```
Spine-02#
```

Return to [Verifying Dual-Homing with Single-Active Redundancy in a BGP EVPN VXLAN Fabric](#), on page 499.



CHAPTER 11

Configuring Private VLANs in a BGP EVPN VXLAN Fabric

- [Restrictions for Private VLANs in a BGP EVPN VXLAN Fabric, on page 515](#)
- [Information About Private VLANs in a BGP EVPN VXLAN Fabric, on page 515](#)
- [How to Configure Private VLANs in a BGP EVPN VXLAN Fabric, on page 521](#)
- [Configuration Examples for Private VLANs in a BGP EVPN VXLAN Fabric, on page 525](#)

Restrictions for Private VLANs in a BGP EVPN VXLAN Fabric

Configuration of Private VLANs in a BGP EVPN VXLAN fabric must be done in the following order:

1. Configure VLAN with primary and secondary associations.
2. Enable EVPN separately in each of the primary, community, and isolated VLANs.

For more information, see [Configuring an EVPN Instance on the VLAN on a VTEP, on page 18](#).

If there's an EVPN configuration already associated with a VLAN, you can't directly configure the PVLAN associations for this VLAN. First disassociate the EVPN configuration from the VLAN. Next, configure the PVLAN association. Then reconfigure EVPN in each of the newly configured primary, community, and isolated VLANs.

Information About Private VLANs in a BGP EVPN VXLAN Fabric

A private VLAN (PVLAN) divides a regular VLAN into logical partitions, allowing limited broadcast boundaries among selected port-groups on a single Layer 2 Ethernet switch. The single Ethernet switch's PVLAN capabilities can be extended over the BGP EVPN VXLAN enabled network to build partitioned bridge-domain between port-groups across multiple Ethernet switches in the BGP EVPN VXLAN VTEP mode. The integration of PVLAN with a BGP EVPN VXLAN network enables the following benefits:

- Microsegmented Layer 2 network segregation across one or more BGP EVPN VXLAN switches.
- Partitioned and secured user-group Layer 2 network that limits the communication with dynamic or static port configuration assignments.
- IP subnet pool conservation across BGP EVPN VXLAN network while extending segregated Layer 2 network across the fabric.

- Conservation of Layer 2 overlay tunnels and peer networks with a single virtual network identifier (VNI) mapped to Primary VLAN.

Primary and Secondary VLANs

Each subdomain in a PVLAN is represented by a pair of VLANs: a primary VLAN and a secondary VLAN. A PVLAN can have multiple VLAN pairs, one pair for each subdomain. All VLAN pairs in a PVLAN share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another. A secondary VLAN can either be an isolated VLAN or a community VLAN. Primary and secondary VLANs have the following characteristics:

- **Primary VLAN:** A PVLAN has only one primary VLAN. Every port in a PVLAN is a member of the primary VLAN. The primary VLAN carries unidirectional traffic downstream from the promiscuous ports to the host (isolated and community) ports and to other promiscuous ports.
- **Isolated VLAN:** A PVLAN has only one isolated VLAN. An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts towards the promiscuous ports and the gateway.
- **Community VLAN:** A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port gateways and to other host ports in the same community. You can configure multiple community VLANs in a PVLAN.

Private VLAN Ports

PVLAN ports are access ports that are one of these types:

- **Promiscuous:** A promiscuous port belongs to the primary VLAN. It can communicate with all interfaces, including the community and isolated host ports that belong to the secondary VLANs associated with the primary VLAN.
- **Isolated:** An isolated port is a host port that belongs to an isolated secondary VLAN. It has complete Layer 2 separation from other ports within the same PVLAN, except for the promiscuous ports. PVLANS block all traffic to isolated ports except traffic from promiscuous ports. Likewise, PVLANS forward the traffic from an isolated port only to promiscuous ports.
- **Community:** A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with promiscuous ports. Community ports are isolated at Layer 2 from all other interfaces in external communities and also from isolated ports within their private VLAN.

For more information about PVLANS and the steps to configure PVLANS, see "Configuring Private VLANs" module in the *VLAN Configuration Guide* for the applicable release.

Extension of Private VLANs in a BGP EVPN VXLAN Fabric

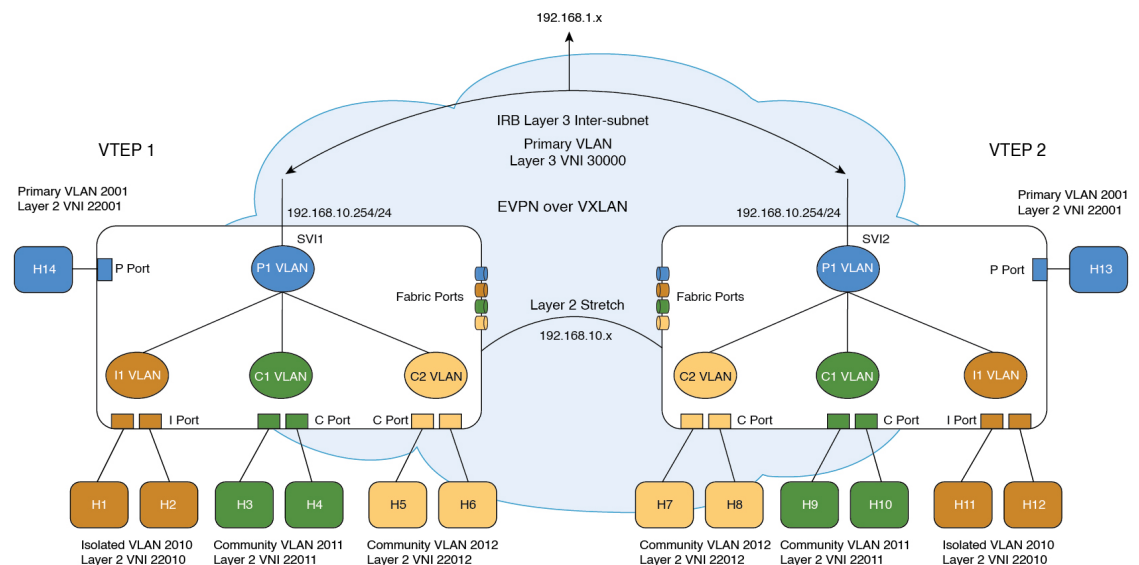
Private VLANs (PVLANS) partition a regular VLAN domain into subdomains and provide Layer 2 isolation between ports within the same PVLAN. Like a regular VLAN, a private VLAN can span multiple Layer 2 switches. In a private VLAN that spans across multiple devices, traffic from an isolated port on Switch A does not reach an isolated port on Switch B. This is achieved by the trunk port carrying the primary VLAN and secondary VLANs to neighboring switches with dot1q tag in a traditional Layer 2 network. With BGP EVPN VXLAN enabled in the PVLANS on the VTEPs, the L2VNI segment preserves the PVLAN semantics and

provides the Layer 2 isolation for the stretched PVLAN segment across the VTEPs in the overlay fabric. PVLAN extension with BGP EVPN VXLAN allows you to:

- Seamlessly migrate and join (or stretch) the PVLAN domain like any regular VLAN.
- Access to centralized common services such as printer or DHCP through the promiscuous port on any VTEP in the EVPN overlay.
- Maintain community and isolated VLAN semantics in the overlay fabric across all the VTEPs. The EVPN fabric provides a logical single switch view for the respective Layer 2 domain.

The following image shows PVLAN extension in a BGP EVPN VXLAN fabric with two VTEPs:

Figure 40: PVLAN Extension in a BGP EVPN VXLAN Fabric



357506

Traffic Forwarding for Private VLANs in a BGP EVPN VXLAN Fabric

You can forward known unicast and broadcast, unknown unicast, and multicast (BUM) traffic between PVLANS in a BGP EVPN VXLAN fabric. On the Source VTEP, the forwarding process on the access PVLAN ports (promiscuous, isolated, community) adheres to the baseline PVLAN forwarding. With BGP EVPN VXLAN enabled in the PVLAN domain, the remote host routes are learned and programmed in the hardware of the respective PVLANS. The following sections illustrate the forwarding scenarios for unicast and BUM traffic between local and remote hosts for each of the secondary VLANs.

Known Unicast Traffic Forwarding

The sending VTEP bridges a known unicast packet with the corresponding secondary VLAN's virtual network identifier (VNI) ID. The packet arrives on the receiving VTEP. After decapsulation, receiving VTEP processes the packet in the same way as a packet from a local PVLAN host port. The packet gets mapped to the respective community, isolated, or primary VLAN.

The following images illustrate the known unicast traffic forwarding scenarios for PVLANS in a BGP EVPN VXLAN fabric:

Figure 41: Unicast Traffic from Promiscuous Port : H14 to H8

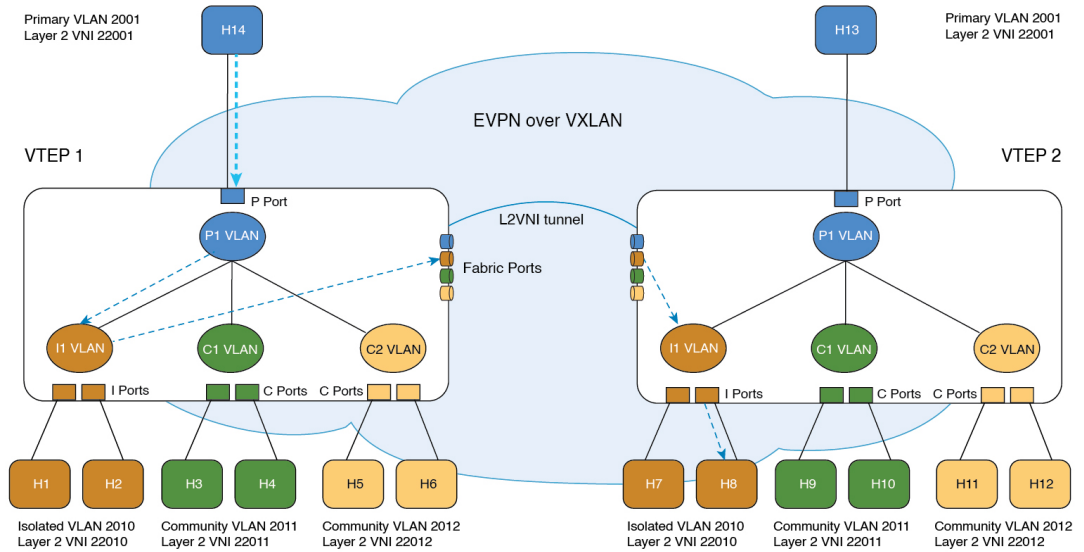
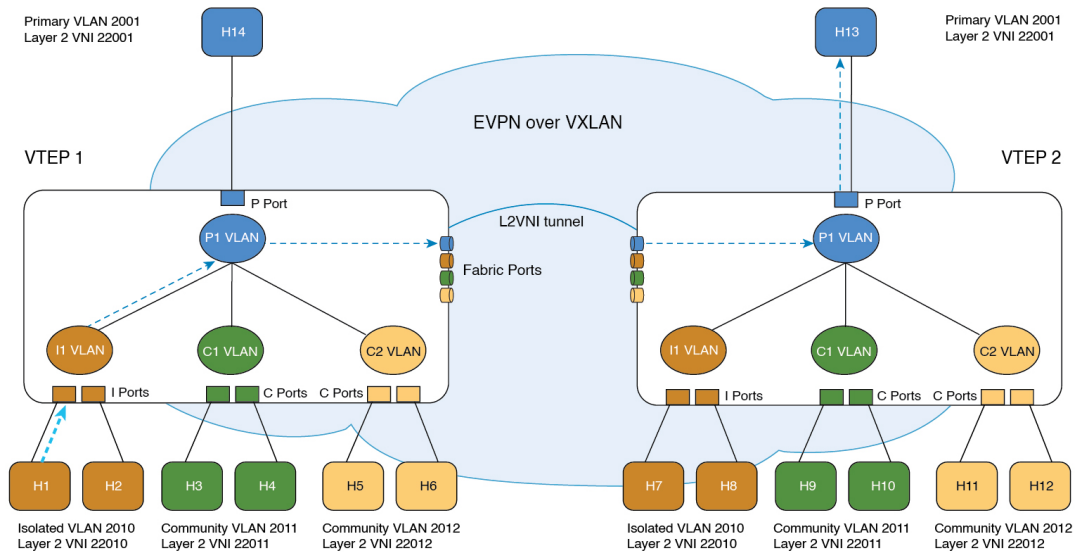


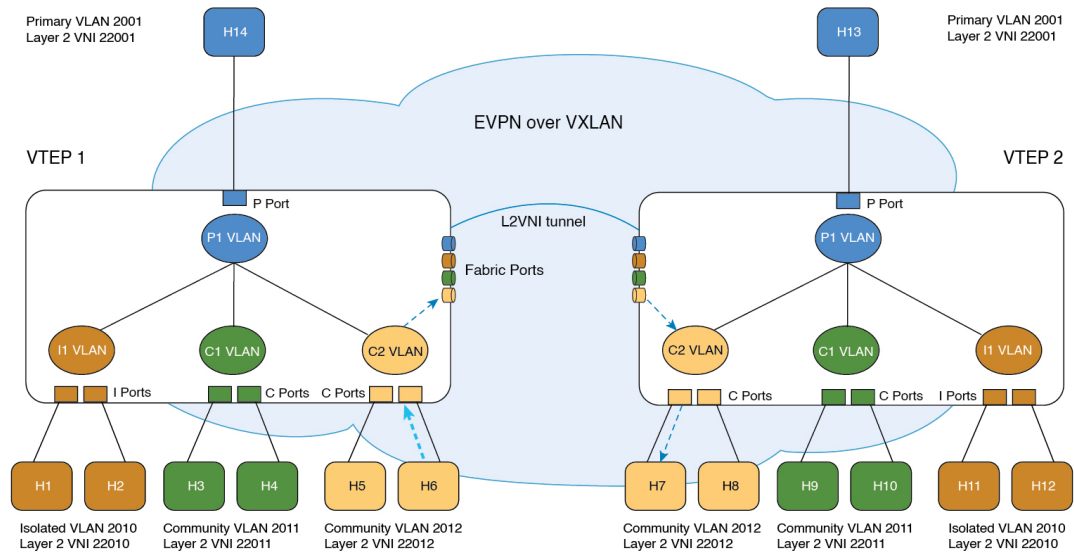
Figure 42: Unicast Traffic from Isolated Port: H1 to H13



357621

357622

Figure 43: Unicast Traffic from Community Port: H6 to H7



957623

Broadcast, Unknown Unicast, and Multicast Traffic Forwarding

In a regular VLAN, broadcasts are forwarded to all ports in that VLAN. Private VLAN broadcast forwarding depends on the port sending the broadcast:

- An isolated port sends a broadcast only to the promiscuous ports or trunk ports.
- A community port sends a broadcast to all promiscuous ports, trunk ports, and ports in the same community VLAN.
- A promiscuous port sends a broadcast to all ports in the private VLAN (other promiscuous ports, trunk ports, isolated ports, and community ports).

In addition to the above, a copy of the flood packet is sent to the remote VTEPs with the respective L2VNI. (See [Configuring EVPN VXLAN Layer 2 Overlay Network](#)). On the remote VTEP, the flood copy is again replicated towards the access as per the PVLAN broadcast rules mentioned above. Flood packets received from the fabric are not sent back to fabric with split-horizon check.

During forwarding, if a packet's MAC address isn't available in the lookup, the VTEP replicates the packet with the VNI ID of the forwarding (or incoming) VLAN. The VTEP forwards the BUM packets with the VNI ID of the corresponding VLAN. The receiving VTEP decapsulates the BUM packet and maps the VNI ID to the corresponding secondary VLAN. This mapping ensures that the flood rules remain local. The VTEP then processes the packet in the same way as a packet from a local host port.

For isolated VLANs, after the destination MAC address lookup results in an unknown unicast from the source port, it's not locally known whether the destination MAC address belongs to the remote isolated VLAN host or the remote primary VLAN host. Hence, the BUM packet copy is allowed to go the egress VTEPs with the isolated VLAN VNI ID. On egress VTEPs, this BUM copy gets flooded on local isolated ports and local promiscuous ports. As a result, BUM traffic from remote isolated ports to local isolated ports is unavoidable.



Note Forwarding of unknown unicast traffic from an isolated port to a remote promiscuous port isn't supported.

The following images illustrate the BUM traffic forwarding scenarios for PVLANS in a BGP EVPN VXLAN fabric:

Figure 44: BUM Traffic from Promiscuous Port (H14)

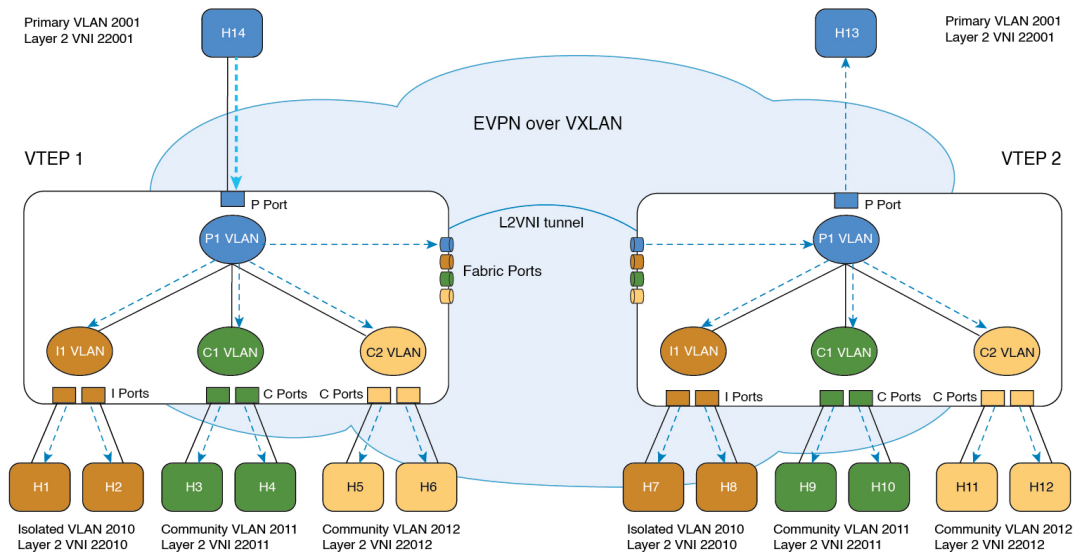


Figure 45: BUM Traffic from Isolated Port (H1)

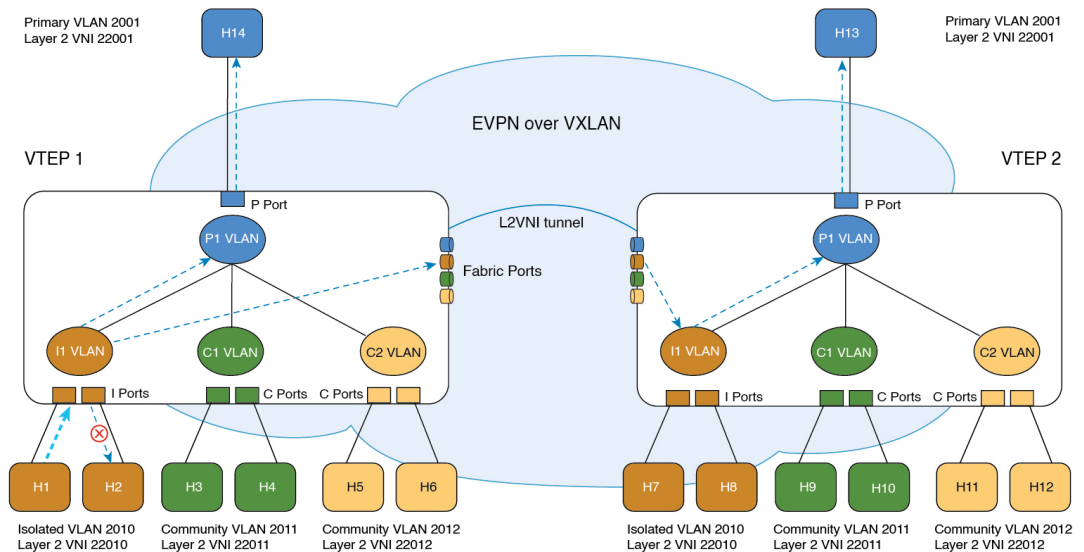
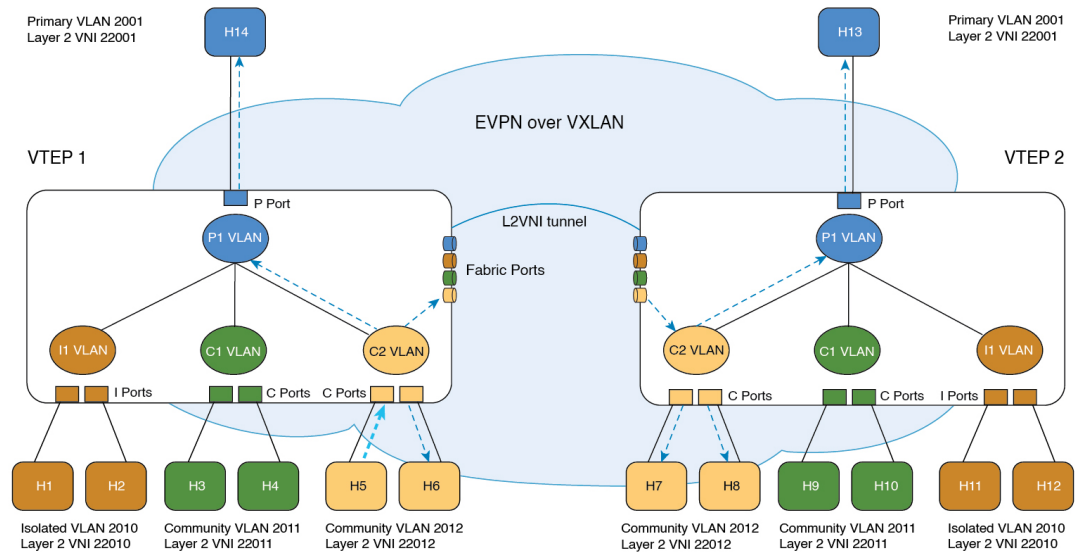


Figure 46: BUM Traffic from Community Port (H5)



957620

Routed Traffic Forwarding

Routed traffic between the hosts in a microsegmented VLAN is through the associated Primary VLAN SVI on the Local VTEP (For more information, see "Configuring Private VLANs" module in the *VLAN Configuration Guide* for the applicable release). When the source and destination hosts are across the EVPN VXLAN fabric, the routed traffic between the microsegmented VLAN hosts follows the Symmetric Integrated Routing and Bridging (IRB) method to cross the fabric (For more information, see [Configuring EVPN VXLAN Integrated Routing and Bridging, on page 93](#)). On the destination VTEP, traffic is routed from the core VLAN SVI to the associated Primary VLAN SVI interface and then bridged in the microsegmented local destination Secondary VLAN.

How to Configure Private VLANs in a BGP EVPN VXLAN Fabric

When you configure PVLANS in a BGP EVPN VXLAN fabric, the existing PVLAN configuration is preserved and the Layer 2 VNI configuration is added to the PVLAN. By adding the Layer 2 VNI configuration, you expand the PVLAN and stretch it over the fabric across the VTEPs in the fabric.

In a BGP EVPN VXLAN fabric, the EVPN control plane distributes the MAC and MAC-IP routes. In addition, PVLANS handle BUM and unicast traffic forwarding differently compared to regular VLANs. Due to these two reasons, you can create and delete PVLANS strictly in the following ways:

- To create a PVLAN, first configure the VLAN with primary and secondary associations. Next, enable EVPN separately in each of the primary, community, and isolated VLANs.



Note If there's an EVPN configuration already associated with a VLAN, you can't directly configure the PVLAN associations for this VLAN. First, use the **member vni** command in VLAN configuration mode to disassociate the EVPN configuration from the VLAN. Next, configure the PVLAN association. Now reconfigure EVPN in each of the newly configured primary, community, and isolated VLANs.

- To delete a PVLAN, ensure that you unconfigure EVPN in the respective VLAN before you modify the PVLAN configuration.

Configuring the Primary and Secondary VLANs for a Private VLAN

To configure the primary and secondary VLANs for a private VLAN, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enters privileged EXEC mode. Enter password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vlan <i>vlan-id</i> Example: Device(config)# vlan 101	Enters VLAN configuration mode for the specified VLAN ID.
Step 4	private-vlan {association [add remove] secondary-vlan-list community isolated primary} Example: Device(config-vlan)# private-vlan primary Device(config-vlan)# private-vlan association 102	Configures the VLAN as a PVLAN and configures the association between primary and secondary VLANs. Use the primary keyword to configure the VLAN as a PVLAN. Use the community keyword to designate the VLAN as a community VLAN. Use the isolated keyword to designate the VLAN as an isolated VLAN. Use the association [add remove] keyword to add or remove the association between a primary and secondary VLAN.

	Command or Action	Purpose
Step 5	exit Example: Device(config-vlan)# exit	Exits VLAN configuration mode and returns to global configuration mode.
Step 6	Repeat steps 3 to 5 for each primary and secondary VLAN, as needed.	--
Step 7	end Example: Device(config)# end	Exits global configuration mode and enters privileged EXEC mode.

Configuring the Port of a Private VLAN

To configure a port of a PVLAN, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enters privileged EXEC mode. Enter password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface GigabitEthernet1/0/1	Enters interface configuration mode for the specified interface ID.
Step 4	switchport mode private-vlan {host promiscuous} Example: Device(config-if)# switchport mode private-vlan host	Configures the interface as either a host PVLAN port or a promiscuous PVLAN port.
Step 5	switchport private-vlan {host-association mapping primary-vlan-id secondary-vlan-id-list}	Associates a PVLAN host port or maps a PVLAN promiscuous port to a primary VLAN.

	Command or Action	Purpose
	Example: <pre>Device(config-if) # switchport private-vlan host-association 101 104</pre>	Note If you configure a port as a PVLAN host port and you do not configure a valid PVLAN association with the switchport private-vlan host-association command, the interface becomes inactive. Note If you configure a port as a PVLAN promiscuous port and you do not configure a valid PVLAN mapping with the switchport private-vlan mapping command, the interface becomes inactive.
Step 6	end Example: <pre>Device(config-if) # end</pre>	Exits interface configuration mode and enters privileged EXEC mode.

Enabling EVPN in a Private VLAN

To enable EVPN in a PVLAN, perform the following steps:



Note Enable EVPN separately in each of the primary, community, and isolated VLANs.

Procedure

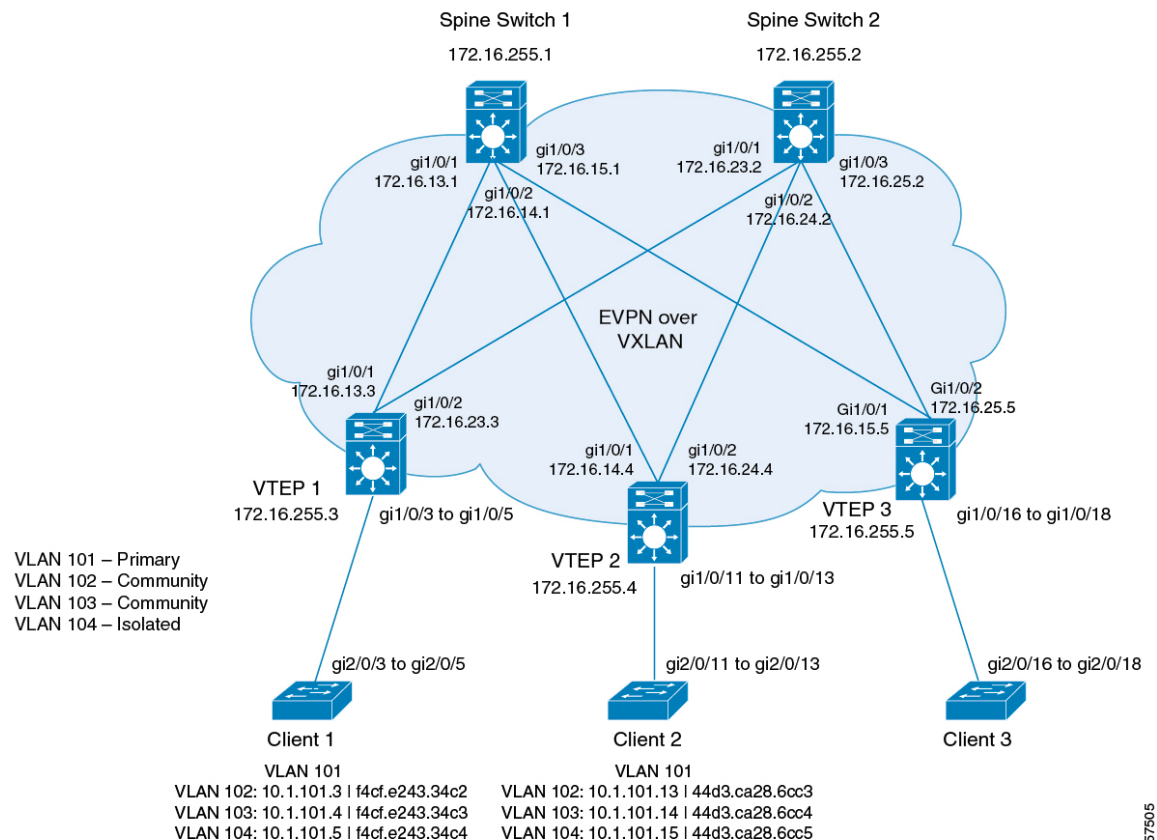
	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enters privileged EXEC mode. Enter password, if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	vlan configuration <i>vlan-id</i> Example: <pre>Device(config)# vlan configuration 101</pre>	Enters VLAN configuration mode for the specified PVLAN interface.
Step 4	member evpn-instance <i>evpn-instance-id</i> vni <i>layer2-vni-id</i>	Adds EVPN instance as a member of the PVLAN configuration.

	Command or Action	Purpose
	Example: Device (config-vlan) # member evpn-instance 1 vni 6000	The VNI here is used as a Layer 2 VNI.
Step 5	end Example: Device (config-vlan) # end	Exits VLAN configuration mode and enters privileged EXEC mode.

Configuration Examples for Private VLANs in a BGP EVPN VXLAN Fabric

This section provides a configuration example for PVLANS in a BGP EVPN VXLAN fabric using the following topology:

Figure 47: Private VLANs in a BGP EVPN VXLAN Fabric



The topology shows an EVPN VXLAN network with two spine switches (Spine Switch 1 and Spine Switch 2) and three VTEPs (VTEP 1, VTEP 2, and VTEP 3). The network has an extended PVLAN with VLAN 101

as the primary VLAN. VLAN 102, VLAN 103, and VLAN 104 are the secondary VLANs. The following tables provide the sample configurations for the devices in this topology:

Table 49: Configuring VTEP 1, VTEP 2, and VTEP 3 for PVLAN Extension in a BGP EVPN VXLAN Fabric

VTEP 1	VTEP 2	VTEP 3
<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 ! vtp mode transparent ! l2vpn evpn replication-type static default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan ! l2vpn evpn instance 103 vlan-based encapsulation vxlan ! l2vpn evpn instance 104 vlan-based encapsulation vxlan ! l2vpn evpn instance 201 vlan-based encapsulation vxlan ! l2vpn evpn instance 202 vlan-based encapsulation vxlan</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 ! vtp mode transparent ! l2vpn evpn replication-type static default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan ! l2vpn evpn instance 103 vlan-based encapsulation vxlan ! l2vpn evpn instance 104 vlan-based encapsulation vxlan ! l2vpn evpn instance 201 vlan-based encapsulation vxlan ! l2vpn evpn instance 202 vlan-based encapsulation vxlan</pre>	<pre>Leaf-03# show running-config hostname Leaf-03 ! 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 ! vtp mode transparent ! l2vpn evpn replication-type static default-gateway advertise ! l2vpn evpn instance 101 vlan-based encapsulation vxlan ! l2vpn evpn instance 102 vlan-based encapsulation vxlan ! l2vpn evpn instance 103 vlan-based encapsulation vxlan ! l2vpn evpn instance 104 vlan-based encapsulation vxlan ! l2vpn evpn instance 201 vlan-based encapsulation vxlan ! l2vpn evpn instance 202 vlan-based encapsulation vxlan</pre>

VTEP 1	VTEP 2	VTEP 3
<pre> ! l2vpn evpn instance 203 vlan-based encapsulation vxlan ! l2vpn evpn instance 204 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 103 member evpn-instance 103 vni 10103 vlan configuration 104 member evpn-instance 104 vni 10104 vlan configuration 201 member evpn-instance 201 vni 10201 vlan configuration 202 member evpn-instance 202 vni 10202 vlan configuration 203 member evpn-instance 203 vni 10203 vlan configuration 204 member evpn-instance 204 vni 10204 vlan configuration 901 member vni 50901 ! vlan 101 private-vlan primary private-vlan association 102-104 ! vlan 102 private-vlan community ! vlan 103 private-vlan community ! vlan 104 private-vlan isolated ! vlan 201 private-vlan primary private-vlan association 202-204 ! vlan 202 private-vlan community ! </pre>	<pre> ! l2vpn evpn instance 203 vlan-based encapsulation vxlan ! l2vpn evpn instance 204 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 103 member evpn-instance 103 vni 10103 vlan configuration 104 member evpn-instance 104 vni 10104 vlan configuration 201 member evpn-instance 201 vni 10201 vlan configuration 202 member evpn-instance 202 vni 10202 vlan configuration 203 member evpn-instance 203 vni 10203 vlan configuration 204 member evpn-instance 204 vni 10204 vlan configuration 901 member vni 50901 ! vlan 101 private-vlan primary private-vlan association 102-104 ! vlan 102 private-vlan community ! vlan 103 private-vlan community ! vlan 104 private-vlan isolated ! vlan 201 private-vlan primary private-vlan association 202-204 ! vlan 202 private-vlan community ! </pre>	<pre> ! l2vpn evpn instance 203 vlan-based encapsulation vxlan ! l2vpn evpn instance 204 vlan-based encapsulation vxlan ! system mtu 9198 ! vlan configuration 101 member evpn-instance 101 vni 10101 vlan configuration 102 member evpn-instance 102 vni 10102 vlan configuration 103 member evpn-instance 103 vni 10103 vlan configuration 104 member evpn-instance 104 vni 10104 vlan configuration 201 member evpn-instance 201 vni 10201 vlan configuration 202 member evpn-instance 202 vni 10202 vlan configuration 203 member evpn-instance 203 vni 10203 vlan configuration 204 member evpn-instance 204 vni 10204 vlan configuration 901 member vni 50901 ! vlan 101 private-vlan primary private-vlan association 102-104 ! vlan 102 private-vlan community ! vlan 103 private-vlan community ! vlan 104 private-vlan isolated ! vlan 201 private-vlan primary private-vlan association 202-204 ! vlan 202 private-vlan community ! </pre>

VTEP 1	VTEP 2	VTEP 3
<pre> vlan 203 private-vlan community ! vlan 204 private-vlan isolated ! vlan 901 ! interface Loopback0 ip address 172.16.255.3 255.255.255.255 ip pim sparse-mode 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 GigabitEthernet1/0/2 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 GigabitEthernet1/0/3 switchport access vlan 102 switchport private-vlan host-association 101 102 switchport mode private-vlan host spanning-tree portfast ! interface GigabitEthernet1/0/4 switchport access vlan 103 switchport private-vlan host-association 101 103 switchport mode private-vlan host spanning-tree portfast ! </pre>	<pre> vlan 203 private-vlan community ! vlan 204 private-vlan isolated ! vlan 901 ! 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 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 GigabitEthernet1/0/2 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 GigabitEthernet1/0/11 switchport access vlan 102 switchport private-vlan host-association 101 102 switchport mode private-vlan host spanning-tree portfast ! interface GigabitEthernet1/0/12 switchport access vlan 103 switchport private-vlan host-association 101 103 switchport mode private-vlan host spanning-tree portfast ! </pre>	<pre> vlan 203 private-vlan community ! vlan 204 private-vlan isolated ! vlan 901 ! interface Loopback0 ip address 172.16.255.5 255.255.255.255 ip pim sparse-mode 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 ! interface GigabitEthernet0/0 vrf forwarding Mgmt-vrf ip address 10.62.149.183 255.255.255.0 negotiation auto ! 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 GigabitEthernet1/0/2 no switchport ip address 172.16.25.5 255.255.255.0 ip pim sparse-mode ip ospf network point-to-point ip ospf 1 area 0 ! interface GigabitEthernet1/0/16 switchport access vlan 202 switchport private-vlan host-association 201 202 switchport mode private-vlan host spanning-tree portfast ! </pre>

VTEP 1	VTEP 2	VTEP 3
<pre> interface GigabitEthernet1/0/5 switchport access vlan 104 switchport private-vlan host-association 101 104 switchport mode private-vlan host spanning-tree portfast ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 private-vlan mapping 102-104 ! interface Vlan201 vrf forwarding green ip address 10.1.201.1 255.255.255.0 private-vlan mapping 202-204 ! 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 225.1.1.1 member vni 10102 mcast-group 225.1.1.1 member vni 10103 mcast-group 225.1.1.1 member vni 10104 mcast-group 225.1.1.1 member vni 10201 mcast-group 225.1.1.1 member vni 10202 mcast-group 225.1.1.1 member vni 10203 mcast-group 225.1.1.1 member vni 10204 mcast-group 225.1.1.1 member vni 50901 vrf green ! router ospf 1 router-id 172.16.255.3 ! </pre>	<pre> interface GigabitEthernet1/0/13 switchport access vlan 104 switchport private-vlan host-association 101 104 switchport mode private-vlan host spanning-tree portfast ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 private-vlan mapping 102-104 ! interface Vlan201 vrf forwarding green ip address 10.1.201.1 255.255.255.0 private-vlan mapping 202-204 ! 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 225.1.1.1 member vni 10102 mcast-group 225.1.1.1 member vni 10103 mcast-group 225.1.1.1 member vni 10104 mcast-group 225.1.1.1 member vni 10201 mcast-group 225.1.1.1 member vni 10202 mcast-group 225.1.1.1 member vni 10203 mcast-group 225.1.1.1 member vni 10204 mcast-group 225.1.1.1 member vni 50901 vrf green ! router ospf 1 router-id 172.16.255.4 ! </pre>	<pre> interface GigabitEthernet1/0/17 switchport access vlan 203 switchport private-vlan host-association 201 203 switchport mode private-vlan host spanning-tree portfast ! interface GigabitEthernet1/0/18 switchport access vlan 204 switchport private-vlan host-association 201 204 switchport mode private-vlan host spanning-tree portfast ! interface Vlan101 vrf forwarding green ip address 10.1.101.1 255.255.255.0 private-vlan mapping 102-104 ! interface Vlan201 vrf forwarding green ip address 10.1.201.1 255.255.255.0 private-vlan mapping 202-204 ! 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 225.1.1.1 member vni 10102 mcast-group 225.1.1.1 member vni 10103 mcast-group 225.1.1.1 member vni 10104 mcast-group 225.1.1.1 member vni 10201 mcast-group 225.1.1.1 member vni 10202 mcast-group 225.1.1.1 member vni 10203 mcast-group 225.1.1.1 member vni 10204 mcast-group 225.1.1.1 member vni 50901 vrf green ! </pre>

VTEP 1	VTEP 2	VTEP 3
<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 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 ! ip pim rp-address 172.16.255.255 ! end Leaf-01# </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 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 ! ip pim rp-address 172.16.255.255 ! end Leaf-02# </pre>	<pre> 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 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 ! ip pim rp-address 172.16.255.255 ! end Leaf-03# </pre>

Table 50: Configuring Spine Switch 1 and Spine Switch 2 for PVLAN Extension in a BGP EVPN VXLAN Fabric

Spine Switch 1	Spine Switch 2
<pre> Spine-01# show running-config hostname Spine-01 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 172.16.255.2 remote-as 65001 neighbor 172.16.255.2 update-source Loopback0 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 ! </pre>	<pre> Spine-02# show running-config hostname Spine-02 ! ip routing ! ip multicast-routing ! system mtu 9198 ! 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 ! interface GigabitEthernet1/0/3 no switchport ip address 172.16.25.2 255.255.255.0 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 neighbor 172.16.255.1 remote-as 65001 neighbor 172.16.255.1 update-source Loopback0 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 ! </pre>

Spine Switch 1	Spine Switch 2
<pre> address-family ipv4 exit-address-family ! address-family l2vpn evpn neighbor 172.16.255.2 activate neighbor 172.16.255.2 send-community both neighbor 172.16.255.2 route-reflector-client 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 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 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 route-reflector-client 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 ip msdp peer 172.16.254.1 connect-source Loopback1 remote-as 65001 ip msdp cache-sa-state ! end Spine-02# </pre>

Verifying PVLAN Extension in a BGP EVPN VXLAN Fabric

The following sections provide sample outputs for **show** commands to verify the PVLAN extension on the devices in the topology configured above:

- [Outputs to Verify the Configuration on VTEP 1, on page 533](#)
- [Outputs to Verify the Configuration on VTEP 2, on page 537](#)
- [Outputs to Verify the Configuration on VTEP 3, on page 542](#)
- [Outputs to Verify the Configuration on Spine Switch 1, on page 545](#)
- [Outputs to Verify the Configuration on Spine Switch 2, on page 547](#)

Outputs to Verify the Configuration on VTEP 1

The following example shows the output for the **show vlan private-vlan** command on VTEP 1:

```
Leaf-01# show vlan private-vlan
```

```

Primary Secondary Type Ports
-----
101 102 community Gi1/0/3
101 103 community Gi1/0/4
101 104 isolated Gi1/0/5
201 202 community
201 203 community
201 204 isolated

```

```
Leaf-01#
```

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

```
Leaf-01# show ip arp vrf green
Protocol Address           Age (min)  Hardware Addr   Type   Interface
-----
Internet 10.1.101.1             -          10b3.d56a.8fc1  ARPA   Vlan101
Internet 10.1.101.3             95         f4cf.e243.34c2  ARPA   Vlan101 pv 102
Internet 10.1.101.4             95         f4cf.e243.34c3  ARPA   Vlan101 pv 103
Internet 10.1.101.5             95         f4cf.e243.34c4  ARPA   Vlan101 pv 104
Internet 10.1.201.1             -          10b3.d56a.8fcc  ARPA   Vlan201
Internet 172.16.254.3           -          10b3.d56a.8fc8  ARPA   Vlan901

Leaf-01#
```

The following example shows the output for the **show mac address-table vlan *vlan-id*** command on VTEP 1:

```
Leaf-01# show mac address-table vlan 101
Mac Address Table
-----
Vlan    Mac Address           Type           Ports
-----
101     10b3.d56a.8fc1       STATIC         Vl101
101     7c21.0dbd.9541       STATIC         Vl101
101     f4cf.e243.34c2       DYNAMIC pv    Gi1/0/3
101     f4cf.e243.34c3       DYNAMIC pv    Gi1/0/4
101     f4cf.e243.34c4       DYNAMIC pv    Gi1/0/5
Total Mac Addresses for this criterion: 5

Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn peers vxlan

Interface VNI      Peer-IP           Num routes  eVNI      UP time
-----
nve1     10101    172.16.254.4     8           10101     01:33:29
nve1     10102    172.16.254.4     1           10102     01:33:29
nve1     10103    172.16.254.4     1           10103     01:33:29
nve1     10104    172.16.254.4     1           10104     00:01:37

Leaf-01#
```

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

```
Leaf-01# show nve peer
Interface VNI      Type Peer-IP           RMAC/Num_RTs  eVNI      state flags UP time
-----
nve1     50901    L3CP 172.16.254.5     7c21.0dbd.2748 50901     UP   A/M/4 01:33:30
nve1     50901    L3CP 172.16.254.4     7c21.0dbd.9548 50901     UP   A/M/4 01:33:29
nve1     10101    L2CP 172.16.254.4     8           10101     UP   N/A   01:33:29
nve1     10102    L2CP 172.16.254.4     1           10102     UP   N/A   01:33:29
nve1     10103    L2CP 172.16.254.4     1           10103     UP   N/A   01:33:29
nve1     10104    L2CP 172.16.254.4     1           10104     UP   N/A   00:01:37

Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn mac local
MAC Address      EVI      VLAN  ESI                               Ether Tag  Next Hop(s)
-----
f4cf.e243.34c2  101     101   0000.0000.0000.0000.0000  0          Gi1/0/3:101
f4cf.e243.34c3  101     101   0000.0000.0000.0000.0000  0          Gi1/0/4:101
f4cf.e243.34c4  101     101   0000.0000.0000.0000.0000  0          Gi1/0/5:101
f4cf.e243.34c2  102     102   0000.0000.0000.0000.0000  0          Gi1/0/3:102
f4cf.e243.34c3  103     103   0000.0000.0000.0000.0000  0          Gi1/0/4:103
f4cf.e243.34c4  104     104   0000.0000.0000.0000.0000  0          Gi1/0/5:104

Leaf-01#
```

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

```
Leaf-01# show l2vpn evpn mac remote
MAC Address      EVI      VLAN  ESI                               Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc3  101     101   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc4  101     101   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc5  101     101   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc3  102     102   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc4  103     103   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc5  104     104   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc6  201     201   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc7  201     201   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc8  201     201   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc6  202     202   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc7  203     203   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc8  204     204   0000.0000.0000.0000.0000  0          172.16.254.5

Leaf-01#
```

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

```
Leaf-01# show l2route evpn mac ip
EVI      ETag  Prod  Mac Address                       Host IP      Next Hop(s)
-----
101      0     L2VPN 10b3.d56a.8fc1                    10.1.101.1  V1101:0
101      0     BGP   44d3.ca28.6cc3                    10.1.101.13 V:10101 172.16.254.4
101      0     BGP   44d3.ca28.6cc4                    10.1.101.14 V:10101 172.16.254.4
101      0     BGP   44d3.ca28.6cc5                    10.1.101.15 V:10101 172.16.254.4
101      0     BGP   7c21.0dbd.9541                    10.1.101.1  V:10101 172.16.254.4
101      0     L2VPN f4cf.e243.34c2                    10.1.101.3  Gi1/0/3:101
101      0     L2VPN f4cf.e243.34c3                    10.1.101.4  Gi1/0/4:101
101      0     L2VPN f4cf.e243.34c4                    10.1.101.5  Gi1/0/5:101
201      0     BGP   44d3.ca28.6cc6                    10.1.102.3  V:10201 172.16.254.5
201      0     BGP   44d3.ca28.6cc7                    10.1.102.4  V:10201 172.16.254.5
201      0     BGP   44d3.ca28.6cc8                    10.1.102.5  V:10201 172.16.254.5
201      0     BGP   7c21.0dbd.274c                    10.1.201.1  V:10201 172.16.254.5

Leaf-01#
```

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

```
Leaf-01# show bgp l2vpn evpn
BGP table version is 70, local router ID is 172.16.255.3
```

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: 172.16.255.3:101					
*>	[2][172.16.255.3:101][0][48][10B3D56A8FC1]	[32]	[10.1.101.1]	/24	:: 32768 ?
*>i	[2][172.16.255.3:101][0][48][44D3CA286CC3]	[32]	[10.1.101.13]	/24	172.16.254.4 0 100 0 ?
*>i	[2][172.16.255.3:101][0][48][44D3CA286CC4]	[32]	[10.1.101.14]	/24	172.16.254.4 0 100 0 ?
*>i	[2][172.16.255.3:101][0][48][44D3CA286CC5]	[32]	[10.1.101.15]	/24	172.16.254.4 0 100 0 ?
*>i	[2][172.16.255.3:101][0][48][7C210DBD9541]	[32]	[10.1.101.1]	/24	172.16.254.4 0 100 0 ?
*>	[2][172.16.255.3:101][0][48][F4CFE24334C2]	[32]	[10.1.101.3]	/24	:: 32768 ?
*>	[2][172.16.255.3:101][0][48][F4CFE24334C3]	[32]	[10.1.101.4]	/24	:: 32768 ?
*>	[2][172.16.255.3:101][0][48][F4CFE24334C4]	[32]	[10.1.101.5]	/24	:: 32768 ?
Route Distinguisher: 172.16.255.3:102					
>i	[2][172.16.255.3:102][0][48][44D3CA286CC3]	[0]	[]	/20	172.16.254.4 0 100 0 ?
>	[2][172.16.255.3:102][0][48][F4CFE24334C2]	[0]	[]	/20	:: 32768 ?
Route Distinguisher: 172.16.255.3:103					
>i	[2][172.16.255.3:103][0][48][44D3CA286CC4]	[0]	[]	/20	172.16.254.4 0 100 0 ?
>	[2][172.16.255.3:103][0][48][F4CFE24334C3]	[0]	[]	/20	:: 32768 ?
Route Distinguisher: 172.16.255.3:104					
>i	[2][172.16.255.3:104][0][48][44D3CA286CC5]	[0]	[]	/20	172.16.254.4 0 100 0 ?
>	[2][172.16.255.3:104][0][48][F4CFE24334C4]	[0]	[]	/20	:: 32768 ?
Route Distinguisher: 172.16.255.3:201					
*>i	[2][172.16.255.3:201][0][48][44D3CA286CC6]	[32]	[10.1.102.3]	/24	172.16.254.5 0 100 0 ?
*>i	[2][172.16.255.3:201][0][48][44D3CA286CC7]	[32]	[10.1.102.4]	/24	172.16.254.5 0 100 0 ?
*>i	[2][172.16.255.3:201][0][48][44D3CA286CC8]	[32]	[10.1.102.5]	/24	172.16.254.5 0 100 0 ?
*>i	[2][172.16.255.3:201][0][48][7C210DBD274C]	[32]	[10.1.201.1]	/24	172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.3:202					
>i	[2][172.16.255.3:202][0][48][44D3CA286CC6]	[0]	[]	/20	172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.3:203					
>i	[2][172.16.255.3:203][0][48][44D3CA286CC7]	[0]	[]	/20	172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.3:204					
>i	[2][172.16.255.3:204][0][48][44D3CA286CC8]	[0]	[]	/20	172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.4:101					
*>i	[2][172.16.255.4:101][0][48][44D3CA286CC3]	[32]	[10.1.101.13]	/24	172.16.254.4 0 100 0 ?
* i					172.16.254.4 0 100 0 ?
*>i	[2][172.16.255.4:101][0][48][44D3CA286CC4]	[32]	[10.1.101.14]	/24	

```

172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
*>i [2] [172.16.255.4:101] [0] [48] [44D3CA286CC5] [32] [10.1.101.15]/24
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
*>i [2] [172.16.255.4:101] [0] [48] [7C210DBD9541] [32] [10.1.101.1]/24
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.255.4:102
*>i [2] [172.16.255.4:102] [0] [48] [44D3CA286CC3] [0] [*]/20
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.255.4:103
*>i [2] [172.16.255.4:103] [0] [48] [44D3CA286CC4] [0] [*]/20
172.16.254.4          0 100 0 ?
* i                   172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.255.4:104
* i [2] [172.16.255.4:104] [0] [48] [44D3CA286CC5] [0] [*]/20
172.16.254.4          0 100 0 ?
*>i                   172.16.254.4          0 100 0 ?
Route Distinguisher: 172.16.255.5:201
*>i [2] [172.16.255.5:201] [0] [48] [44D3CA286CC6] [32] [10.1.102.3]/24
172.16.254.5          0 100 0 ?
* i                   172.16.254.5          0 100 0 ?
*>i [2] [172.16.255.5:201] [0] [48] [44D3CA286CC7] [32] [10.1.102.4]/24
172.16.254.5          0 100 0 ?
* i                   172.16.254.5          0 100 0 ?
*>i [2] [172.16.255.5:201] [0] [48] [44D3CA286CC8] [32] [10.1.102.5]/24
172.16.254.5          0 100 0 ?
* i                   172.16.254.5          0 100 0 ?
*>i [2] [172.16.255.5:201] [0] [48] [7C210DBD274C] [32] [10.1.201.1]/24
172.16.254.5          0 100 0 ?
      Network          Next Hop          Metric LocPrf Weight Path
* i                   172.16.254.5          0 100 0 ?
Route Distinguisher: 172.16.255.5:202
*>i [2] [172.16.255.5:202] [0] [48] [44D3CA286CC6] [0] [*]/20
172.16.254.5          0 100 0 ?
* i                   172.16.254.5          0 100 0 ?
Route Distinguisher: 172.16.255.5:203
*>i [2] [172.16.255.5:203] [0] [48] [44D3CA286CC7] [0] [*]/20
172.16.254.5          0 100 0 ?
* i                   172.16.254.5          0 100 0 ?
Route Distinguisher: 172.16.255.5:204
*>i [2] [172.16.255.5:204] [0] [48] [44D3CA286CC8] [0] [*]/20
172.16.254.5          0 100 0 ?
* i                   172.16.254.5          0 100 0 ?
Route Distinguisher: 1:1 (default for vrf green)
*> [5] [1:1] [0] [24] [10.1.101.0]/17
0.0.0.0                0 32768 ?
*>i [5] [1:1] [0] [24] [10.1.201.0]/17
172.16.254.5          0 100 0 ?
* i                   172.16.254.5          0 100 0 ?
Leaf-01#

```

Outputs to Verify the Configuration on VTEP 2

The following example shows the output for the **show vlan private-vlan** command on VTEP 2:

```

Leaf-02# show vlan private-vlan
Primary Secondary Type          Ports
-----

```

```

101    102    community    Gi1/0/11
101    103    community    Gi1/0/12
101    104    isolated     Gi1/0/13
201    202    community
201    203    community
201    204    isolated

```

```
Leaf-02#
```

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

```
Leaf-02# show ip arp vrf green
Protocol Address          Age (min)  Hardware Addr  Type   Interface
Internet 10.1.101.1        -          7c21.0dbd.9541 ARPA   Vlan101
Internet 10.1.101.13       95         44d3.ca28.6cc3 ARPA   Vlan101 pv 102
Internet 10.1.101.14       95         44d3.ca28.6cc4 ARPA   Vlan101 pv 103
Internet 10.1.101.15       95         44d3.ca28.6cc5 ARPA   Vlan101 pv 104
Internet 10.1.201.1        -          7c21.0dbd.954c ARPA   Vlan201
Internet 172.16.254.4      -          7c21.0dbd.9548 ARPA   Vlan901

```

```
Leaf-02#
```

The following example shows the output for the **show mac address-table vlan *vlan-id*** command on VTEP 2:

```
Leaf-02# show mac address-table vlan 101
Mac Address Table
-----
Vlan    Mac Address          Type           Ports
----    -
101     10b3.d56a.8fc1      STATIC         Vl101
101     44d3.ca28.6cc3      DYNAMIC pv    Gi1/0/11
101     44d3.ca28.6cc4      DYNAMIC pv    Gi1/0/12
101     44d3.ca28.6cc5      DYNAMIC pv    Gi1/0/13
101     7c21.0dbd.9541      STATIC         Vl101
Total Mac Addresses for this criterion: 5

```

```
Leaf-02#
```

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

```
Leaf-02# show l2vpn evpn peers vxlan
Interface VNI      Peer-IP          Num routes  eVNI      UP time
-----
nve1     10101    172.16.254.3    8           10101     01:34:10
nve1     10102    172.16.254.3    1           10102     01:34:10
nve1     10103    172.16.254.3    1           10103     01:34:10
nve1     10104    172.16.254.3    1           10104     00:02:13

```

```
Leaf-02#
```

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

```
Leaf-02# show nve peer
Interface VNI      Type Peer-IP          RMAC/Num_RTs  eVNI      state flags UP time
nve1     50901    L3CP 172.16.254.3    10b3.d56a.8fc8 50901     UP   A/M/4 01:34:10
nve1     50901    L3CP 172.16.254.5    7c21.0dbd.2748 50901     UP   A/M/4 01:34:10

```

```

nve1      10101    L2CP 172.16.254.3    8          10101    UP    N/A    01:34:10
nve1      10102    L2CP 172.16.254.3    1          10102    UP    N/A    01:34:10
nve1      10103    L2CP 172.16.254.3    1          10103    UP    N/A    01:34:10
nve1      10104    L2CP 172.16.254.3    1          10104    UP    N/A    00:02:13

```

Leaf-02#

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

```

Leaf-02# show l2vpn evpn mac local
MAC Address      EVI    VLAN  ESI                               Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc3  101    101   0000.0000.0000.0000.0000  0          Gi1/0/11:101
44d3.ca28.6cc4  101    101   0000.0000.0000.0000.0000  0          Gi1/0/12:101
44d3.ca28.6cc5  101    101   0000.0000.0000.0000.0000  0          Gi1/0/13:101
44d3.ca28.6cc3  102    102   0000.0000.0000.0000.0000  0          Gi1/0/11:102
44d3.ca28.6cc4  103    103   0000.0000.0000.0000.0000  0          Gi1/0/12:103
44d3.ca28.6cc5  104    104   0000.0000.0000.0000.0000  0          Gi1/0/13:104

```

Leaf-02#

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

```

Leaf-02# show l2vpn evpn mac remote
MAC Address      EVI    VLAN  ESI                               Ether Tag  Next Hop(s)
-----
f4cf.e243.34c2  101    101   0000.0000.0000.0000.0000  0          172.16.254.3
f4cf.e243.34c3  101    101   0000.0000.0000.0000.0000  0          172.16.254.3
f4cf.e243.34c4  101    101   0000.0000.0000.0000.0000  0          172.16.254.3
f4cf.e243.34c2  102    102   0000.0000.0000.0000.0000  0          172.16.254.3
f4cf.e243.34c3  103    103   0000.0000.0000.0000.0000  0          172.16.254.3
f4cf.e243.34c4  104    104   0000.0000.0000.0000.0000  0          172.16.254.3
44d3.ca28.6cc6  201    201   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc7  201    201   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc8  201    201   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc6  202    202   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc7  203    203   0000.0000.0000.0000.0000  0          172.16.254.5
44d3.ca28.6cc8  204    204   0000.0000.0000.0000.0000  0          172.16.254.5

```

Leaf-02#

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

```

Leaf-02# show l2route evpn mac ip
EVI    ETag  Prod  Mac Address                       Host IP      Next Hop(s)
-----
101    0     BGP  10b3.d56a.8fc1                    10.1.101.1  V:10101 172.16.254.3
101    0     L2VPN 44d3.ca28.6cc3                    10.1.101.13 Gi1/0/11:101
101    0     L2VPN 44d3.ca28.6cc4                    10.1.101.14 Gi1/0/12:101
101    0     L2VPN 44d3.ca28.6cc5                    10.1.101.15 Gi1/0/13:101
101    0     L2VPN 7c21.0dbd.9541                    10.1.101.1  V1101:0
101    0     BGP  f4cf.e243.34c2                    10.1.101.3  V:10101 172.16.254.3
101    0     BGP  f4cf.e243.34c3                    10.1.101.4  V:10101 172.16.254.3
101    0     BGP  f4cf.e243.34c4                    10.1.101.5  V:10101 172.16.254.3
201    0     BGP  44d3.ca28.6cc6                    10.1.102.3  V:10201 172.16.254.5
201    0     BGP  44d3.ca28.6cc7                    10.1.102.4  V:10201 172.16.254.5
201    0     BGP  44d3.ca28.6cc8                    10.1.102.5  V:10201 172.16.254.5
201    0     BGP  7c21.0dbd.274c                    10.1.201.1  V:10201 172.16.254.5

```

Leaf-02#

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

```
Leaf-02# show bgp l2vpn evpn
BGP table version is 65, local router ID is 172.16.255.4
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: 172.16.255.3:101
* i [2][172.16.255.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
    172.16.254.3          0 100 0 ?
*>i      172.16.254.3          0 100 0 ?
* i [2][172.16.255.3:101][0][48][F4CFE24334C2][32][10.1.101.3]/24
    172.16.254.3          0 100 0 ?
*>i      172.16.254.3          0 100 0 ?
* i [2][172.16.255.3:101][0][48][F4CFE24334C3][32][10.1.101.4]/24
    172.16.254.3          0 100 0 ?
*>i      172.16.254.3          0 100 0 ?
* i [2][172.16.255.3:101][0][48][F4CFE24334C4][32][10.1.101.5]/24
    172.16.254.3          0 100 0 ?
*>i      172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.255.3:102
* i [2][172.16.255.3:102][0][48][F4CFE24334C2][0][*]/20
    172.16.254.3          0 100 0 ?
*>i      172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.255.3:103
* i [2][172.16.255.3:103][0][48][F4CFE24334C3][0][*]/20
    172.16.254.3          0 100 0 ?
*>i      172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.255.3:104
*>i [2][172.16.255.3:104][0][48][F4CFE24334C4][0][*]/20
    172.16.254.3          0 100 0 ?
* i      172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.255.4:101
*>i [2][172.16.255.4:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
    172.16.254.3          0 100 0 ?
*> [2][172.16.255.4:101][0][48][44D3CA286CC3][32][10.1.101.13]/24
    ::                    32768 ?
*> [2][172.16.255.4:101][0][48][44D3CA286CC4][32][10.1.101.14]/24
    ::                    32768 ?
*> [2][172.16.255.4:101][0][48][44D3CA286CC5][32][10.1.101.15]/24
    ::                    32768 ?
*> [2][172.16.255.4:101][0][48][7C210DBD9541][32][10.1.101.1]/24
    ::                    32768 ?
*>i [2][172.16.255.4:101][0][48][F4CFE24334C2][32][10.1.101.3]/24
    172.16.254.3          0 100 0 ?
*>i [2][172.16.255.4:101][0][48][F4CFE24334C3][32][10.1.101.4]/24
      Network          Next Hop          Metric LocPrf Weight Path
*>i [2][172.16.255.4:101][0][48][F4CFE24334C4][32][10.1.101.5]/24
    172.16.254.3          0 100 0 ?
Route Distinguisher: 172.16.255.4:102
*> [2][172.16.255.4:102][0][48][44D3CA286CC3][0][*]/20
    ::                    32768 ?
*>i [2][172.16.255.4:102][0][48][F4CFE24334C2][0][*]/20
    172.16.254.3          0 100 0 ?
```



```

Route Distinguisher: 172.16.255.4:103
*> [2][172.16.255.4:103][0][48][44D3CA286CC4][0][*]/20
:: 32768 ?
*>i [2][172.16.255.4:103][0][48][F4CFE24334C3][0][*]/20
172.16.254.3 0 100 0 ?
Route Distinguisher: 172.16.255.4:104
*> [2][172.16.255.4:104][0][48][44D3CA286CC5][0][*]/20
:: 32768 ?
*>i [2][172.16.255.4:104][0][48][F4CFE24334C4][0][*]/20
172.16.254.3 0 100 0 ?
Route Distinguisher: 172.16.255.4:201
*>i [2][172.16.255.4:201][0][48][44D3CA286CC6][32][10.1.102.3]/24
172.16.254.5 0 100 0 ?
*>i [2][172.16.255.4:201][0][48][44D3CA286CC7][32][10.1.102.4]/24
172.16.254.5 0 100 0 ?
*>i [2][172.16.255.4:201][0][48][44D3CA286CC8][32][10.1.102.5]/24
172.16.254.5 0 100 0 ?
*>i [2][172.16.255.4:201][0][48][7C210DBD274C][32][10.1.201.1]/24
172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.4:202
*>i [2][172.16.255.4:202][0][48][44D3CA286CC6][0][*]/20
172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.4:203
*>i [2][172.16.255.4:203][0][48][44D3CA286CC7][0][*]/20
172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.4:204
*>i [2][172.16.255.4:204][0][48][44D3CA286CC8][0][*]/20
172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.5:201
*>i [2][172.16.255.5:201][0][48][44D3CA286CC6][32][10.1.102.3]/24
172.16.254.5 0 100 0 ?
* i 172.16.254.5 0 100 0 ?
*>i [2][172.16.255.5:201][0][48][44D3CA286CC7][32][10.1.102.4]/24
172.16.254.5 0 100 0 ?
* i 172.16.254.5 0 100 0 ?
*>i [2][172.16.255.5:201][0][48][44D3CA286CC8][32][10.1.102.5]/24
172.16.254.5 0 100 0 ?
* i 172.16.254.5 0 100 0 ?
*>i [2][172.16.255.5:201][0][48][7C210DBD274C][32][10.1.201.1]/24
172.16.254.5 0 100 0 ?
Network Next Hop Metric LocPrf Weight Path
* i 172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.5:202
*>i [2][172.16.255.5:202][0][48][44D3CA286CC6][0][*]/20
172.16.254.5 0 100 0 ?
* i 172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.5:203
*>i [2][172.16.255.5:203][0][48][44D3CA286CC7][0][*]/20
172.16.254.5 0 100 0 ?
* i 172.16.254.5 0 100 0 ?
Route Distinguisher: 172.16.255.5:204
*>i [2][172.16.255.5:204][0][48][44D3CA286CC8][0][*]/20
172.16.254.5 0 100 0 ?
* i 172.16.254.5 0 100 0 ?
Route Distinguisher: 1:1 (default for vrf green)
* i [5][1:1][0][24][10.1.101.0]/17
172.16.254.3 0 100 0 ?
* i 172.16.254.3 0 100 0 ?
*> 0.0.0.0 0 32768 ?
*>i [5][1:1][0][24][10.1.201.0]/17
172.16.254.5 0 100 0 ?
* i 172.16.254.5 0 100 0 ?
Leaf-02#

```

Outputs to Verify the Configuration on VTEP 3

The following example shows the output for the **show vlan private-vlan** command on VTEP 3:

```
Leaf-03# show vlan private-vlan
```

Primary	Secondary	Type	Ports
101	102	community	
101	103	community	
101	104	isolated	
201	202	community	Gi1/0/16
201	203	community	Gi1/0/17
201	204	isolated	Gi1/0/18

```
Leaf-03#
```

The following example shows the output for the **show ip arp vrf green** command on VTEP 3:

```
Leaf-03# show ip arp vrf green
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.101.1	-	7c21.0dbd.2741	ARPA	Vlan101
Internet	10.1.201.1	-	7c21.0dbd.274c	ARPA	Vlan201
Internet	172.16.254.5	-	7c21.0dbd.2748	ARPA	Vlan901

```
Leaf-03#
```

The following example shows the output for the **show mac address-table vlan *vlan-id*** command on VTEP 3:

```
Leaf-03# show mac address-table vlan 101
```

```
Mac Address Table
```

Vlan	Mac Address	Type	Ports
101	7c21.0dbd.2741	STATIC	Vl101

Total Mac Addresses for this criterion: 1

```
Leaf-03#
```

The following example shows the output for the **show l2vpn evpn peers vxlan** command on VTEP 3:

```
Leaf-03# show l2vpn evpn peers vxlan
```

```
Leaf-03#
```

The following example shows the output for the **show nve peer** command on VTEP 3:

```
Leaf-03# show nve peer
```

Interface	VNI	Type	Peer-IP	RMAC/Num_RTs	eVNI	state	flags	UP time
nve1	50901	L3CP	172.16.254.3	10b3.d56a.8fc8	50901	UP	A/M/4	01:34:51
nve1	50901	L3CP	172.16.254.4	7c21.0dbd.9548	50901	UP	A/M/4	01:34:51

```
Leaf-03#
```

The following example shows the output for the **show l2vpn evpn mac local** command on VTEP 3:

```
Leaf-03# show l2vpn evpn mac local
MAC Address      EVI      VLAN  ESI                               Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc6  201      201   0000.0000.0000.0000.0000  0          Gi1/0/16:201
44d3.ca28.6cc7  201      201   0000.0000.0000.0000.0000  0          Gi1/0/17:201
44d3.ca28.6cc8  201      201   0000.0000.0000.0000.0000  0          Gi1/0/18:201
44d3.ca28.6cc6  202      202   0000.0000.0000.0000.0000  0          Gi1/0/16:202
44d3.ca28.6cc7  203      203   0000.0000.0000.0000.0000  0          Gi1/0/17:203
44d3.ca28.6cc8  204      204   0000.0000.0000.0000.0000  0          Gi1/0/18:204

Leaf-03#
```

The following example shows the output for the **show l2vpn evpn mac remote** command on VTEP 3:

```
Leaf-03# show l2vpn evpn mac remote
MAC Address      EVI      VLAN  ESI                               Ether Tag  Next Hop(s)
-----
44d3.ca28.6cc3  101      101   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc4  101      101   0000.0000.0000.0000.0000  0          172.16.254.4
44d3.ca28.6cc5  101      101   0000.0000.0000.0000.0000  0          172.16.254.4
f4cf.e243.34c2  101      101   0000.0000.0000.0000.0000  0          172.16.254.3
f4cf.e243.34c3  101      101   0000.0000.0000.0000.0000  0          172.16.254.3
f4cf.e243.34c4  101      101   0000.0000.0000.0000.0000  0          172.16.254.3
44d3.ca28.6cc3  102      102   0000.0000.0000.0000.0000  0          172.16.254.4
f4cf.e243.34c2  102      102   0000.0000.0000.0000.0000  0          172.16.254.3
44d3.ca28.6cc4  103      103   0000.0000.0000.0000.0000  0          172.16.254.4
f4cf.e243.34c3  103      103   0000.0000.0000.0000.0000  0          172.16.254.3
44d3.ca28.6cc5  104      104   0000.0000.0000.0000.0000  0          172.16.254.4
f4cf.e243.34c4  104      104   0000.0000.0000.0000.0000  0          172.16.254.3

Leaf-03#
```

The following example shows the output for the **show l2route evpn mac ip** command on VTEP 3:

```
Leaf-03# show l2route evpn mac ip
EVI      ETag  Prod  Mac Address                       Host IP      Next Hop(s)
-----
101      0     BGP  10b3.d56a.8fc1                    10.1.101.1  V:10101 172.16.254.3
101      0     BGP  44d3.ca28.6cc3                    10.1.101.13 V:10101 172.16.254.4
101      0     BGP  44d3.ca28.6cc4                    10.1.101.14 V:10101 172.16.254.4
101      0     BGP  44d3.ca28.6cc5                    10.1.101.15 V:10101 172.16.254.4
101      0     BGP  7c21.0dbd.9541                    10.1.101.1  V:10101 172.16.254.4
101      0     BGP  f4cf.e243.34c2                    10.1.101.3  V:10101 172.16.254.3
101      0     BGP  f4cf.e243.34c3                    10.1.101.4  V:10101 172.16.254.3
101      0     BGP  f4cf.e243.34c4                    10.1.101.5  V:10101 172.16.254.3
201      0     L2VPN 44d3.ca28.6cc6                    10.1.102.3  Gi1/0/16:201
201      0     L2VPN 44d3.ca28.6cc7                    10.1.102.4  Gi1/0/17:201
201      0     L2VPN 44d3.ca28.6cc8                    10.1.102.5  Gi1/0/18:201
201      0     L2VPN 7c21.0dbd.274c                    10.1.201.1  V1201:0

Leaf-03#
```

The following example shows the output for the **show bgp l2vpn evpn** command on VTEP 3:

```
Leaf-03# show bgp l2vpn evpn
BGP table version is 82, local router ID is 172.16.255.5
```

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: 172.16.255.3:101					
* i [2][172.16.255.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24	172.16.254.3	0	100	0	?
*>i	172.16.254.3	0	100	0	?
* i [2][172.16.255.3:101][0][48][F4CFE24334C2][32][10.1.101.3]/24	172.16.254.3	0	100	0	?
*>i	172.16.254.3	0	100	0	?
* i [2][172.16.255.3:101][0][48][F4CFE24334C3][32][10.1.101.4]/24	172.16.254.3	0	100	0	?
*>i	172.16.254.3	0	100	0	?
* i [2][172.16.255.3:101][0][48][F4CFE24334C4][32][10.1.101.5]/24	172.16.254.3	0	100	0	?
*>i	172.16.254.3	0	100	0	?
Route Distinguisher: 172.16.255.3:102					
* i [2][172.16.255.3:102][0][48][F4CFE24334C2][0][*]/20	172.16.254.3	0	100	0	?
*>i	172.16.254.3	0	100	0	?
Route Distinguisher: 172.16.255.3:103					
* i [2][172.16.255.3:103][0][48][F4CFE24334C3][0][*]/20	172.16.254.3	0	100	0	?
*>i	172.16.254.3	0	100	0	?
Route Distinguisher: 172.16.255.3:104					
>i [2][172.16.255.3:104][0][48][F4CFE24334C4][0][]/20	172.16.254.3	0	100	0	?
* i	172.16.254.3	0	100	0	?
Route Distinguisher: 172.16.255.4:101					
*>i [2][172.16.255.4:101][0][48][44D3CA286CC3][32][10.1.101.13]/24	172.16.254.4	0	100	0	?
* i	172.16.254.4	0	100	0	?
*>i [2][172.16.255.4:101][0][48][44D3CA286CC4][32][10.1.101.14]/24	172.16.254.4	0	100	0	?
* i	172.16.254.4	0	100	0	?
*>i [2][172.16.255.4:101][0][48][44D3CA286CC5][32][10.1.101.15]/24	172.16.254.4	0	100	0	?
* i	172.16.254.4	0	100	0	?
*>i [2][172.16.255.4:101][0][48][7C210DBD9541][32][10.1.101.1]/24	172.16.254.4	0	100	0	?
* i	172.16.254.4	0	100	0	?
Route Distinguisher: 172.16.255.4:102					
>i [2][172.16.255.4:102][0][48][44D3CA286CC3][0][]/20	172.16.254.4	0	100	0	?
* i	172.16.254.4	0	100	0	?
Route Distinguisher: 172.16.255.4:103					
>i [2][172.16.255.4:103][0][48][44D3CA286CC4][0][]/20	172.16.254.4	0	100	0	?
* i	172.16.254.4	0	100	0	?
Route Distinguisher: 172.16.255.4:104					
* i [2][172.16.255.4:104][0][48][44D3CA286CC5][0][*]/20	172.16.254.4	0	100	0	?
*>i	172.16.254.4	0	100	0	?
Route Distinguisher: 172.16.255.5:101					
*>i [2][172.16.255.5:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24	172.16.254.3	0	100	0	?
*>i [2][172.16.255.5:101][0][48][44D3CA286CC3][32][10.1.101.13]/24	172.16.254.4	0	100	0	?

```

*>i [2] [172.16.255.5:101] [0] [48] [44D3CA286CC4] [32] [10.1.101.14]/24
      172.16.254.4          0      100      0 ?
*>i [2] [172.16.255.5:101] [0] [48] [44D3CA286CC5] [32] [10.1.101.15]/24
      172.16.254.4          0      100      0 ?
*>i [2] [172.16.255.5:101] [0] [48] [7C210DBD9541] [32] [10.1.101.1]/24
      172.16.254.4          0      100      0 ?
*>i [2] [172.16.255.5:101] [0] [48] [F4CFE24334C2] [32] [10.1.101.3]/24
      172.16.254.3          0      100      0 ?
*>i [2] [172.16.255.5:101] [0] [48] [F4CFE24334C3] [32] [10.1.101.4]/24
      172.16.254.3          0      100      0 ?
*>i [2] [172.16.255.5:101] [0] [48] [F4CFE24334C4] [32] [10.1.101.5]/24
      172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.5:102
*>i [2] [172.16.255.5:102] [0] [48] [44D3CA286CC3] [0] [*]/20
      172.16.254.4          0      100      0 ?
*>i [2] [172.16.255.5:102] [0] [48] [F4CFE24334C2] [0] [*]/20
      172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.5:103
*>i [2] [172.16.255.5:103] [0] [48] [44D3CA286CC4] [0] [*]/20
      172.16.254.4          0      100      0 ?
*>i [2] [172.16.255.5:103] [0] [48] [F4CFE24334C3] [0] [*]/20
      172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.5:104
*>i [2] [172.16.255.5:104] [0] [48] [44D3CA286CC5] [0] [*]/20
      172.16.254.4          0      100      0 ?
*>i [2] [172.16.255.5:104] [0] [48] [F4CFE24334C4] [0] [*]/20
      172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.5:201
*> [2] [172.16.255.5:201] [0] [48] [44D3CA286CC6] [32] [10.1.102.3]/24
      ::                      32768 ?
*> [2] [172.16.255.5:201] [0] [48] [44D3CA286CC7] [32] [10.1.102.4]/24
      ::                      32768 ?
      Network      Next Hop      Metric LocPrf Weight Path
*> [2] [172.16.255.5:201] [0] [48] [44D3CA286CC8] [32] [10.1.102.5]/24
      ::                      32768 ?
*> [2] [172.16.255.5:201] [0] [48] [7C210DBD274C] [32] [10.1.201.1]/24
      ::                      32768 ?
Route Distinguisher: 172.16.255.5:202
*> [2] [172.16.255.5:202] [0] [48] [44D3CA286CC6] [0] [*]/20
      ::                      32768 ?
Route Distinguisher: 172.16.255.5:203
*> [2] [172.16.255.5:203] [0] [48] [44D3CA286CC7] [0] [*]/20
      ::                      32768 ?
Route Distinguisher: 172.16.255.5:204
*> [2] [172.16.255.5:204] [0] [48] [44D3CA286CC8] [0] [*]/20
      ::                      32768 ?
Route Distinguisher: 1:1 (default for vrf green)
* i [5] [1:1] [0] [24] [10.1.101.0]/17
      172.16.254.3          0      100      0 ?
*>i [5] [1:1] [0] [24] [10.1.101.0]/17
      172.16.254.3          0      100      0 ?
*> [5] [1:1] [0] [24] [10.1.201.0]/17
      0.0.0.0                0      32768 ?
Leaf-03#

```

Outputs to Verify the Configuration on Spine Switch 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 113, main routing table version 113
23 network entries using 8832 bytes of memory
47 path entries using 10528 bytes of memory
15/14 BGP path/bestpath attribute entries using 4440 bytes of memory
3 BGP rinfo entries using 120 bytes of memory
15 BGP extended community entries using 720 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 24640 total bytes of memory
BGP activity 47/24 prefixes, 107/60 paths, scan interval 60 secs
25 networks peaked at 13:03:03 Feb 19 2021 UTC (03:26:23.575 ago)

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.255.2	4	65001	259	261	113	0	0	03:27:45	23
172.16.255.3	4	65001	240	250	113	0	0	03:27:49	8
172.16.255.4	4	65001	238	258	113	0	0	03:27:25	8
172.16.255.5	4	65001	236	258	113	0	0	03:27:19	8

Spine-01#

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

```

Spine-01# show bgp l2vpn evpn
BGP table version is 113, local router ID is 172.16.255.1
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: 172.16.255.3:101
* i [2] [172.16.255.3:101] [0] [48] [10B3D56A8FC1] [32] [10.1.101.1]/24
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.255.3:101] [0] [48] [F4CFE24334C2] [32] [10.1.101.3]/24
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.255.3:101] [0] [48] [F4CFE24334C3] [32] [10.1.101.4]/24
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
* i [2] [172.16.255.3:101] [0] [48] [F4CFE24334C4] [32] [10.1.101.5]/24
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.3:102
* i [2] [172.16.255.3:102] [0] [48] [F4CFE24334C2] [0] [*]/20
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.3:103
* i [2] [172.16.255.3:103] [0] [48] [F4CFE24334C3] [0] [*]/20
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.3:104
* i [2] [172.16.255.3:104] [0] [48] [F4CFE24334C4] [0] [*]/20
      172.16.254.3          0      100      0 ?
*>i          172.16.254.3          0      100      0 ?
Route Distinguisher: 172.16.255.4:101
* i [2] [172.16.255.4:101] [0] [48] [44D3CA286CC3] [32] [10.1.101.13]/24
      172.16.254.4          0      100      0 ?
*>i          172.16.254.4          0      100      0 ?
* i [2] [172.16.255.4:101] [0] [48] [44D3CA286CC4] [32] [10.1.101.14]/24

```

```

172.16.254.4          0 100 0 ?
*>i 172.16.254.4      0 100 0 ?
* i [2][172.16.255.4:101][0][48][44D3CA286CC5][32][10.1.101.15]/24
172.16.254.4          0 100 0 ?
*>i 172.16.254.4      0 100 0 ?
* i [2][172.16.255.4:101][0][48][7C210DBD9541][32][10.1.101.1]/24
172.16.254.4          0 100 0 ?
*>i 172.16.254.4      0 100 0 ?
Route Distinguisher: 172.16.255.4:102
  Network      Next Hop      Metric LocPrf Weight Path
* i [2][172.16.255.4:102][0][48][44D3CA286CC3][0][*]/20
172.16.254.4          0 100 0 ?
*>i 172.16.254.4      0 100 0 ?
Route Distinguisher: 172.16.255.4:103
* i [2][172.16.255.4:103][0][48][44D3CA286CC4][0][*]/20
172.16.254.4          0 100 0 ?
*>i 172.16.254.4      0 100 0 ?
Route Distinguisher: 172.16.255.4:104
* i [2][172.16.255.4:104][0][48][44D3CA286CC5][0][*]/20
172.16.254.4          0 100 0 ?
*>i 172.16.254.4      0 100 0 ?
Route Distinguisher: 172.16.255.5:201
* i [2][172.16.255.5:201][0][48][44D3CA286CC6][32][10.1.102.3]/24
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?
* i [2][172.16.255.5:201][0][48][44D3CA286CC7][32][10.1.102.4]/24
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?
* i [2][172.16.255.5:201][0][48][44D3CA286CC8][32][10.1.102.5]/24
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?
* i [2][172.16.255.5:201][0][48][7C210DBD274C][32][10.1.201.1]/24
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?
Route Distinguisher: 172.16.255.5:202
* i [2][172.16.255.5:202][0][48][44D3CA286CC6][0][*]/20
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?
Route Distinguisher: 172.16.255.5:203
* i [2][172.16.255.5:203][0][48][44D3CA286CC7][0][*]/20
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?
Route Distinguisher: 172.16.255.5:204
* i [2][172.16.255.5:204][0][48][44D3CA286CC8][0][*]/20
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?
Route Distinguisher: 1:1
* i [5][1:1][0][24][10.1.101.0]/17
172.16.254.4          0 100 0 ?
* i 172.16.254.3      0 100 0 ?
*>i 172.16.254.3      0 100 0 ?
* i [5][1:1][0][24][10.1.201.0]/17
172.16.254.5          0 100 0 ?
*>i 172.16.254.5      0 100 0 ?

Spine-01#

```

Outputs to Verify the Configuration on Spine Switch 2

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 113, main routing table version 113
23 network entries using 8832 bytes of memory
47 path entries using 10528 bytes of memory
15/14 BGP path/bestpath attribute entries using 4440 bytes of memory
3 BGP rrinfo entries using 120 bytes of memory
15 BGP extended community entries using 720 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 24640 total bytes of memory
BGP activity 46/23 prefixes, 107/60 paths, scan interval 60 secs
25 networks peaked at 13:03:07 Feb 19 2021 UTC (03:27:53.810 ago)

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
172.16.255.1  4      65001   263   261    113    0    0 03:29:16    23
172.16.255.3  4      65001   243   251    113    0    0 03:29:17     8
172.16.255.4  4      65001   240   259    113    0    0 03:28:48     8
172.16.255.5  4      65001   240   257    113    0    0 03:28:45     8

Spine-02#

```

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

```

Spine-02# show bgp l2vpn evpn
BGP table version is 113, local router ID is 172.16.255.2
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: 172.16.255.3:101
* i [2][172.16.255.3:101][0][48][10B3D56A8FC1][32][10.1.101.1]/24
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
* i [2][172.16.255.3:101][0][48][F4CFE24334C2][32][10.1.101.3]/24
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
* i [2][172.16.255.3:101][0][48][F4CFE24334C3][32][10.1.101.4]/24
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
* i [2][172.16.255.3:101][0][48][F4CFE24334C4][32][10.1.101.5]/24
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
Route Distinguisher: 172.16.255.3:102
* i [2][172.16.255.3:102][0][48][F4CFE24334C2][0][*]/20
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
Route Distinguisher: 172.16.255.3:103
* i [2][172.16.255.3:103][0][48][F4CFE24334C3][0][*]/20
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
Route Distinguisher: 172.16.255.3:104
* i [2][172.16.255.3:104][0][48][F4CFE24334C4][0][*]/20
      172.16.254.3      0 100 0 ?
*>i      172.16.254.3      0 100 0 ?
Route Distinguisher: 172.16.255.4:101
* i [2][172.16.255.4:101][0][48][44D3CA286CC3][32][10.1.101.13]/24
      172.16.254.4      0 100 0 ?

```



```

*>i          172.16.254.4          0    100    0 ?
* i   [2] [172.16.255.4:101] [0] [48] [44D3CA286CC4] [32] [10.1.101.14]/24
      172.16.254.4          0    100    0 ?
*>i          172.16.254.4          0    100    0 ?
* i   [2] [172.16.255.4:101] [0] [48] [44D3CA286CC5] [32] [10.1.101.15]/24
      172.16.254.4          0    100    0 ?
*>i          172.16.254.4          0    100    0 ?
* i   [2] [172.16.255.4:101] [0] [48] [7C210DBD9541] [32] [10.1.101.1]/24
      172.16.254.4          0    100    0 ?
*>i          172.16.254.4          0    100    0 ?
Route Distinguisher: 172.16.255.4:102
  Network      Next Hop          Metric LocPrf Weight Path
* i   [2] [172.16.255.4:102] [0] [48] [44D3CA286CC3] [0] [*]/20
      172.16.254.4          0    100    0 ?
*>i          172.16.254.4          0    100    0 ?
Route Distinguisher: 172.16.255.4:103
* i   [2] [172.16.255.4:103] [0] [48] [44D3CA286CC4] [0] [*]/20
      172.16.254.4          0    100    0 ?
*>i          172.16.254.4          0    100    0 ?
Route Distinguisher: 172.16.255.4:104
* i   [2] [172.16.255.4:104] [0] [48] [44D3CA286CC5] [0] [*]/20
      172.16.254.4          0    100    0 ?
*>i          172.16.254.4          0    100    0 ?
Route Distinguisher: 172.16.255.5:201
* i   [2] [172.16.255.5:201] [0] [48] [44D3CA286CC6] [32] [10.1.102.3]/24
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?
* i   [2] [172.16.255.5:201] [0] [48] [44D3CA286CC7] [32] [10.1.102.4]/24
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?
* i   [2] [172.16.255.5:201] [0] [48] [44D3CA286CC8] [32] [10.1.102.5]/24
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?
* i   [2] [172.16.255.5:201] [0] [48] [7C210DBD274C] [32] [10.1.201.1]/24
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?
Route Distinguisher: 172.16.255.5:202
* i   [2] [172.16.255.5:202] [0] [48] [44D3CA286CC6] [0] [*]/20
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?
Route Distinguisher: 172.16.255.5:203
* i   [2] [172.16.255.5:203] [0] [48] [44D3CA286CC7] [0] [*]/20
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?
Route Distinguisher: 172.16.255.5:204
* i   [2] [172.16.255.5:204] [0] [48] [44D3CA286CC8] [0] [*]/20
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?
Route Distinguisher: 1:1
* i   [5] [1:1] [0] [24] [10.1.101.0]/17
      172.16.254.4          0    100    0 ?
* i          172.16.254.3          0    100    0 ?
*>i          172.16.254.3          0    100    0 ?
* i   [5] [1:1] [0] [24] [10.1.201.0]/17
      172.16.254.5          0    100    0 ?
*>i          172.16.254.5          0    100    0 ?

```

Spine-02#



CHAPTER 12

Scale and Performance Capabilities for BGP EVPN VXLAN on Cisco Catalyst 9500 Series Switches

- [Scale and Performance Capabilities for BGP EVPN VXLAN on Cisco Catalyst 9500 Series Switches, on page 551](#)

Scale and Performance Capabilities for BGP EVPN VXLAN on Cisco Catalyst 9500 Series Switches

This section provides information about the platform capabilities and the scale values and for various components of a BGP EVPN VXLAN fabric. The numbers in the following table depend on the ternary content addressable memory (TCAM) scale. Use the SDM configuration templates to adjust the Layer 2 and Layer 3 TCAM forwarding table sizes based on the platform positioning. For more information, see *Configuring SDM Templates* module of the *System Management Configuration Guide* for the applicable release.

The scale values listed here are validated with a unidimensional configuration. The values provided in these tables focus on the scalability of one particular feature at a time.

The scale values are applicable to all the models, unless otherwise stated.

Product Specification								
Switch Model	C9500-12Q, C9500-24Q, C9500-16X, C9500-40X				C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C			
BGP EVPN / VXLAN -- Leaf Scale								
SDM Template	Distribution	Core	NAT	Custom	Distribution	Core	NAT	Custom
VXLAN Leaf Node (Per Fabric Domain)	500	500	500	500	500	500	500	500
VRF	1000	1000	1000	1000	1000	1000	1000	1000

Product Specification								
Switch Model	C9500-12Q, C9500-24Q, C9500-16X, C9500-40X				C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C			
Layer 2 Virtual Network Instances (VNIs)	512	512	512	512	512	512	512	512
Layer 3 VNIs	512	512	512	512	512	512	512	512
Layer 3 VRF SVI Interface	4094	4094	4094	4094	4094	4094	4094	4094
MAC Entries (Local L2 Network)	64000	16000	16000	128000	82000	32000	32000	128000
MAC Entries (Remote EVPN VXLAN Network)	64000	16000	16000	128000	82000	32000	32000	128000
Overlay IPv4 routes (LPM or Indirect Routes)	64000	64000	64000	256000	114000	212000	212000	256000
Overlay IPv4 Host routes	48000	32000	48000	256000	114000	212000	212000	256000
Overlay IPv6 routes (LPM or Indirect Routes)	64000	64000	64000	256000	114000	212000	212000	256000
Overlay IPv6 Host routes	48000	32000	48000	256000	114000	212000	212000	256000
Tenant Routed Multicast (TRM) IPv4	16000	16000	16000	32000	2000	2000	2000	32000
TRM IPv6	16000	16000	16000	32000	2000	2000	2000	32000
TRMv4: Overlay Multicast Routes (*,G and S,G)	16000	32000	32000	32000	16000	32000	32000	32000
TRMv6: Overlay Multicast Routes (*,G and S,G)	16000	32000	32000	32000	16000	32000	32000	32000
EVPN VXLAN Aware Flexible NetFlow - Pv4 Ingress Cache Entries	128000	128000	128000	64000	98000	32000	32000	64000

Product Specification								
Switch Model	C9500-120, C9500-240, C9500-16X, C9500-40X				C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C			
EVPN VXLAN Aware Flexible NetFlow - IPv4 Egress Cache Entries	128000	128000	128000	64000	32000	32000	32000	64000
EVPN VXLAN Aware Flexible NetFlow - IPv6 Ingress Cache Entries	128000	128000	128000	64000	32000	32000	32000	64000
EVPN VXLAN Aware Flexible NetFlow - IPv6 Egress Cache Entries	128000	128000	128000	64000	32000	32000	32000	64000
Layer 2 VNI (L2VNI) Multicast Replication BUM Rate-Limiter	512	512	512	512	512	512	512	512
MicroSegmentation - Community VLAN to L2VNI	2000	2000	2000	2000	2000	2000	2000	2000
NanoSegmentation - Isolated VLAN to L2VNI	384	384	384	384	384	384	384	384
Wide Area Bonjour (mDNS) over VXLAN Service Instance Count	15000	15000	15000	15000	15000	15000	15000	15000
BGP EVPN / VXLAN -- Spine Scale								
SDM Template	Distribution	Core	NAT	Custom	Distribution	Core	NAT	Custom
BGP IPv4 Peer Scale	250	250	250	250	1000	1000	1000	1000
BGP IPv6 Peer Scale	250	250	250	250	1000	1000	1000	1000
BGP L2VPN EVPN Peer Scale	250	250	250	250	1000	1000	1000	1000

Product Specification								
Switch Model	C9500-12Q, C9500-24Q, C9500-16X, C9500-40X				C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C			
Overlay IPv4 routes (LPM or Indirect Routes)	64000	64000	64000	256000	114000	212000	212000	256000
Overlay IPv4 Host routes	48000	32000	48000	256000	114000	212000	212000	256000
Overlay IPv6 routes (LPM or Indirect Routes)	64000	64000	64000	256000	114000	212000	212000	256000
Overlay IPv6 Host routes	48000	32000	48000	256000	114000	212000	212000	256000
BGP EVPN / VXLAN -- Border Scale								
SDM Template	Distribution	Core	NAT	Custom	Distribution	Core	NAT	Custom
EVPN to Layer 2 Handoff: IEEE 802.1Q	512	512	512	512	512	512	512	512
EVPN to Layer 2 Handoff: IEEE 802.1ad (QinQ)	512	512	512	512	512	512	512	512
EVPN to VRF Handoff: IP VRF (IPv4 and IPv6)	1000	1000	1000	1000	1000	1000	1000	1000
EVPN to MPLS Layer 3 VRF Unicast Handoff: VPNv4	256	256	256	256	512	512	512	512
EVPN to MPLS Layer 3 VRF Unicast Handoff: VPNv6	256	256	256	256	512	512	512	512
EVPN to MPLS Layer 3 VRF Multicast Handoff: mVPNv4	256	256	256	256	512	512	512	512
EVPN to MPLS Layer 3 VRF Multicast Handoff: mVPNv6	512	512	512	512	512	512	512	512

Product Specification								
Switch Model	C9500-120, C9500-240, C9500-16X, C9500-40X				C9500-32C, C9500-32QC, C9500-48Y4C, C9500-24Y4C			
EVPN to VPLS Layer 2 Handoff: Virtual Forwarding Instances (VFIs)	512	512	512	512	512	512	512	512
EVPN to VPLS Layer 2 Handoff: Neighbors Per VFI	128	128	128	128	128	128	128	128
EVPN to VPLS Layer 2 Handoff: Pseudowire	512	512	512	512	512	512	512	512



CHAPTER 13

Troubleshooting BGP EVPN VXLAN

- [Troubleshooting Scenarios for BGP EVPN VXLAN, on page 557](#)
- [Troubleshooting Broadcast, Unknown Unicast, Multicast Traffic Forwarding, on page 558](#)
- [Troubleshooting Unicast Forwarding Between VTEPs in the Same VLAN Through a Layer 2 VNI, on page 562](#)
- [Troubleshooting Unicast Forwarding Between VTEPs in Different VLANs Through a Layer 3 VNI, on page 574](#)
- [Troubleshooting Unicast Forwarding Between a VXLAN Network and an IP Network, on page 587](#)
- [Troubleshooting Tenant Routed Multicast, on page 590](#)

Troubleshooting Scenarios for BGP EVPN VXLAN

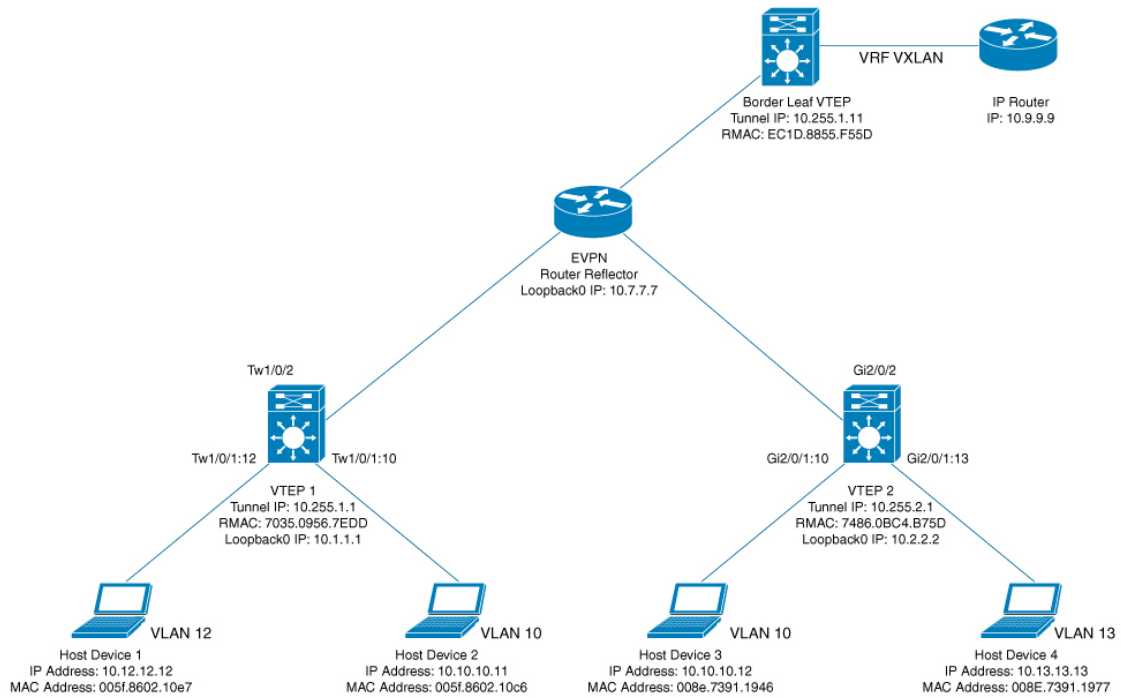
This document provides information about the various troubleshooting scenarios that are applicable to BGP EVPN VXLAN and how to troubleshoot each scenario.

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 48: EVPN VXLAN Topology



The following are the various troubleshooting scenarios that apply to BGP EVPN VXLAN for the topology illustrated in the [Figure 48: EVPN VXLAN Topology](#) above:

- **Scenario 1:** Troubleshooting Broadcast, Unkown 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, Unkown Unicast, Multicast Traffic Forwarding

This scenario might occur when host device 2 attempts to learn the ARP for host device 3 in [Figure 48: EVPN VXLAN Topology, on page 558](#). Perform the checks listed in the following table before troubleshooting BUM traffic forwarding:

Table 51: 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"> • Check the host device. • Check the SVI configuration on the VTEP.
Has the remote MAC address been learned for unknown unicast traffic?	Run the show platform software fed switch active macTable vlan <i>vlan-id</i> 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 559](#)
2. [Set Up Embedded Capture Towards the Core-Facing Interface, on page 560](#)
3. [Ping the Remote Host Device, on page 560](#)
4. [Verify that an ARP Request Has Been Received and a Multicast Route Has Been Built, on page 560](#)
5. [Confirm the Presence of ARP Request Replies in Embedded Capture, on page 561](#)
6. [Verify that the Encapsulated ARP Request is Leaving in a Multicast Group to a VXLAN UDP Destination Port, on page 561](#)
7. [Verify that the ARP Reply from Core Interface is Encapsulated in Unicast to a VXLAN UDP Destination Port, on page 562](#)

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
v 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
v 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]
v Virtual eXtensible Local Area Network
  > Flags: 0x0000, VXLAN Network ID (VNI)
  Group Policy ID: 0
  VXLAN Network Identifier (VNI): 10001
  Reserved: 0
v 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
v 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
v 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]
v Virtual eXtensible Local Area Network
  > Flags: 0x0000, VXLAN Network ID (VNI)
  Group Policy ID: 0
  VXLAN Network Identifier (VNI): 10001
  Reserved: 0
v 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 52: 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 arp -a 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"> • Check the host device. • Check the SVI configuration 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"> • show run section l2vpn • show run section vlan config • show run interface nve interface-number
Has the remote MAC address been learned in platform MATM in the same VLAN as the local host?	Run the show platform software fed switch active matm macTable vlan vlan-id 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 563](#)
2. [Ensure that an NVE Peer is Present for the Layer 2 VNI, on page 565](#)
3. [Verify the Provisioning of the Layer 2 VNI in NVE Component, on page 565](#)
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 566](#)

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 00:47:02
```

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

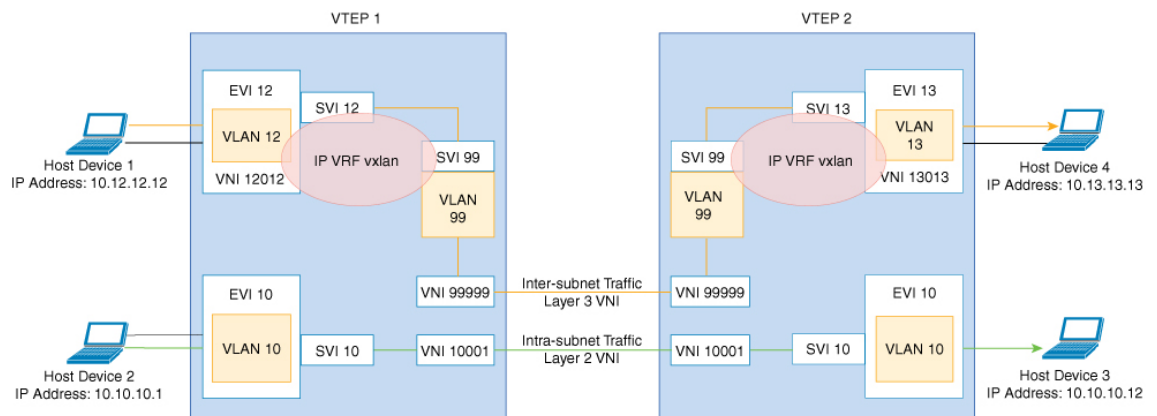
Unicast Address table information :
005f.8602.10c6 VXLAN_CP L:10001:10.255.2.1 R:10001:10.255.1.1

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 49: 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 568](#)
2. [Verify that Both Local and Remote MAC Addresses are Learned in FED-MATM, on page 568](#)

3. Confirm that the ICMP Echo Request Leaves VTEP 1 Encapsulated and Goes to a UDP Destination Port on VTEP 2, on page 569
4. Verify ARP for Local Host Devices, on page 569
5. Verify that the MAC Address Entries are Learned in SISF Device Tracking Table, on page 569
6. Verify that EVPN Manager Has Been Updated with the MAC Address Entries, on page 570
7. Verify that EVPN Manager Has Updated the MAC Routes into Layer 2 RIB, on page 571
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 571
9. Verify that the MAC Routes Learned from BGP and Updated to Layer 2 RIB are Also Updated to L2FIB, on page 573



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      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

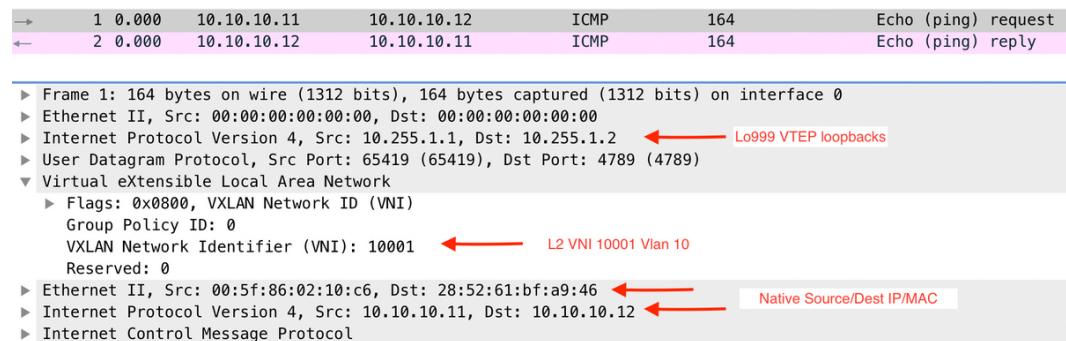
<<- Local MAC address is displayed here

```

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 50:



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 l2vpn 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 l2vpn 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 l2vpn 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:    0000.0000.0000.0000.0000
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:    0000.0000.0000.0000.0000
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 l2fib 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 53: 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 show ip interface brief exclude unassigned 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"> • show run section l2vpn • show run section vlan config • show run interface nve interface-number

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 575](#)
2. [Verify that the IP VRF is Provisioned with the Correct SVIs, Stitching Route-Targets, and Route Distinguisher, on page 576](#)
3. [Verify that Both Layer 2 and Layer 3 VNIs are provisioned in the VRF and are UP, on page 577](#)
4. [Verify that EVPN Manager is Updated from the NVE with all the Layer 2 and IRB Attributes, on page 578](#)
5. [Verify that the Remote Layer 3 VNI Details are Learned on Each VTEP, on page 579](#)
6. [Verify that the Layer 3 VNI Tunnel Pseudoport is Installed into Layer 2 FIB in the Core VLAN, on page 579](#)

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 Pseudoport
-----
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 Pseudoport
-----
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_RTs 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_RTs 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

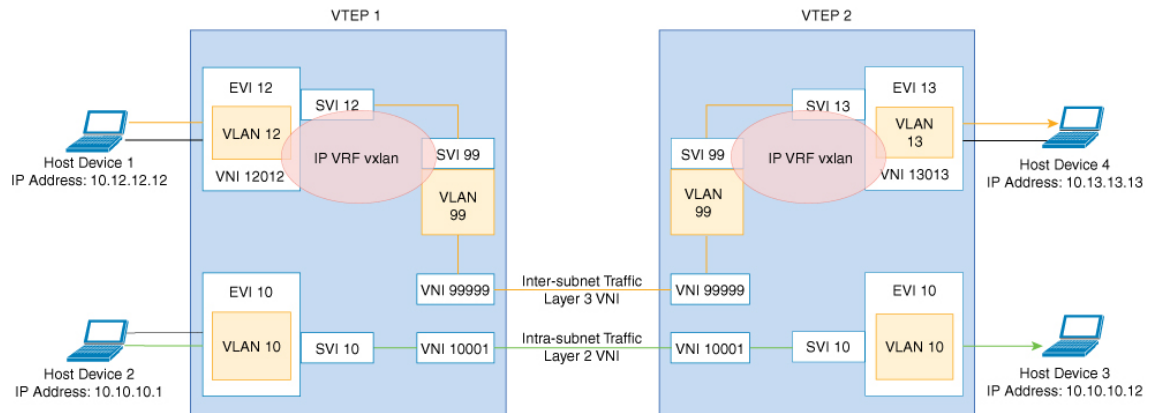
IP Multicast Prefix table information :
Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 5111, Ports: 0
Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 5111, Ports: 0
Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 5111, Ports: 0

```

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 51: 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 581
2. Verify that MAC Address and IP Address Entries are Learned in EVPN Manager, on page 582
3. Verify that MAC Address and IP Address Entries are Learned in Layer 2 RIB, on page 583
4. Verify that Local MAC Address and IP Address Entries are Learned in MAC VRF, on page 583
5. Verify that Remote MAC-IP Address Pair is Learend in the VRF, on page 584
6. Verify that IP Routes are Inserted in RIB, on page 585
7. Verify that the Adjacency Table Contains Entries for the VRF-Enabled Core VLAN Interface, on page 585
8. Confirm that Adjacency Exists to the VTEP Tunnel IP Address for a Host Device in IP VRF, on page 586
9. Confirm that Adjacency Exists to Reach Tunnel Destination, on page 586
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 586

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 54: 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>Note 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?	Run the show bgp l2vpn evpn all command in privileged EXEC mode on the VTEP. Look for remote prefix to be displayed as [5] for route type 5.

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 575](#) 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 587](#)
2. [Check the Table Entries for BGP, EVPN, and VPNv4 Tables, on page 587](#)
3. [Confirm that Adjacency exists to Reach Tunnel Destination, on page 590](#)

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  ports  siHandle
      riHandle                             diHandle      *a_time *e_time
-----
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
103
99    ec1d.8b55.f55d                         0x1000001 0      0      64  0x7ffa48d065e8  0x7ffa48d01d08
      0x7ffa48c9a618                       0x7ffa4838cc18 0      0
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
```

Troubleshooting Tenant Routed Multicast

See [Troubleshoot EVPN VxLAN TRM on Catalyst 9000 Switches](#) document to learn how to troubleshoot issues with TRM in a BGP EVPN VXLAN fabric.



PART I

Integration with Cisco DNA Service for Bonjour

- [Cisco DNA Service for Bonjour Solution Overview, on page 593](#)
- [Configuring Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks, on page 605](#)
- [Configuring VRF-Aware Local Area Bonjour Services, on page 627](#)



CHAPTER 14

Cisco DNA Service for Bonjour Solution Overview

- [About Cisco DNA Service for Bonjour Solution, on page 593](#)
- [Solution Components, on page 594](#)
- [Supported Platforms, on page 595](#)
- [Supported Network Design, on page 596](#)

About Cisco DNA Service for Bonjour Solution

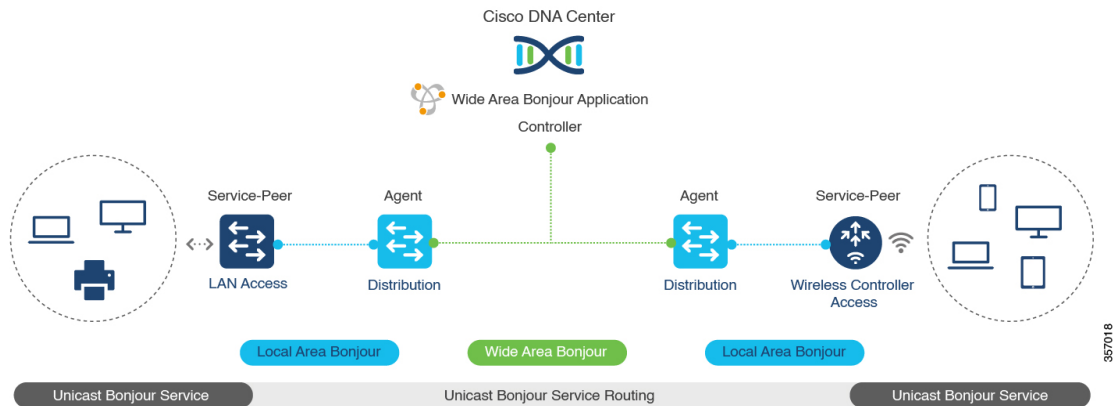
The Apple Bonjour protocol is a zero-configuration solution that simplifies network configuration and enables communication between connected devices, services, and applications. Using Bonjour, you can discover and use shared services with minimal intervention and configuration. Bonjour is designed for single Layer 2 domains that are ideal for small, flat, single-domain setups, such as home networks. The Cisco DNA Service for Bonjour solution eliminates the single Layer 2 domain constraint and expands the scope to enterprise-grade traditional wired and wireless networks, including overlay networks such as Cisco Software-Defined Access (SD-Access) and industry-standard BGP EVPN with VXLAN. The Cisco Catalyst 9000 series LAN switches and wireless LAN controllers follow the industry standard, RFC 6762-based multicast DNS (mDNS) specification to support interoperability with various compatible wired and wireless consumer products in enterprise networks.

The Cisco Wide Area Bonjour application is a software-defined, controller-based solution that enables devices to advertise and discover Bonjour services across Layer 2 domains, making these services applicable to a wide variety of wired and wireless enterprise networks. The Cisco Wide Area Bonjour application also addresses problems relating to security, policy enforcement, and services administration on a large scale. The new distributed architecture is designed to eliminate mDNS flood boundaries and transition to unicast-based service routing, providing policy enforcement points and enabling the management of Bonjour services. With the Cisco Wide Area Bonjour application, you can seamlessly introduce new services into the existing enterprise environment without modifying the existing network design or configuration.

The enhanced intuitive GUI provides you with centralized access control and monitoring capabilities, combined with the scalability and performance required for large-scale Bonjour services deployments for various supporting enterprise network types.

The following figure illustrates how the Cisco Wide Area Bonjour application operates across two integrated domain networks with end-to-end unicast-based service routing.

Figure 52: Cisco Wide Area Bonjour Solution



- Local-Area Service Discovery Gateway Domain - Multicast DNS Mode:** The classic Layer 2 multicast flood-n-learn-based deployment model. The service provider and receiver can discover and browse within the common VLAN or broadcast domain without any security and location-based policy enforcement. The Cisco Catalyst switches at the Layer 3 boundary function as the Service Discovery Gateway (SDG) to discover and distribute services between local wired or wireless VLANs based on applied policies. The inter-VLAN service routing at a single gateway is known as Local Area Bonjour.
- Local Area Service Discovery Gateway Domain - Unicast Mode:** The new enhanced Layer 2 unicast policy-based deployment model. The new mDNS service discovery and distribution using Layer 2 unicast address enables flood-free LAN and wireless networks. Cisco Catalyst switches and Cisco Catalyst 9800 series wireless LAN controllers in Layer 2 mode introduce a new service-peer role, replacing classic flood-n-learn, for new unicast-based service routing support in the network. The service-peer switch and wireless LAN controller also replace mDNS flood-n-learn with unicast-based communication with any RFC 6762 mDNS-compatible wired and wireless endpoints.
- Wide-Area Service Discovery Gateway Domain:** The Wide Area Bonjour domain is a controller-based solution. The Bonjour gateway role and responsibilities of Cisco Catalyst switches are extended from a single SDG switch to an SDG agent, enabling Wide Area Bonjour service routing beyond a single IP gateway. The network-wide distributed SDG agent devices establish a lightweight, stateful, and reliable communication channel with a centralized Cisco DNA Center controller running the Cisco Wide Area Bonjour application. Service routing between the SDG agents and the controller operates over regular IP networks using TCP port 9991. The SDG agents route locally discovered services based on the export policy.

Solution Components

The Cisco DNA Service for Bonjour solution is an end-to-end solution that includes the following key components and system roles to enable unicast-based service routing across the local area and Wide Area Bonjour domain:

- Cisco Service peer:** A Cisco Catalyst switch and Catalyst Wireless LAN Controller (WLC) in Layer 2 access function in service peer mode to support unicast-based communication with local attached endpoints and export service information to the upstream Cisco SDG agent in the distribution layer.

- **Cisco SDG agent:** A Cisco Catalyst switch functions as an SDG agent and communicates with the Bonjour service endpoints in Layer 3 access mode. At the distribution layer, the SDG agent aggregates information from the downstream Cisco service peer switch and WLC, and exports information to the central Cisco DNA controller.
- **Cisco DNA controller:** The Cisco DNA controller builds the Wide Area Bonjour domain with network-wide and distributed trusted SDG agents using a secure communication channel for centralized services management and controlled service routing.
- **Endpoints:** A Bonjour endpoint is any device that advertises or queries Bonjour services conforming to RFC 6762. The Bonjour endpoints can be in either LANs or WLANs. The Cisco Wide Area Bonjour application is designed to integrate with RFC 6762-compliant Bonjour services, including AirPlay, Google Chrome cast, AirPrint, and so on.

Supported Platforms

The following table lists the supported controllers, along with the supported hardware and software versions.

Table 55: Supported Controllers with Supported Hardware and Software Versions

Supported Controller	Hardware	Software Version
Cisco DNA Center appliance	DN2-HW-APL DN2-HW-APL-L DN2-HW-APL-XL	Cisco DNA Center, Release 2.2.3
Cisco Wide Area Bonjour application	—	2.4.264.12003

The following table lists the supported SDG agents along with their license and software requirements.

Table 56: Supported SDG Agents with Supported License and Software Requirements

Supported Platform	Supported Role	Local Area SDG	Wide Area SDG	Minimum Software
Cisco Catalyst 9200 Series Switches	SDG	Cisco DNA Advantage	Unsupported	Cisco IOS XE Bengaluru 17.6.1
Cisco Catalyst 9200L Series Switches	—	Unsupported	Unsupported	—
Cisco Catalyst 9300 Series Switches	Service peer SDG agent	Cisco DNA Advantage	Cisco DNA Advantage	Cisco IOS XE Bengaluru 17.6.1
Cisco Catalyst 9400 Series Switches	Service peer SDG agent	Cisco DNA Advantage	Cisco DNA Advantage	Cisco IOS XE Bengaluru 17.6.1
Cisco Catalyst 9500 Series Switches	Service peer SDG agent	Cisco DNA Advantage	Cisco DNA Advantage	Cisco IOS XE Bengaluru 17.6.1

Supported Platform	Supported Role	Local Area SDG	Wide Area SDG	Minimum Software
Cisco Catalyst 9500 High Performance Series Switches	Service peer SDG agent	Cisco DNA Advantage	Cisco DNA Advantage	Cisco IOS XE Bengaluru 17.6.1
Cisco Catalyst 9600 Series Switches	Service peer SDG agent	Cisco DNA Advantage	Cisco DNA Advantage	Cisco IOS XE Bengaluru 17.6.1
Cisco Catalyst 9800 WLC	Service peer	Cisco DNA Advantage	Unsupported	Cisco IOS XE Bengaluru 17.6.1
Cisco Catalyst 9800-L WLC	Service peer	Cisco DNA Advantage	Unsupported	Cisco IOS XE Bengaluru 17.6.1

Supported Network Design

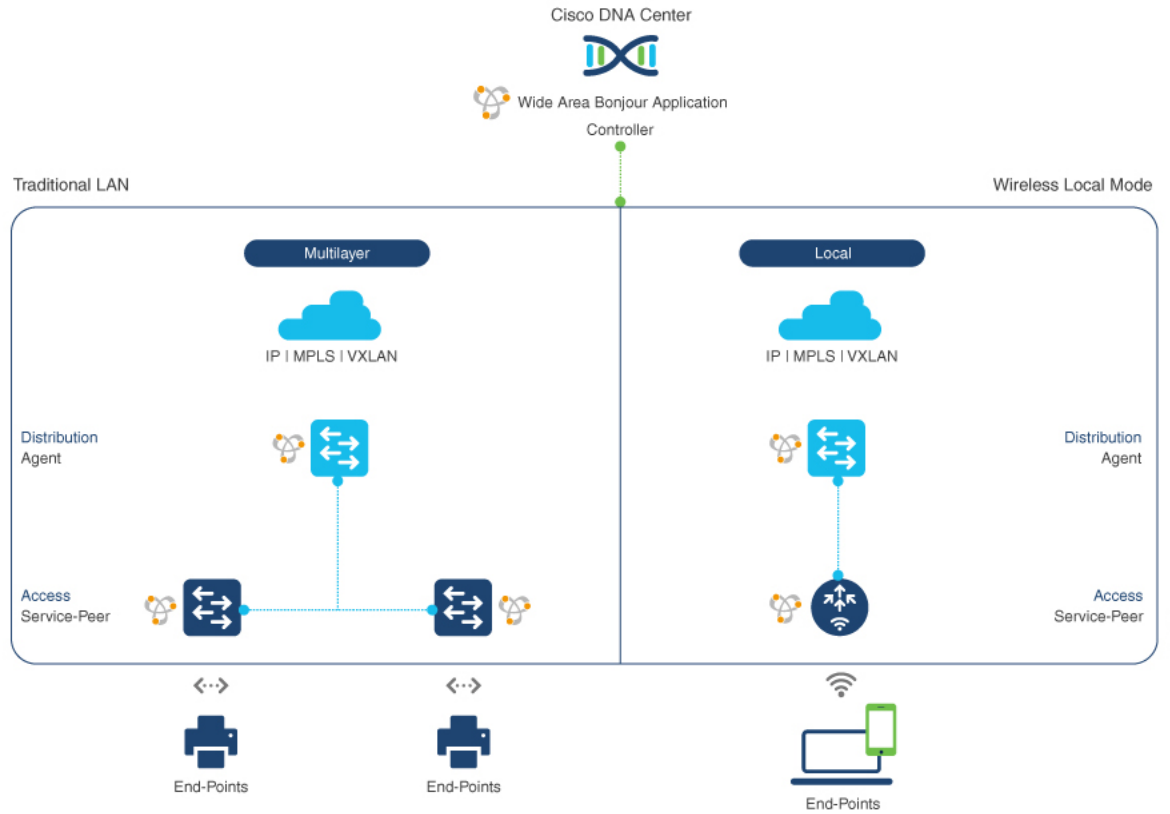
The Cisco DNA Service for Bonjour supports a broad range of enterprise-grade networks. The end-to-end unicast-based Bonjour service routing is supported on traditional, Cisco SD-Access, and BGP EVPN-enabled wired and wireless networks.

Traditional Wired and Wireless Networks

Traditional networks are classic wired and wireless modes deployed in enterprise networks. Cisco DNA Service for Bonjour supports a broad range of network designs to enable end-to-end service routing.

The following figure illustrates traditional LAN network designs that are commonly deployed in an enterprise.

Figure 53: Enterprise Traditional LAN Network Design

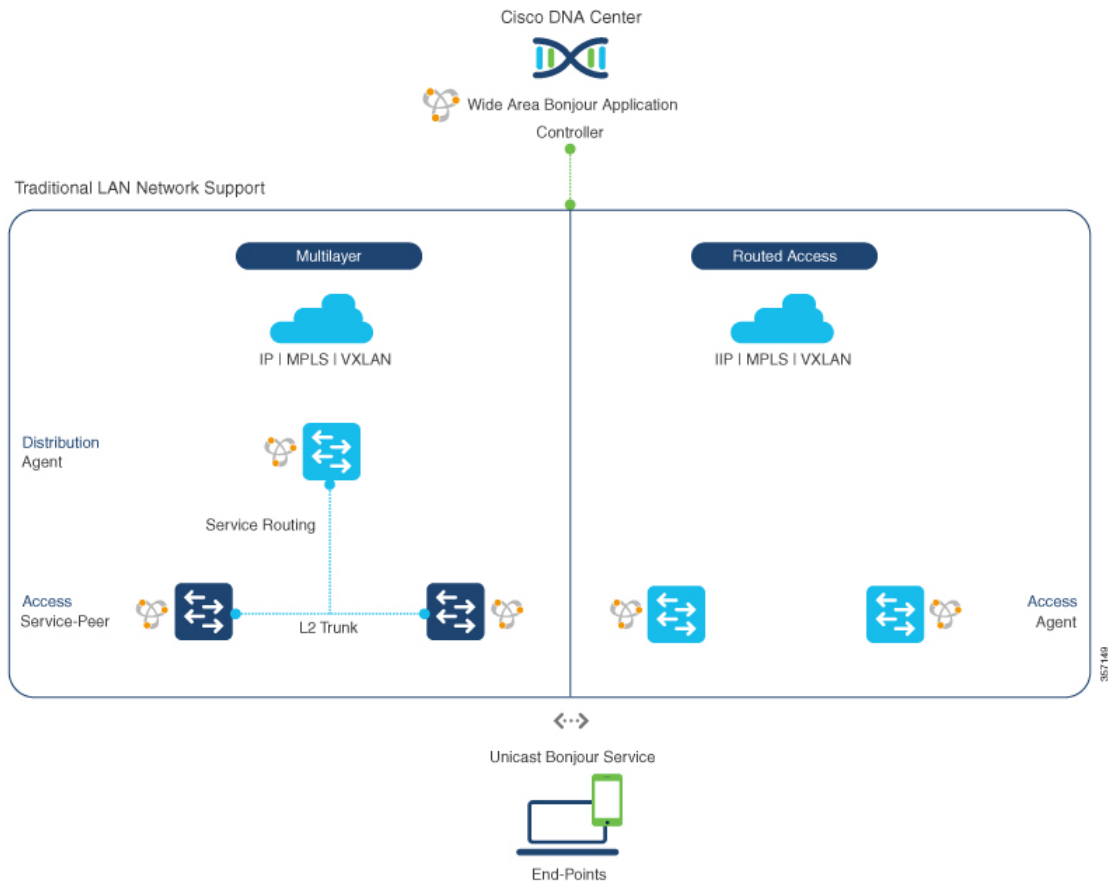


357153

Wired Networks

The following figure shows the supported LAN network designs that are commonly deployed in an enterprise.

Figure 54: Enterprise Multilayer and Routed Access Network Design



The SDG agent that provides Bonjour gateway functions is typically an IP gateway for wired endpoints that could reside in the distribution layer in multilayer network designs, or in the access layer in routed access network designs:

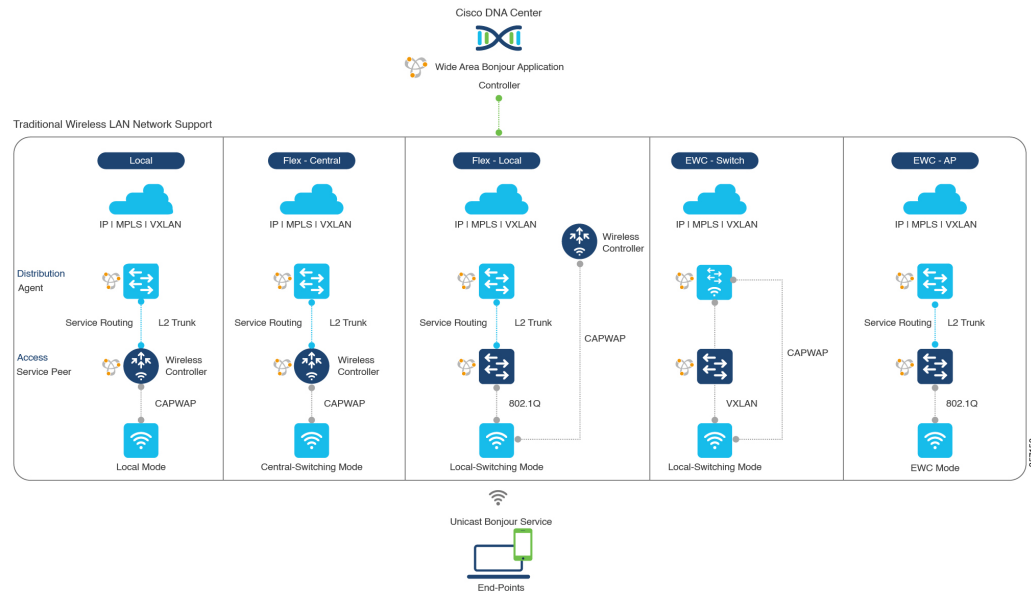
- **Multilayer LAN:** In this deployment mode, the Layer 2 access switch provides the first-hop Bonjour gateway function to locally attached wired endpoints. The Bonjour services and global discovery request are routed to the distribution layer systems that act as the IP gateway or SDG agent. There's no additional configuration or new requirement to modify the existing Layer 2 trunk settings between the access and distribution layers of the Cisco Catalyst switches. The policy-based service routing between the Layer 2 service-peer switches is performed by the SDG agent. The policy-based service routing between the SDG agents is performed by the Cisco DNA Center controller.
- **Routed Access:** In this deployment mode, the first-hop switch is an IP gateway boundary and, therefore, it must also perform the SDG agent role. The policy-based service routing between the SDG agents is performed by the Cisco DNA Center controller.

Wireless Networks

The Cisco DNA Service for Bonjour also supports various wireless LAN network designs that are commonly deployed in an enterprise. The Cisco Catalyst 9800 Series Wireless LAN Controller (WLC) can be deployed in a service-peer role supporting the mDNS gateway and paired with an upstream gateway switch for end-to-end service routing.

The following figure shows the supported wireless LAN network designs that are commonly deployed in an enterprise.

Figure 55: Enterprise Traditional Wireless LAN Network Design



The Cisco DNA Service for Bonjour supports the following modes for wireless LAN networks:

- Local Mode:** In this central switching wireless deployment mode, the Bonjour traffic is encapsulated within the CAPWAP tunnel from the Cisco access points to the centrally deployed Cisco Wireless LAN Controller. The Cisco access points are configured to be in local mode (central switching also allows the access point to be configured in FlexConnect mode). With central switching, the Cisco Catalyst 9800 Series Wireless LAN Controller provides the mDNS gateway function of Bonjour services in the service-peer role. The WLC can discover and distribute services to local wireless users and perform unicast service routing over a wireless management interface to the Cisco Catalyst switch in the distribution layer, which acts as the IP gateway and the SDG agent. There's no additional configuration or requirement to modify the existing Layer 2 trunk settings between the Cisco Wireless LAN Controller and the distribution layer of the Cisco Catalyst switch. The Cisco Wireless LAN Controller must be configured with Global Multicast and AP Multicast in Multicast mode. Unless the access point joins the wireless LAN controller-announced multicast group, communication to and from Bonjour endpoints is not enabled for the wireless user group.
- FlexConnect:** In FlexConnect local switching mode, both wired and wireless users share the same gateway in the access layer. The Layer 2 access switch provides the policy-based mDNS gateway function to locally attached wired and wireless users. The Cisco Catalyst switches in the distribution layer function as SDG agents for the LAN and wireless LAN user groups.
- Embedded Wireless Controller - Switch:** The Cisco Embedded Wireless Controller solution enables the lightweight integrated wireless LAN controller function within the Cisco Catalyst 9300 series switch. The Cisco Catalyst switches in the distribution layer function as SDG agents to the LAN and wireless LAN user groups. The SDG agent in the distribution layer provides unicast service routing across all wireless access point and Layer 2 service-peer switches without any mDNS flooding. The embedded Cisco Wireless LAN Controller switch must be configured with Global Multicast and AP Multicast in Multicast mode and mDNS must be set in bridging mode.

- **Embedded Wireless Controller - Access Point:** The Cisco Embedded Wireless Controller solution enables the lightweight integrated wireless LAN controller function within the Cisco access points configured in the primary role. The wireless users share the same Bonjour gateway in the access layer as the wired endpoints. The Cisco Catalyst switches in the access layer function as service peers to the LAN and wireless LAN user groups. The SDG agent in the distribution layer provides unicast service routing across all Layer 2 service-peer switches in the Layer 2 network block without any mDNS flooding. AP multicast is required for Embedded Wireless mode AP, and mDNS must be set in bridging mode.



Note The Cisco AireOS-based WLC can be deployed as an mDNS pass-through network device between the wireless endpoints. The upstream SDG agent provides consistent Bonjour gateway functions for wireless endpoints, as for wired networks. In general, the IP gateway of wireless clients is also a Bonjour gateway. However, the placement of the SDG agent may vary depending on the wireless LAN deployment mode.

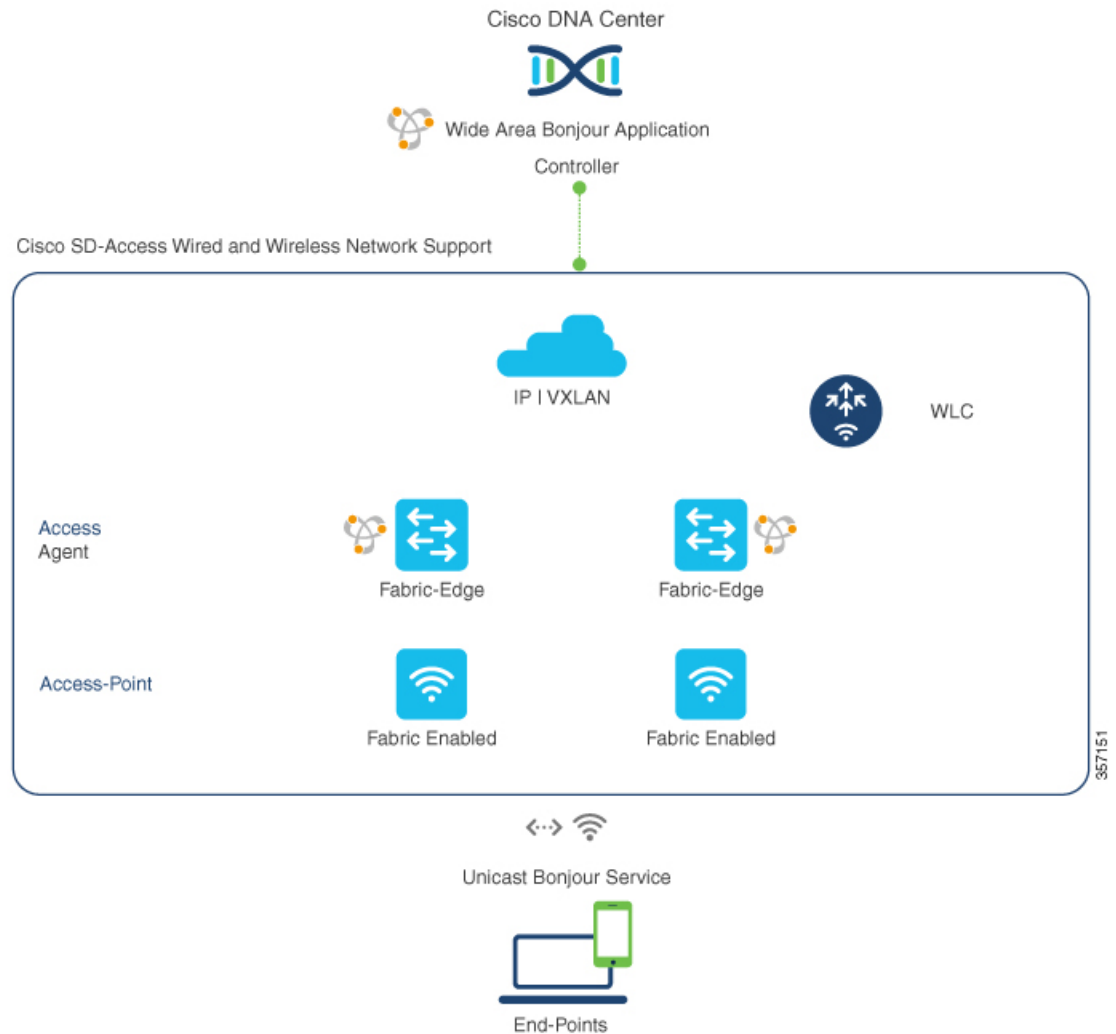
Cisco SD-Access Wired and Wireless Networks

Cisco SD-Access-enabled wired and wireless networks support Cisco DNA Service for Bonjour. From Cisco IOS-XE Release 17.4.1, the VRF-aware Wide Area Bonjour service routing provides secure and segmented mDNS service discovery and distribution management for fabric-enabled wired and wireless networks. The VRF-aware Wide Area Bonjour service routing eliminates the need for Layer 2 flooding. The Layer 3 Fabric Edge switch in the access layer must be configured as the SDG agent and paired with the central Cisco DNA Center for end-to-end service routing. Wide Area Bonjour policies must be aligned with the SD-Access network policies for virtual networks and SGT policies, if any.

Fabric-Enabled Wired and Wireless Networks

The following figure shows Cisco SD-Access-enabled wired and wireless networks without extending the Layer 2 network boundaries.

Figure 56: Cisco SD-Access Network Design



The Cisco DNA Service for Bonjour for SD-Access-enabled wired and wireless networks uses two logical components:

- **SDG agent:** The Layer 3 Fabric Edge switch in the access layer network is configured as the SDG agent. The VRF-aware mDNS gateway and Wide Area Bonjour service routing configuration is added only after SD-Access is configured.
- **Cisco DNA controller:** The Cisco Wide Area Bonjour application on Cisco DNA Center acts as the controller that supports policy and location-based service discovery and distribution between network-wide distributed Fabric Edge switches.

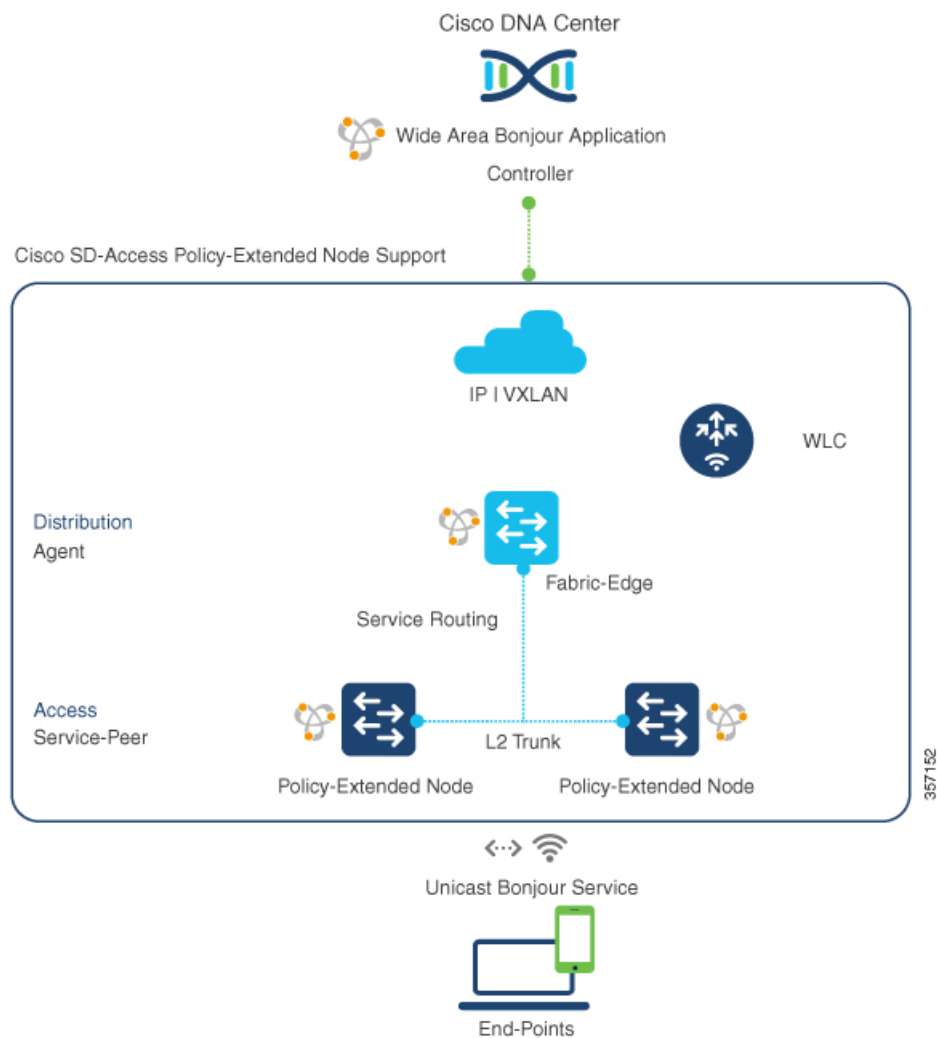
The Wide Area Bonjour communication between the SDG agent and the controller takes place through the network underlay. The SDG agent forwards the endpoint announcements or queries to the controller through the fabric underlay based on policies. After discovering a service, a Bonjour-enabled application establishes direct unicast communication between endpoints through the fabric overlay. This communication is subject to configured overlay IP routing and SGT policies, if any.

The Cisco Wireless LAN Controller must be configured with Global Multicast and AP Multicast in Multicast mode. The network administrator must enable IP Multicast in the underlay and ensure all fabric-enabled Cisco wireless access points have successfully joined the multicast group. The mDNS snooping configuration on the Cisco Wireless LAN Controller is ineffective and must remain in disabled mode.

Fabric-Enabled Policy Extended Node

The security policy can be extended to Cisco Catalyst 9000 Series Switches at Layer 2 access with the Policy Extended Node (PEN) function in a Cisco SD-Access fabric network. The network security and mDNS service policy can be combined at the Layer 2 access PEN switch in a service-peer role combined with Fabric Edge supporting SDG agent mode in Layer 2/3 distribution layer for Wide Area Bonjour service routing with Cisco DNA Center.

Figure 57: Fabric-Enabled Policy Extended Node

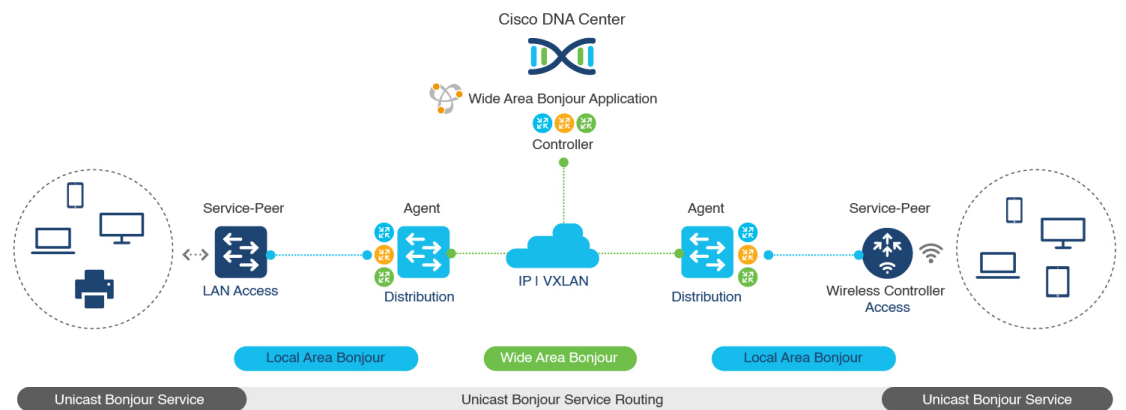


BGP EVPN Networks

The BGP EVPN-based enterprise network provides a flexible Layer 3 segmentation and Layer 2 extension overlay network. From Cisco IOS-XE Release 17.4.1, the VRF-aware Wide Area Bonjour service routing provides secure and segmented mDNS service discovery and distribution management for all common VXLAN overlay deployment models. The VRF-aware Wide Area Bonjour service routing eliminates mDNS flooding over Layer 2 extended EVPN VXLAN networks (symmetric and asymmetric integrated routing and bridging (IRB)) and service reachability challenges for Layer 3 segmented EVPN VXLAN networks in the fabric.

The following figure shows the BGP EVPN leaf switch in Layer 3 access mode, supporting overlay Bonjour service routing for a BGP EVPN-enabled wired and wireless enterprise network over various types of Layer 2 networks and Layer 3 segmented VRF-enabled networks.

Figure 58: Overlay Bonjour Service for a BGP EVPN-Enabled Enterprise Network

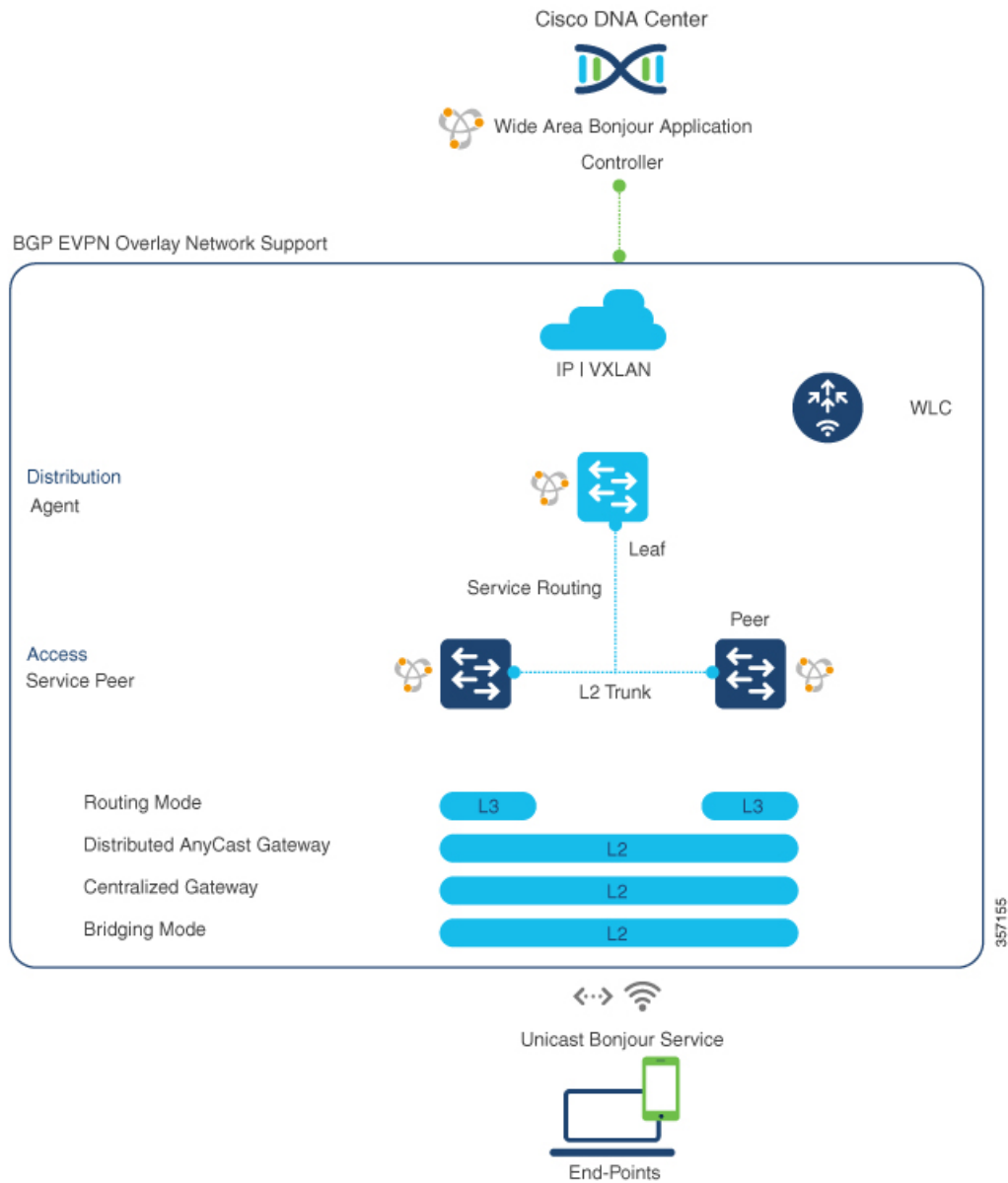


Cisco DNA Service for Bonjour supports Wide Area Bonjour service routing for BGP EVPN networks extended with Layer 2 service-peer network devices, such as a Cisco Catalyst switch or 9800 series WLC. The BGP EVPN leaf device in the distribution layer supports the SDG agent role for overlay service routing.

The Cisco DNA Service for Bonjour solution for BGP EVPN networks enables policy-based end-to-end service routing for virtual network environments. The solution helps to protect enterprise network scale and performance by eliminating the Layer 2 mDNS flood over the VXLAN across the IP core network.

The following figure shows mDNS endpoints connecting the Layer 2 access switch in service-peer mode to the upstream BGP EVPN leaf switch in the Layer 2/3 distribution layer supporting overlay Bonjour service routing for a BGP EVPN-enabled wired and wireless enterprise network over various types of Layer 2 networks and Layer 3 segmented VRF-enabled networks.

Figure 59: mDNS Endpoints Connecting an L2 Switch to an Upstream BGP EVPN Leaf Switch



For more information about BGP EVPN networks, see [Cisco DNA Service for Bonjour Configuration Guide, Cisco IOS XE Bengaluru 17.6.x \(Catalyst 9600 Switches\)](#).



CHAPTER 15

Configuring Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks

- [Prerequisites for Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks, on page 605](#)
- [Restrictions for Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks, on page 606](#)
- [Information About Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks, on page 606](#)
- [How to Configure Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks, on page 609](#)
- [Verifying Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks, on page 619](#)
- [Additional References for Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks, on page 626](#)

Prerequisites for Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks

This section provides the list of prerequisites for a Cisco Catalyst leaf switch that needs to be deployed in SDG Agent mode.:

- Ensure that you successfully configure and operate the BGP EVPN VXLAN overlay networks on the Cisco Catalyst devices before you configure Cisco Local Area and Wide Area Bonjour for LAN and WLAN networks.
- Verify that the targeted leaf switch is supported in SDG Agent and the Layer 2 access switch is supported in Service-Peer mode. See [Supported Platforms, on page 595](#) for more information.
- Verify that the targeted SDG Agent leaf switch, Service-Peer switch and Service-Peer wireless controller (WLC) run on the minimum required Cisco IOS XE software version.
- Ensure that the SDG Agent leaf switch, Service-Peer switch, and Service-Peer WLC run on a valid Cisco DNA Advantage license.
- In a multilayer network with Layer 2 unicast service-routing between an SDG Agent leaf switch in distribution layer and Service-Peer, ensure that the connection is through a Layer 2 trunk in static mode.
- Ensure that the Cisco DNA Center has IP connectivity with the SDG Agent leaf switch in either the underlay or overlay network.

- Ensure that the SDG Agent leaf switch has IP connectivity with the Service-Peer switch and Service-Peer WLC in the same IPv4 subnet when the traffic is globally routed through the management VLAN.
- Verify that wireless AP multicast is configured in the underlay network and Cisco Wireless APs have successfully joined the AP multicast group announced by the WLC.



Note The leaf switch can also enable unicast-based service-routing with a downstream Layer 2 access switch and Catalyst 9800 Series WLC.

Restrictions for Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks

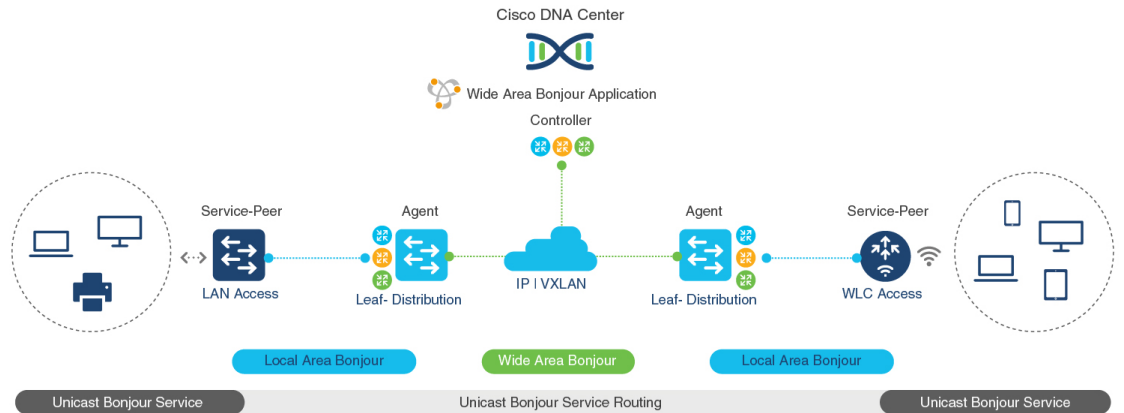
- Only Cisco Catalyst 9000 Series Layer 2 Ethernet switches that are connected to a distribution layer leaf switch can be deployed in Service-Peer mode. No other classic Cisco Catalyst LAN switches are supported in Service-Peer mode.
- Cisco Embedded WLC is not supported on Cisco Catalyst Series switches.
- Multicast DNS (mDNS) Trust port is not supported on the Layer 2 port channel interface of Cisco Catalyst Series switches.
- Local-area service-routing is not supported on the management port of Cisco Catalyst Series switches.

Information About Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks

Cisco Catalyst 9000 Series leaf switches or VTEPs introduce hierarchical service-routing for standard-based BGP EVPN VXLAN overlay networks. Leaf switches can function as Service Discovery Gateway Agents (SDG-Agents) to enable mDNS service-routing for Layer 3 and Layer 2 overlay networks. The enhanced gateway function (at the first hop of wired and wireless networks) communicates with directly associated, industry-standard, [RFC 6762](#)-compliant multicast DNS (mDNS) end points in Layer 2 unicast mode. Communication in unicast mode eliminates Layer 2 mDNS traffic flooding and enables service discovery across Layer 3 networks for large scale and enterprise-grade LAN and WLAN overlay networks. By doing this, unicast mode enhances security, bandwidth, scale, and performance for underlay and overlay fabric networks.

The following image illustrates a BGP EVPN VXLAN fabric that supports service-routing for wired and wireless user endpoints over Layer 3 overlay networks:

Figure 60: Cisco Wide Area Bonjour over BGP EVPN VXLAN Layer 3 Overlay Networks



A leaf switch in enterprise campus access or distribution layer can perform Wide Area Bonjour service-routing. Service-routing allows the leaf switch to establish stateful and reliable communication with a centralized Cisco DNA Center in the underlay network. The leaf can also discover and distribute policy-based services. Service data communication between mDNS sources and receivers over an EVPN VXLAN network uses segmented Layer 3 overlay forwarding topologies. A leaf switch can extend unicast-based service-routing through a downstream Layer 2 access device to build a Local Area Bonjour domain for the VXLAN fabric. The domain allows end-to-end, unicast-based service-routing for EVPN VXLAN Layer 3 overlay networks. The downstream Layer 2 access device in the fabric can be one of:

- A Cisco Catalyst 9000 Series Ethernet switch
- A Cisco Catalyst 9800 Series WLC

Understanding Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks

A BGP EVPN Layer 3 leaf switch provides an IP gateway to a Layer 3-segmented, VRF-enabled interface to attach wired and wireless endpoints locally. Each BGP EVPN leaf switch in an access or distribution layer network provides a demarcation point to the broadcast boundary of its local Layer 2 network. This demarcation prevents mDNS service discovery and distribution beyond the leaf switch in the access or distribution layer network.

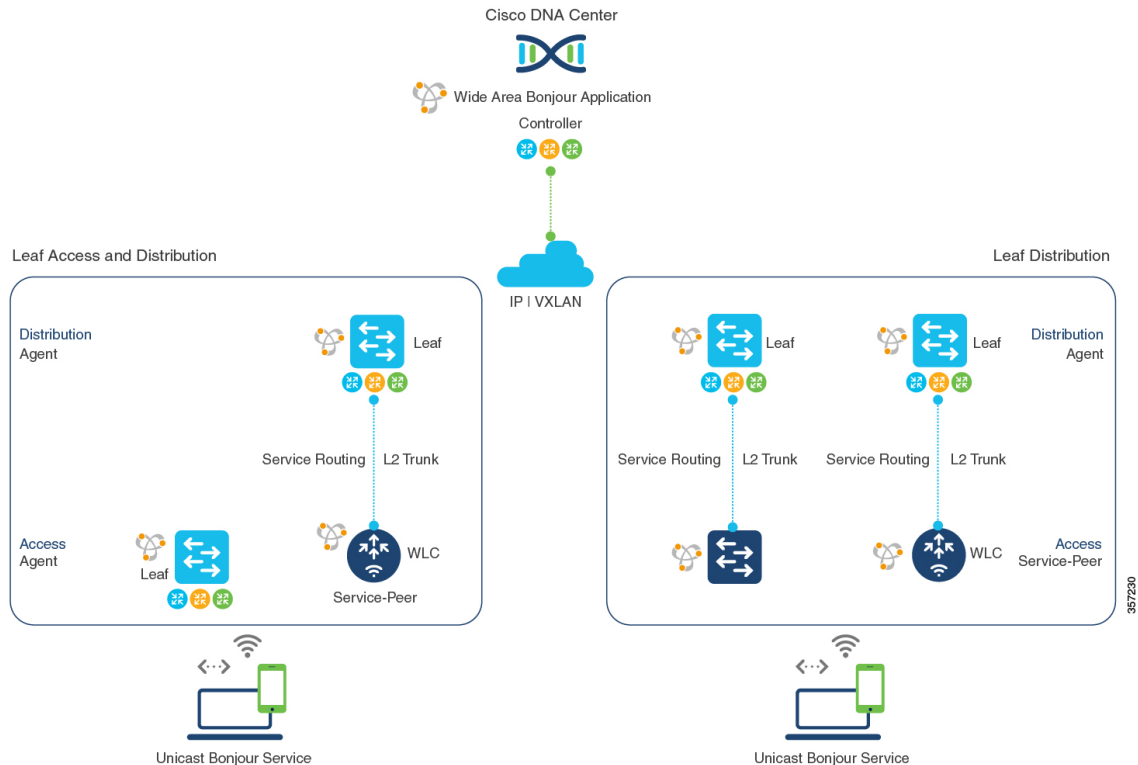
Cisco Wide Area Bonjour for BGP EVPN overlay networks enables unicast-based mDNS service discovery and distribution beyond the boundary of the single IP network. A network-wide distributed BGP EVPN leaf switch in an access or distribution layer network builds stateful and reliable communication with a Cisco DNA Center. This link enables policy- and location-based mDNS service-routing between the leaf switches for Layer 3 overlay networks.

The BGP EVPN fabric boundary can be initiated from the access or distribution layer network depending on network design. The following [Figure 61: Wide Area Bonjour for Wired and Wireless BGP EVPN Layer 3 Overlay Networks](#) provides two deployment alternatives to determine the handling of mDNS communication with downstream Layer 2 switches or WLCs and upstream fabric networks:

- **Leaf Access and Distribution:** The LAN access in Layer 3 mode provides the Layer 2–Layer 3 boundary to enable mDNS service-routing over the EVPN fabric network. The leaf switch in distribution layer can connect to a downstream Layer 2 WLC in Service-Peer mode and enable Unicast-based service-routing.

- **Leaf Distribution:** The LAN and Wireless distribution layer provides the Layer 2–Layer 3 boundary to enable mDNS service-routing over the EVPN fabric network. The leaf switch in Distribution layer can connect to a downstream Layer 2 LAN access switch and WLC in Service-Peer mode and enable Unicast-based service-routing.

Figure 61: Wide Area Bonjour for Wired and Wireless BGP EVPN Layer 3 Overlay Networks



mDNS service-routing for EVPN VXLAN Layer 3 overlay networks supports dynamic service discovery from a local network and performs service-routing with a centralized Cisco DNA-Center. The mDNS service-routing functions in the following ways with Bonjour domains:

- **Local Area Bonjour Domain:** A BGP EVPN Layer 3 leaf switch discovers and distributes mDNS services dynamically. The leaf switch distributes the mDNS services to directly or indirectly attached wired and wireless endpoints in a Layer 2 network. The leaf switch in distribution layer can establish unicast-based service-routing with a downstream Layer 2 access device to improve scale, performance, and location-based service capabilities. The downstream Layer 2 access device can be either a Cisco Catalyst Series switch or a Cisco 9800 Series WLC.

See [Configuring Local Area Bonjour Unicast Mode for Wired and Wireless Local Mode Networks](#) for detailed configuration steps.

- **Wide Area Bonjour Domain:** A BGP EVPN Layer 3 leaf switch and Wide Area Bonjour application in a Cisco DNA Center work together to build secure service-routing and peering. The link-up enables policy-based and location-based mDNS service-routing across Layer 3 overlay networks. The Layer 3 overlay network allows the exchange of VXLAN data only between the leaf switch and the Cisco DNA Center. The endpoints of mDNS service providers and receivers do not forward VXLAN data between themselves.

See [Configuring Wide Area Bonjour for BGP EVPN VXLAN Layer 3 Overlay Network](#) for detailed configuration steps for Cisco Catalyst 9000 Series switches.

See [Cisco Wide Area Bonjour Application on Cisco DNA Center User Configuration Guide, 2.1.2 Release](#) for more information about how to configure Cisco DNA Center.

How to Configure Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks

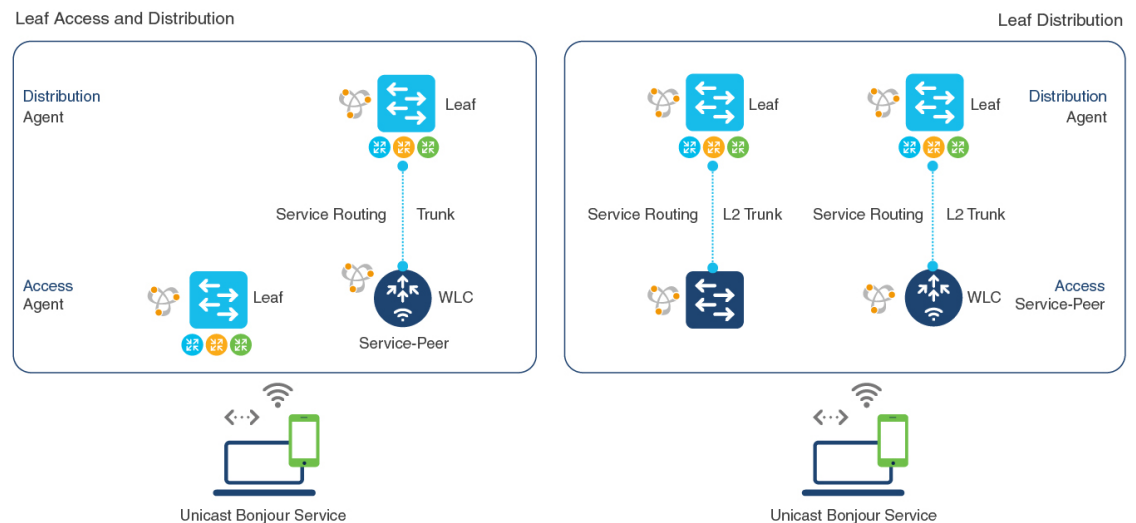
This section shows how to configure Local and Wide Area Bonjour over Layer 3 overlay networks.

Configuring Local Area Bonjour in Unicast Mode over EVPN VXLAN Layer 3 Overlay Networks

Local Area Bonjour enables unicast-based mDNS service-routing function between the service provider and receiver within the local Layer 2 network boundary. The leaf switch can be directly or indirectly attached to one of:

- Wired endpoint using an intermediate Layer 2 access switch
- Wireless endpoint using a Cisco Catalyst 9800 Series WLC

Figure 62: Local Area Bonjour Unicast Mode for Wired and Wireless Local Mode Networks



Service-routing for Local Area Bonjour enables unicast-based mDNS service discovery and distribution in the access layer through the BGP EVPN leaf switch. In the distribution layer, service-routing achieves this with multiple downstream Layer 2 Ethernet switches or Cisco WLCs in Service-Peer mode. To configure Local Area Bonjour in unicast mode over an EVPN VXLAN Layer 3 overlay network for the deployment shown above, perform the following operations:

- (Optional) Configure the Layer 2 access switch in Service-Peer mode to enable the first hop mDNS gateway that directly connects it to mDNS wired and wireless endpoints.
- (Optional) Configure the Cisco Catalyst 9800 series WLC in Service-Peer mode to enable the first hop mDNS gateway that directly connects it to mDNS wired and wireless endpoints.
- Configure the Layer 3 BGP EVPN access leaf switch as an SDG agent.
- Configure the Layer 3 BGP EVPN distribution leaf switch as an SDG agent.

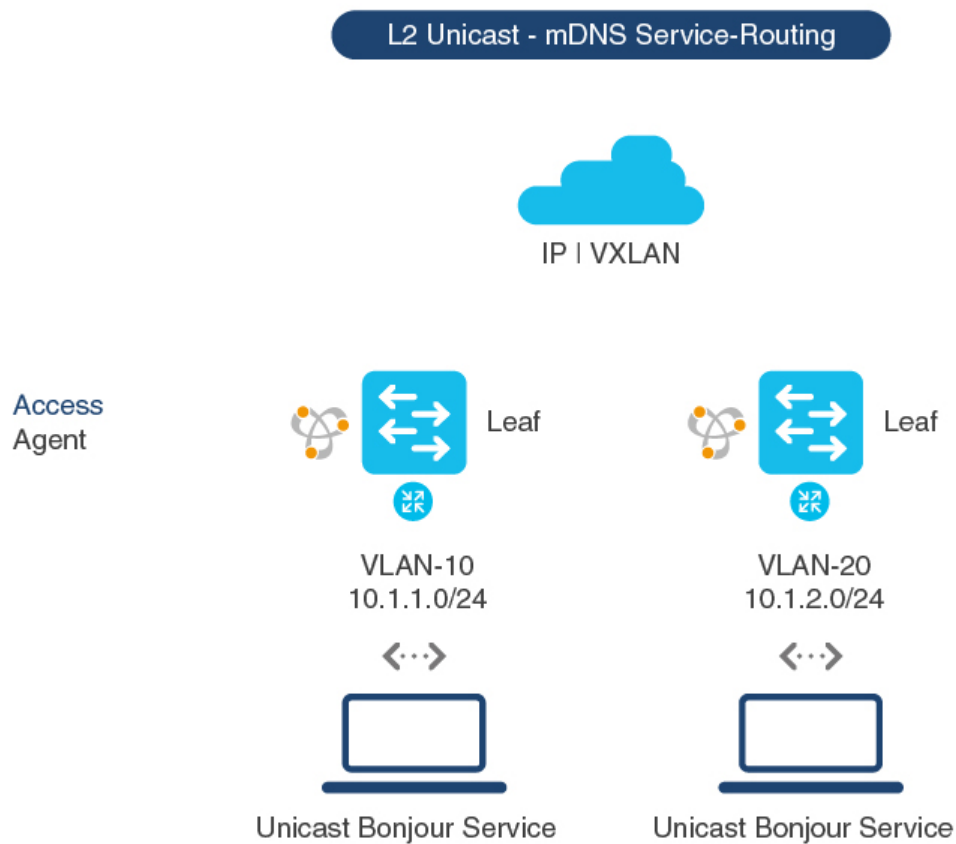


Note Ensure that you BGP EVPN VXLAN before you enable an mDNS gateway on a VLAN mapped to a Layer 3 overlay network. For detailed information and configuration steps, see *BGP EVPN VXLAN Configuration Guide* for the applicable release.

Configuring Layer 3 BGP EVPN Access Leaf SDG Agents

This sub-section shows a sample configuration to enable unicast-based service-routing and mDNS gateway functionality on a Layer 3 BGP EVPN access leaf switch for the topology in the figure below:

Figure 63: Layer 3 BGP EVPN Leaf Access SDG Agents



357232

Table 57: Configuring Layer 3 BGP EVPN Access Leaf SDG Agents

Configuration Step	Layer 3 Access Leaf SDG Agent in VLAN 10	Layer 3 Access Leaf SDG Agent in VLAN 20
Enable mDNS gateway and set the gateway mode on the Layer 3 Access switch.	! mdns-sd gateway mode sdg-agent !	mdns-sd gateway mode sdg-agent !
Create a unique mDNS inbound policy to permit ingress AirPrint service announcement from the service provider.	! mdns-sd service-list LOCAL-AREA-SERVICES-IN in match printer-ipp !	! mdns-sd service-list LOCAL-AREA-SERVICES-IN in match printer-ipp !
Create a unique mDNS outbound policy to permit egress AirPrint service response to the service receiver. Associate location-filter to share AirPrint service information from the grouped VLAN	! mdns-sd service-list LOCAL-AREA-SERVICES-OUT out match printer-ipp !	! mdns-sd service-list LOCAL-AREA-SERVICES-OUT out match printer-ipp !
Associate the inbound and outbound service lists to a unique service-policy.	! mdns-sd service-policy LOCAL-AREA-POLICY service-list LOCAL-AREA-SERVICES-IN service-list LOCAL-AREA-SERVICES-OUT !	! mdns-sd service-policy LOCAL-AREA-POLICY service-list LOCAL-AREA-SERVICES-IN service-list LOCAL-AREA-SERVICES-OUT !
Enable unicast-based mDNS gateway for wired users in VLAN 10 and VLAN 20 that are mapped to the Layer 3 VRF for BGP EVPN VXLAN. Configure the service-policy with advanced parameters and associate it with the VLAN.	! ! mDNS Unicast based gateway ! vlan configuration 10 mdns-sd gateway service-policy LOCAL-AREA-POLICY active-query timer 3600 !	! ! mDNS Unicast based gateway ! vlan configuration 20 mdns-sd gateway service-policy LOCAL-AREA-POLICY active-query timer 3600 !
Configure the wired users for VLAN 10 and VLAN 20 with IP address, Layer 3 VRF, and any other required parameters.	! interface Vlan 10 description BLUE VRF WIRED USER VLAN vrf forwarding BLUE_VRF ip address 10.1.1.254 255.255.255.0 no shutdown !	! interface Vlan 20 description BLUE VRF WIRED USER VLAN vrf forwarding BLUE_VRF ip address 10.1.2.254 255.255.255.0 no shutdown !

Configuring the BGP EVPN Leaf Switch and the Layer 2 Access Switch for Service-Routing

This sub-section shows a sample configuration to enable unicast-based service-routing between a Layer 2 access switch and a BGP EVPN VXLAN distribution leaf switch for the topology in the figure below:

Figure 64: Layer 2 Access Switch Service-Peer and Distribution Leaf Switch SDG Agent

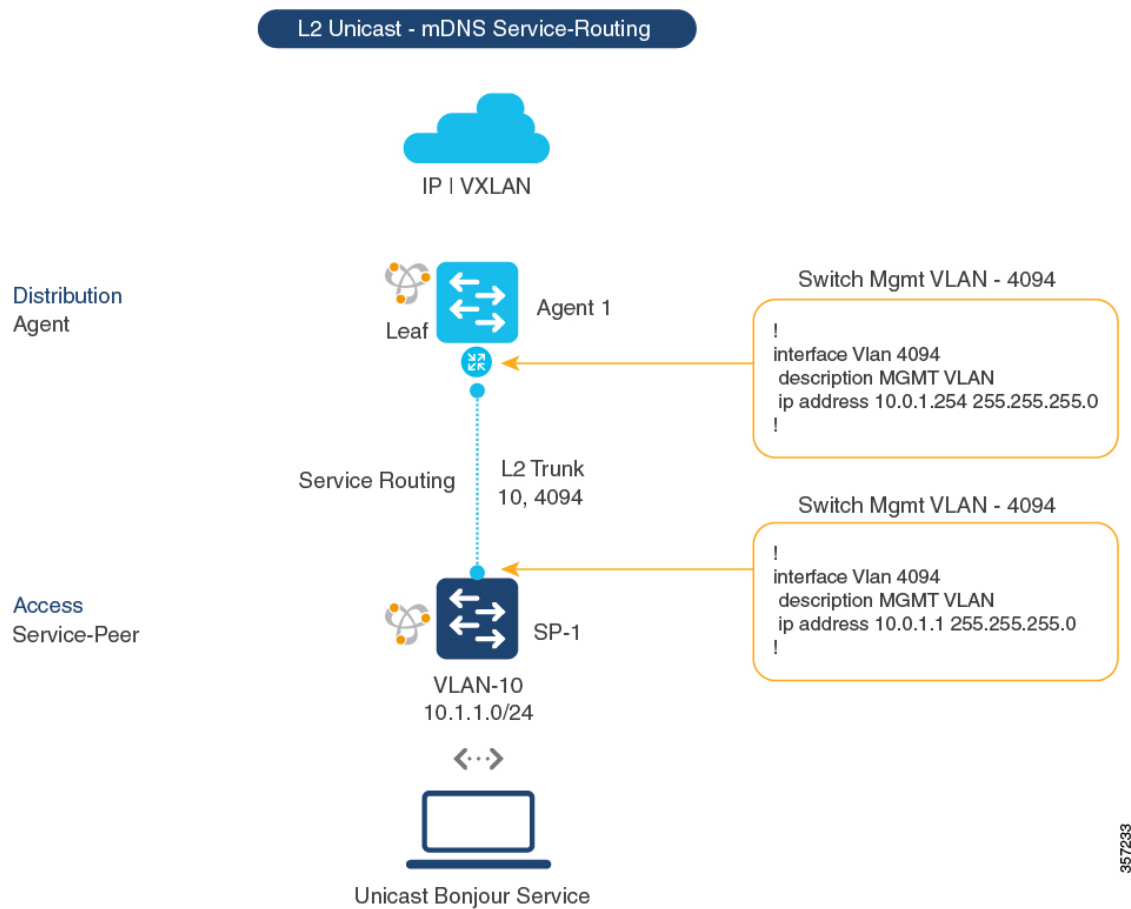


Table 58: Configuring the BGP EVPN Leaf Switch and the Layer 2 Access Switch for Service-Routing

Configuration Step	Layer 2 Access Switch as Service Peer	Distribution Leaf as SDG Agent
Step 1: Enable mDNS gateway and the gateway modes on the access switch (SP-1) and distribution switch (Agent-1).	<pre>! mdns-sd gateway mode service-peer !</pre>	<pre>! mdns-sd gateway mode sdg-agent !</pre>
Create a unique mDNS inbound policy to permit ingress AirPrint service announcement from the service provider on the access switch (SP-1).	<pre>! mdns-sd service-list LOCAL-AREA-SERVICES-IN in match printer-ipp !</pre>	<pre>!</pre>

Configuration Step	Layer 2 Access Switch as Service Peer	Distribution Leaf as SDG Agent
Step 3: Create a unique mDNS outbound policy to permit egress AirPrint service response to the locally attached wired service receiver.	<pre>! mdns-sd service-list LOCAL-AREA-SERVICES-OUT out match printer-ipp !</pre>	!
Step 4: Associate the inbound and outbound service lists to a unique service-policy.	<pre>! mdns-sd service-policy LOCAL-AREA-POLICY service-list LOCAL-AREA-SERVICES-IN service-list LOCAL-AREA-SERVICES-OUT !</pre>	!
<p>Step 5: Enable unicast-based mDNS gateway on wired users in VLAN 10 mapped to the Layer 3 VRF for BGP EVPN VXLAN.</p> <p>Associate the service-policy with advanced parameters.</p> <p>On the Service-Peer, configure the network IP address for the SDG-Agent's management VLAN and the parameters for the local source management VLAN to enable unicast service-routing over Layer 2 trunk interface.</p>	<pre>! ! mDNS Unicast based gateway ! vlan configuration 10 mdns-sd gateway service-policy LOCAL-AREA-POLICY active-query timer 3600 sdg-agent 10.0.1.254 source-interface Vlan 4094 !</pre>	<pre>! mDNS Unicast based gateway ! vlan configuration 10 mdns-sd gateway !</pre>
Step 6: Configure the wired user in VLAN 10 with the IP address, Layer 3 VRF, and any other parameters, as required.	!	<pre>! interface Vlan 10 description BLUE VRF WIRED USER VLAN vrf forwarding BLUE_VRF ip address 10.1.1.254 255.255.255.0 no shutdown !</pre>

Configuring the Layer 2 Cisco WLC and the BGP EVPN Leaf Switch for Service-Routing

This sub-section shows a sample configuration to enable unicast-based service-routing between a Cisco Catalyst 9800 WLC and a BGP EVPN VXLAN distribution leaf switch for the topology in the figure below:

Figure 65: Catalyst 9800 WLC Service-Peer and Distribution Leaf Switch SDG Agent

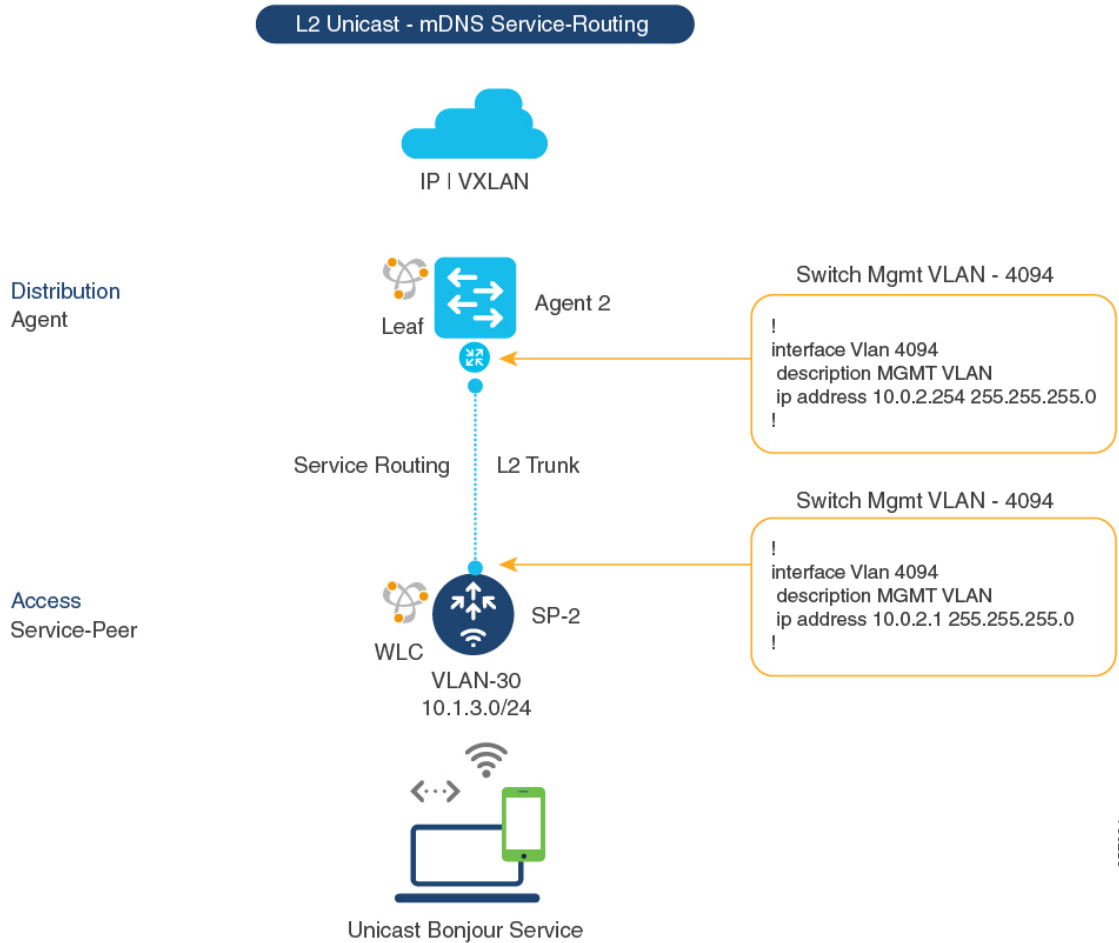


Table 59: Configuring the Layer 2 Cisco WLC and the BGP EVPN Leaf Switch for Service-Routing

Configuration Step	WLC Service-Peer	Layer 3 Distribution Leaf Switch SDG Agent
<p>Step 1: Enable mDNS gateway and set the gateway mode.</p> <ul style="list-style-type: none"> WLC Service Peer: Service Peer mode is the default mode with mDNS gateway configuration. Layer 3 Distribution Leaf Switch: SDG Agent mode is the default mode with mDNS gateway configuration. 	<pre>! mdns-sd gateway !</pre>	<pre>! mdns-sd gateway mode sdg-agent !</pre>

Configuration Step	WLC Service-Peer	Layer 3 Distribution Leaf Switch SDG Agent
Step 2: Create a unique mDNS inbound policy to permit ingress AirPrint service announcement on the WLC in Service-Peer mode.	<pre>! mdns-sd service-list LOCAL-AREA-SERVICES-IN in match printer-ipp !</pre>	!
Step 3: Create a unique mDNS outbound policy to permit egress AirPrint service response on the WLC in Service-Peer mode.	<pre>! mdns-sd service-list LOCAL-AREA-SERVICES-OUT out match printer-ipp !</pre>	!
Step 4: Associate the inbound and outbound service lists to a unique service-policy.	<pre>! mdns-sd service-policy LOCAL-AREA-POLICY service-list LOCAL-AREA-SERVICES-IN service-list LOCAL-AREA-SERVICES-OUT !</pre>	!
<p>Step 5: Activate unicast mDNS gateway and attach the service-policy on the WLAN and the wired VLAN.</p> <ul style="list-style-type: none"> • WLC: Activate mDNS gateway for each WLAN Policy Profile and SSID. • Switch: Activate mDNS gateway for each VLAN. 	<pre>! wireless profile policy WLAN-PROFILE shutdown mdns-sd service-policy LOCAL-AREA-POLICY no shutdown ! wlan WLAN-PROFILE 1 blizzard shutdown mdns-sd-interface gateway no shutdown !</pre>	<pre>! ! mDNS Unicast based gateway ! vlan configuration 30 mdns-sd gateway !</pre>
Step 6: (Optional) Enable service-routing on the wired Service-Peer mDNS between the local VLANs. Enable location-based wireless service on the WLC.	<pre>! mdns-sd service-policy LOCAL-AREA-POLICY location ap-location !</pre>	!
<p>Step 7: Enable unicast service-routing between the wired and wireless Service-Peer and SDG Agent.</p> <p>Configure the SDG Agent's IP address and the wireless management source VLAN's ID and IP address on the WLC.</p>	<pre>! mdns-sd gateway source-interface vlan 4094 sdg-agent 10.0.2.254 !</pre>	!

Configuration Step	WLC Service-Peer	Layer 3 Distribution Leaf Switch SDG Agent
Step 8: Configure the wireless user VLAN with the IP address, Layer 3 VRF, and any other parameters, as required.	!	! interface Vlan 30 description BLUE VRF WIRELESS USER VLAN vrf forwarding BLUE_VRF ip address 10.1.3.254 255.255.255.0 no shutdown !

Configuring Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks

To configure Wide Area Bonjour and connect with wired or wireless endpoints across the IP network boundary, enable unicast-mode service-routing on:

- the Layer 3 access leaf switch SDG agent
- the WLC distribution leaf switch SDG agent

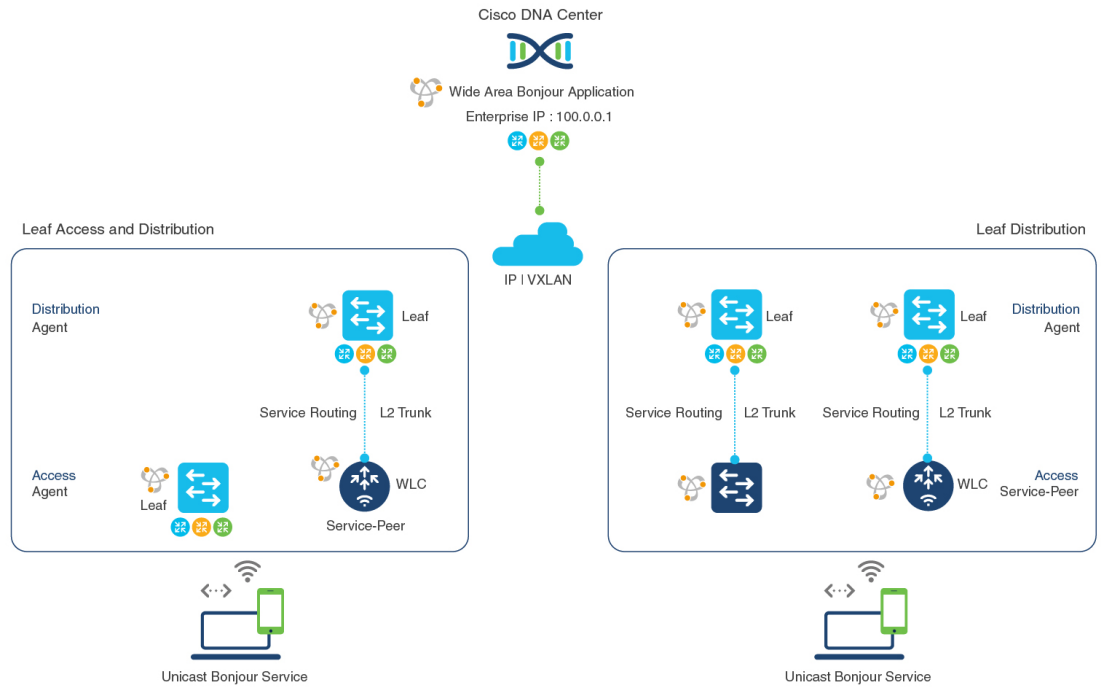
Once you configure both SDG agents, you enable service-routing peering with Cisco DNA Center Wide Area Bonjour application that supports BGP EVPN VXLAN. Once you enable service-routing, global discovery and distribution is enabled and the Layer 3 leaf switch communicates with Cisco DNA Center to perform service-discovery beyond its boundary. mDNS services can be discovered and distributed among the endpoints in the same or in different Layer 3 overlay networks.

For detailed steps about how to configure Wide Area Bonjour service-routing and service policies on a Cisco DNA Center, see *Wide Area Bonjour Configuration Guidelines* in [Cisco Wide Area Bonjour Application on Cisco DNA Center User Configuration Guide, 2.1.2 Release](#).



Note We recommend that you use the Loopback interface to establish service-routing between the leaf switches and Cisco DNA Center in the underlay network.

Figure 66: Deployment Model for Wide Area Bonjour over BGP EVPN VXLAN



The following table shows how to enable unicast-based service-routing on Layer 3 overlay networks connected to wired endpoints or wireless endpoints across IP network boundary:



Note In Cisco IOS XE 17.6.2 and later releases, you can configure a customized controller service policy.

Cisco IOS XE 17.6.1 and Earlier Releases

Table 60: Configuring SDG Agents to Configure Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks

Configuration Step	Layer 3 Access Leaf SDG Agent	WLC Distribution Leaf SDG Agent
Step 1: Enable mDNS gateway and set the gateway mode.	! mdns-sd gateway !	! mdns-sd gateway !
Step 2: Activate the unicast mDNS gateway on the wired VLAN and Wireless user VLAN on the respective SDG Agents.	! vlan configuration 10 mdns-sd gateway !	! vlan configuration 30 mdns-sd gateway !

Configuration Step	Layer 3 Access Leaf SDG Agent	WLC Distribution Leaf SDG Agent
<p>Step 3: Create unique controller-bound mDNS policy to permit egress AirPrint service discovery and distribution from the Catalyst Switch in SDG Agent mode.</p> <p>Inbound policy towards the controller is not required.</p>	<pre>! mdns-sd service-list WIDE-AREA-SERVICES-OUT out match printer-ipp !</pre>	<pre>! mdns-sd service-list WIDE-AREA-SERVICES-OUT out match printer-ipp !</pre>
<p>Step 4: Associate the outbound service-list to a unique service-policy.</p>	<pre>! mdns-sd service-policy WIDE-AREA-POLICY service-list WIDE-AREA-SERVICES-OUT !</pre>	<pre>! mdns-sd service-policy WIDE-AREA-POLICY service-list WIDE-AREA-SERVICES-OUT !</pre>
<p>Step 5: Enable Wide Area Bonjour service-routing. Configure the service export with the associated controller parameters like IP Address, source interface for stateful connection, and mandatory egress policy for Wide Area service-routing.</p>	<pre>! service-export mdns-sd controller DNAC-CONTROLLER-POLICY controller-address 100.0.0.1 controller-source-interface LOOPBACK 0 controller-service-policy WIDE-AREA-POLICY !</pre>	<pre>! service-export mdns-sd controller DNAC-CONTROLLER-POLICY controller-address 100.0.0.1 controller-source-interface LOOPBACK 0 controller-service-policy WIDE-AREA-POLICY !</pre>

Cisco IOS XE 17.6.2 and Later Releases

Table 61: Configuring SDG Agents to Configure Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks

Configuration Step	Layer 3 Access Leaf SDG Agent	WLC Distribution Leaf SDG Agent
<p>Step 1: Enable mDNS gateway and set the gateway mode.</p>	<pre>! mdns-sd gateway !</pre>	<pre>! mdns-sd gateway !</pre>

Configuration Step	Layer 3 Access Leaf SDG Agent	WLC Distribut
Step 2: Activate the unicast mDNS gateway on the wired VLAN and Wireless user VLAN on the respective SDG Agents.	! <pre>vlan configuration 10 mdns-sd gateway !</pre>	! <pre>vlan config mdns-sd gat !</pre>
Step 3: Enable Wide Area Bonjour service-routing. Configure the service export with the associated controller parameters like IP Address and source interface for stateful connection for Wide Area service-routing. Inbound policy towards the controller is not required.	! <pre>service-export mdns-sd controller DNAC-CONTROLLER-POLICY controller-address 100.0.0.1 controller-source-interface LOOPBACK 0 !</pre>	! <pre>service-expo controller DNAC-CONTRO controller- controller- LOOPBACK 0 !</pre>
Step 4: (Optional) Create a custom controller service list to permit egress AirPrint service discovery and distribution from the Catalyst Switch in SDG Agent mode.	! <pre>mdns-sd controller service-list WIDEAREA-SERVICES match printer-ipp !</pre>	! <pre>mdns-sd cont service-list match printe !</pre>
Step 5: (Optional) Associate the controller service list to a custom controller service policy.	! <pre>mdns-sd controller service-policy WIDE-AREA-POLICY service-list WIDE-AREA-SERVICES !</pre>	! <pre>mdns-sd cont WIDE-AREA-P service-list WIDE-AREA-S !</pre>
Step 6: (Optional) Associate the custom controller service policy to Wide Area Service routing. Note A default controller service policy is configured if you do not create a custom controller service policy.	! <pre>service-export mdns-sd controller DNAC-CONTROLLER-POLICY controller-service-policy WIDE-AREA-POLICY !</pre>	! <pre>service-expo DNAC-CONTRO controller- WIDE-AREA-P !</pre>

Verifying Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks

This section shows how to verify Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 overlay networks.

Verifying Local Area Bonjour over an EVPN VXLAN Layer 3 Overlay Network

This section shows how to verify Local Area Bonjour over EVPN VXLAN Layer 3 overlay networks. The examples in this section show the mDNS configuration, service discovery status, and service distribution status for the policies applied on devices in Service-Peer and SDG Agent modes.

Verifying Wired Service-Peer Configuration

The examples in this section show how to verify the wired Service-Peer configuration on the Cisco Catalyst Series switch (SP-1) in Service-Peer mode. You can determine the operational status once you configure and discover AirPrint service from the local network.

This example shows a sample output for the **show mdns-sd summary vlan *vlan-id*** command on SP-1:

```
SP-1# show mdns-sd summary vlan 10
VLAN : 10
=====
mDNS Gateway : Enabled
mDNS Service Policy : LOCAL-AREA-POLICY
Active Query : Enabled
Periodicity : 3600 Seconds
Transport Type : IPv4
Service Instance Suffix : Not-Configured
mDNS Query Type : ALL
SDG Agent IP : 10.0.1.254
Source Interface : Vlan4094
```

SP-1#

This example shows a sample output for the **show mdns-sd service-policy name *policy-name*** command on SP-1:

```
SP-1# show mdns-sd service-policy name LOCAL-AREA-POLICY
Service Policy Name Service List IN Name Service List Out Name
=====
LOCAL-AREA-POLICY LOCAL-AREA-SERVICES-IN LOCAL-AREA-SERVICES-OUT
```

SP-1#

This example shows a sample output for the **show mdns-sd cache vlan *vlan-id*** command on SP-1:

```
SP-1# show mdns-sd cache vlan 10
NAME                               TYPE          TTL/Remaining  Vlan-Id/If-name
Mac Address      RR Record Data
_universal._sub._ipp._tcp.local    PTR          4500/4486     V110
ac18.2651.03fe   Bldg-1-FL1-PRN._ipp._tcp.local    PTR          4500/4486     V110
_ac18.2651.03fe   Bldg-1-FL1-PRN._ipp._tcp.local    SRV          4500/4486     V110
Bldg-1-FL1-PRN._ipp._tcp.local     0 0 631 Bldg-1-FL1-PRN.local              A           4500/4486     V110
ac18.2651.03fe   10.1.1.1
Bldg-1-FL1-PRN.local               AAAA         4500/4486     V110
ac18.2651.03fe   2001:10:153:1:79:A40C:6BEE:AEEC
Bldg-1-FL1-PRN._ipp._tcp.local     TXT          4500/4486     V110
ac18.2651.03fe   (451)'txtvers=1''priority=30''ty=EPSON WF-3620
Series''usb_MFG=EPSON''usb_MDL=W~'~
```

SP-1#

This example shows a sample output for the **show mdns-sd statistics vlan *vlan-id*** command on SP-1:

```
SP-1# show mdns-sd statistics vlan 10
mDNS Statistics
```



```

Vl10:
mDNS packets sent : 612
  IPv4 sent : 612
    IPv4 advertisements sent : 0
    IPv4 queries sent : 612
  IPv6 sent : 0
    IPv6 advertisements sent : 0
    IPv6 queries sent : 0
Unicast sent : 0
mDNS packets rate limited : 0
mDNS packets received : 42
  advertisements received : 28
  queries received : 14
    IPv4 received : 42
      IPv4 advertisements received : 28
      IPv4 queries received : 14
    IPv6 received : 0
      IPv6 advertisements received : 0
      IPv6 queries received : 0
mDNS packets dropped : 0
=====
Query Type                               : Count
=====
PTR                                       : 12
SRV                                       : 0
A                                         : 0
AAAA                                      : 0
TXT                                       : 0
ANY                                       : 3
=====
PTR Name                                Advertisement      Query
=====
_ipp._tcp.local                          9                  4
SP-1#

```

Verifying Wired SDG Agent Configuration and Service-Routing Status

The examples in this section show how to verify the mDNS configuration and service-routing on Wired SDG Agent (SDG-1). SDG-1 is locally connected to a Layer 2 access switch (SP-1) in Service-Peer mode and centrally paired with a Cisco DNA Center for Wide Area Bonjour service-routing.

This example shows a sample output for the **show mdns-sd summary vlan *vlan-id*** command on SDG-1:

```

SP-1# show mdns-sd summary vlan 10
VLAN : 10
=====
mDNS Gateway           : Enabled
mDNS Service Policy    : LOCAL-AREA-POLICY
Active Query           : Disabled
Transport Type         : IPv4
Service Instance Suffix : Not-Configured
mDNS Query Type       : ALL
SDG Agent IP           : Not-Configured
Source Interface       : Not-Configured
SDG-1#

```

This example shows a sample output for the **show mdns-sd cache vlan *vlan-id*** command on SDG-1:

```
SP-1# show mdns-sd cache vlan 10

NAME                               TYPE      TTL/Remaining  Vlan-Id/If-name
  Mac Address      RR Record Data
_universal._sub._ipp._tcp.local    PTR      4500/4500      V110
 ac18.2651.03fe      Bldg-1-FL1-PRN._ipp._tcp.local
_ipp._tcp.local                    PTR      4500/4500      V110
 ac18.2651.03fe      Bldg-1-FL1-PRN._ipp._tcp.local
Bldg-1-FL1-PRN._ipp._tcp.local    SRV      4500/4500      V110
 ac18.2651.03fe      0 0 631      Bldg-1-FL1-PRN.local
Bldg-1-FL1-PRN.local              A        4500/4500      V110
 ac18.2651.03fe      10.1.1.1
Bldg-1-FL1-PRN.local              AAAA     4500/4500      V110
 ac18.2651.03fe      2001:10:153:1:79:A40C:6BEE:AEEC
Bldg-1-FL1-PRN._ipp._tcp.local    TXT      4500/4500      V110
 ac18.2651.03fe      (451)'txtvers=1''priority=30''ty=EPSON WF-3620
Series''usb_MFG=EPSON''usb_MDL=W~'~

SDG-1#
```

This example shows a sample output for the **show mdns-sd sp-sdg statistics** command on SDG-1:

```
SP-1# show mdns-sd sp-sdg statistics

                                One min, 5 mins, 1 hour
Average Input rate (pps)       :      0,      0,      0
Average Output rate (pps)     :      0,      0,      0
Messages received:
  Query                        : 15796
  ANY query                    : 0
  Advertisements               : 28
  Advertisement Withdraw      : 0
  Interface down               : 0
  Vlan down                    : 0
  Service-peer ID change      : 0
  Service-peer cache clear    : 12
  Resync response              : 6
Messages sent:
  Query response               : 5975
  ANY Query response           : 0
  Cache-sync                   : 61
  Get service-instance         : 0

SDG-1#
```

This example shows a sample output for the **show mdns-sd controller detail** command on SDG-1:

```
SP-1# show mdns-sd controller detail

Controller : DNAC-Policy
IP : 100.0.0.1, Dest Port : 9991, Src Port : 42446, State : UP
Source Interface : Loopback0, MD5 Disabled
Hello Timer 30 sec, Dead Timer 120 sec, Next Hello 00:00:24
Uptime 2d05h (17:02:37 UTC Jan 15 2021)
Service Buffer : Enabled

Service Announcement :
Filter : DNAC-CONTROLLER-POLICY
Count 50, Delay Timer 30 sec, Pending Announcement 0, Pending Withdraw 0
Total Export Count 56, Next Export in 00:00:24
```

```
Service Query :
Query Suppression Enabled
Query Count 50, Query Delay Timer 15 sec, Pending 0
Total Query Count 15791, Next Query in 00:00:09
```

```
SDG-1#
```

Verifying Wireless Service-Peer and Service Routing Status

The examples in this section show how to verify the wireless Service-Peer configuration on the Cisco Catalyst 9800 WLC (SP-2) in Service-Peer mode. You can determine the operational status once you configure and discover AirPrint service from the remote network.

This example shows a sample output for the **show mdns-sd summary** command on SP-2:

```
SP-2# show mdns-sd summary
mDNS Gateway: Enabled
Mode: Service Peer
Service Announcement Periodicity(in seconds): 30
Service Announcement Count: 50
Service Query Periodicity(in seconds): 15
Service Query Count: 50
Active Response Timer (in seconds): Disabled
ANY Query Forward: Disabled
SDG Agent IP: 10.0.2.254
Source Interface: Vlan4094
Active Query Periodicity (in minutes): 15
Transport Type: IPv4
mDNS AP service policy: default-mdns-service-policy
```

```
SP-2#
```

This example shows a sample output for the **show wireless profile policy detailed wireless-profile-name** command on SP-2:

```
SP-2# show wireless profile policy detailed WLAN-PROFILE | sec mDNS
mDNS Gateway
  mDNS Service Policy name          : LOCAL-AREA-POLICY
```

```
SP-2#
```

This example shows a sample output for the **show mdns-sd statistics wlan-id wlan-id-value** command on SP-2:

```
SP-2# show mdns-sd statistics wlan-id 1
mDNS Packet Statistics
-----
mDNS stats last reset time: 01/10/21 21:38:19
mDNS packets sent: 4592
  IPv4 sent: 4592
    IPv4 advertisements sent: 4592
    IPv4 queries sent: 0
  IPv6 sent: 0
    IPv6 advertisements sent: 0
    IPv6 queries sent: 0
  Multicast sent: 0
    IPv4 sent: 0
    IPv6 sent: 0
mDNS packets received: 297
```

```

advertisements received: 80
queries received: 217
IPv4 received: 297
  IPv4 advertisements received: 80
  IPv4 queries received: 217
IPv6 received: 0
  IPv6 advertisements received: 0
  IPv6 queries received: 0
mDNS packets dropped: 297
Query Type Statistics
  PTR queries received: 1720
  SRV queries received: 8
  A queries received: 8
  AAAA queries received: 8
  TXT queries received: 97
  ANY queries received: 153
  OTHER queries received: 0

SP-2#

```

This example shows a sample output for the **show mdns-sd sp-sdg statistics** command on SP-2:

```

SP-2# show mdns-sd sp-sdg statistics
mDNS SP Statistics
last reset time: 01/10/21 21:37:36

Messages sent:
  Query                : 12675
  ANY query            : 0
  Advertisements       : 24
  Advertisement Withdraw : 0
  Service-peer ID change : 0
  Service-peer cache clear : 7
  Resync response      : 5
Messages received:
  Query response       : 4619
  ANY Query response  : 0
  Cache-sync          : 48
  Get service-instance : 0

```

SP-2#

This example shows a sample output for the **show mdns-sd query-db** command on SP-2:

```

SP-2# show mdns-sd query-db
MDNS QUERY DB

Client MAC: 4c32.7593.e3af
Vlan ID: 30
Wlan ID: 1
Location Group ID: 0
PTR Name(s):
  _ipp._tcp.local

```

SP-2#

Verifying Wide Area Bonjour over EVPN VXLAN Layer 3 Overlay Networks

This section shows how to verify Wide Area Bonjour over EVPN VXLAN Layer 3 overlay networks.

The examples in this section show the mDNS configuration and service-routing on a wireless SDG-Agent (SDG-2). SDG-2 is locally connected to a Catalyst 9800 series WLC (SP-2) in Service-Peer mode and centrally paired with a Cisco DNA Center for Wide Area Bonjour service-routing.

This example shows a sample output for the **show mdns-sd summary vlan *vlan-id*** command on SDG-2.

```
SDG-2# show mdns-sd summary vlan 30
VLAN : 30
=====
mDNS Gateway      : Enabled
mDNS Service Policy : LOCAL-AREA-POLICY
Active Query      : Disabled
Transport Type    : IPv4
Service Instance Suffix : Not-Configured
mDNS Query Type   : ALL
SDG Agent IP      : Not-Configured
Source Interface   : Not-Configured

SDG-2#
```

This example shows a sample output for the **show mdns-sd sp-sdg statistics** command on SDG-2.

```
SDG-2# show mdns-sd sp-sdg statistics
                                One min, 5 mins, 1 hour
Average Input rate (pps) :      0,      0,      0
Average Output rate (pps) :      0,      0,      0
Messages received:
  Query                      : 12191
  ANY query                   : 0
  Advertisements              : 0
  Advertisement Withdraw      : 0
  Interface down              : 0
  Vlan down                   : 0
  Service-peer ID change      : 0
  Service-peer cache clear    : 18
  Resync response              : 10
Messages sent:
  Query response              : 1975
  ANY Query response          : 0
  Cache-sync                  : 19
  Get service-instance        : 0

SDG-2#
```

This example shows a sample output for the **show mdns-sd controller detail** command on SDG-2.

```
SDG-2# show mdns-sd controller detail
Controller : DNAC-Policy
IP : 100.0.0.1, Dest Port : 9991, Src Port : 42931, State : UP
Source Interface : Loopback0, MD5 Disabled
Hello Timer 30 sec, Dead Timer 120 sec, Next Hello 00:00:19
Uptime 2d05h (17:10:18 UTC Jan 15 2021)
Service Buffer : Enabled

Service Announcement :
Filter : DNAC-CONTROLLER-POLICY
Count 50, Delay Timer 30 sec, Pending Announcement 0, Pending Withdraw 0
Total Export Count 0, Next Export in 00:00:19

Service Query :
```

```

Query Suppression Enabled
Query Count 50, Query Delay Timer 15 sec, Pending 0
Total Query Count 17093, Next Query in 00:00:19

```

SDG-2#

Verifying Cisco DNA Center Configuration and Service-Routing Status

The Cisco Wide Area Bonjour application supports multilevel assurance capabilities for day-to-day operations. The service routing, instance monitoring, management, and troubleshooting in the Cisco Wide Area Bonjour application are divided into three major categories. Each category serves a unique function to manage and troubleshoot Wide Area Bonjour service routing for day-2 operation. The Monitor function comprises the following categories:

- **Dashboard:** The landing page of Cisco Wide Area Bonjour application provides an overview of key statistics in multiple formats. It allows you to quickly determine the service-routing health across the network. The network administrator can monitor the operational status of service-routing using:
- **Sub-Domain 360°:** Network administrators can collect statistics and status counts in the 360° view. The left-panel monitoring and configuration bar opens automatically upon clicking a subdomain. You can verify configured policies and discovered service instances on a per-subdomain basis of the Configuration section.
- **Monitor:** A comprehensive three-tier monitoring and troubleshooting function of the Cisco Wide Area Bonjour application for various day-2 operations. Network administrators use the detail view of the SDG Agent, service instance, and advanced troubleshooting capabilities to manage and troubleshoot the Wide Area Bonjour domain with a single pane of glass in Cisco DNA Center.

For detailed information about assurance capabilities and monitoring of operation details, see [Monitor the Cisco Wide Area Bonjour Application](#) module of Cisco Wide Area Bonjour on Cisco DNA Center User Guide, Release 2.1.2.

Additional References for Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks

Table 62: Additional References for Cisco DNA Service for Bonjour over EVPN VXLAN Layer 3 Overlay Networks

Related Topic	Document Title
DNA Service for Bonjour Deployment on Cisco Catalyst 9500 Series Switches	Cisco DNA Service for Bonjour Configuration Guide (Catalyst 9500 Switches)
DNA Service for Bonjour Deployment on Cisco Catalyst 9800 Wireless LAN Controllers	Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide
Cisco Wide Area Bonjour Application on Cisco DNA Center User Guide	Cisco Wide Area Bonjour Application on Cisco DNA Center User Guide, Release 2.1.2



CHAPTER 16

Configuring VRF-Aware Local Area Bonjour Services

Beginning from Cisco IOS XE Bengaluru 17.4.1, Cisco Catalyst 9000 Series switches supports Virtual Routing and Forwarding-Aware (VRF-Aware) services in Local Area Bonjour domain. VRF-Aware Local Area Bonjour services provide boundary-based service discovery for Layer 3 segmented IPv4 and IPv6 network and support policy-based (secure) routing services for Wired and Wireless networks. VRF-Aware Local Area Bonjour service is supported on enterprise-grade, traditional, and next-generation fabric-based deployment models as described in [Cisco DNA Service for Bonjour Solution Overview](#).

- [Prerequisites for VRF-Aware Local Area Bonjour Services, on page 627](#)
- [Restrictions for VRF-Aware Local Area Bonjour Services, on page 628](#)
- [Information about VRF-Aware Local Area Bonjour Services, on page 628](#)
- [Understanding VRF-Aware Wide Area Bonjour Services, on page 630](#)
- [Understanding VRF-Aware Service on Multilayered Wired and Wireless Networks, on page 632](#)
- [How to configure Intra-Virtual Network Proxy Service on Local Area Bonjour Domain, on page 633](#)
- [How to configure Inter-Virtual Network Proxy Service on Local Area Bonjour Domain, on page 634](#)
- [Verifying VRF-Aware Local Area Bonjour Services, on page 637](#)

Prerequisites for VRF-Aware Local Area Bonjour Services

- You must understand the mDNS service segmentation capabilities to implement, manage, and troubleshoot the proxy service in Local Area Bonjour domain.
- Ensure that the Cisco Catalyst 9000 Series switch is configured in SDG-Agent mode. VRF-Aware Local Area Bonjour service is supported on first-hop IP gateway of switches configured in SDG-Agent mode in Wired and Wireless networks.
- Ensure that the software version installed on the Cisco Catalyst 9000 Series switch is Cisco IOS XE Bengaluru 17.4.1 or higher.
- Ensure that all required IP VRF with IPv4 or IPv6 address-family configurations is completed. These configurations are required to activate VRF on the switch configured in SDG-Agent mode.
- Ensure that the IP VRF configured to a local SVI interface supports IP gateway so that the mDNS Wired and Wireless endpoint can be attached directly or remotely.

- To activate mDNS gateway in Unicast mode for a VLAN, ensure that the mDNS gateway and service policy is configured after enabling the VLAN using the **vlan configuration id** command.
- Ensure that all configurations for IPv4 or IPv6-based data routing and forwarding both within the same VRF or different VRFs are complete including network requirements such as stateful firewall configuration, route-leaking configuration and so on.
- Ensure that all the prerequisites described in *Configuring Local Area Bonjour in Unicast Mode for LAN Networks* module are completed.

Restrictions for VRF-Aware Local Area Bonjour Services

- VRF-Aware Local Area Bonjour service is not supported on a Layer 2 Cisco Catalyst 9000 Series switch or a Layer 2 Cisco Catalyst 9800 WLC in Service-Peer mode.
- VRF-Aware Local Area Bonjour services are configured to provide mDNS service discovery information between Layer 3 segments within the same or different IP VRF, or share services from non-VRF enabled networks only. Any additional IP routing and data forwarding configurations are beyond the scope of this implementation.

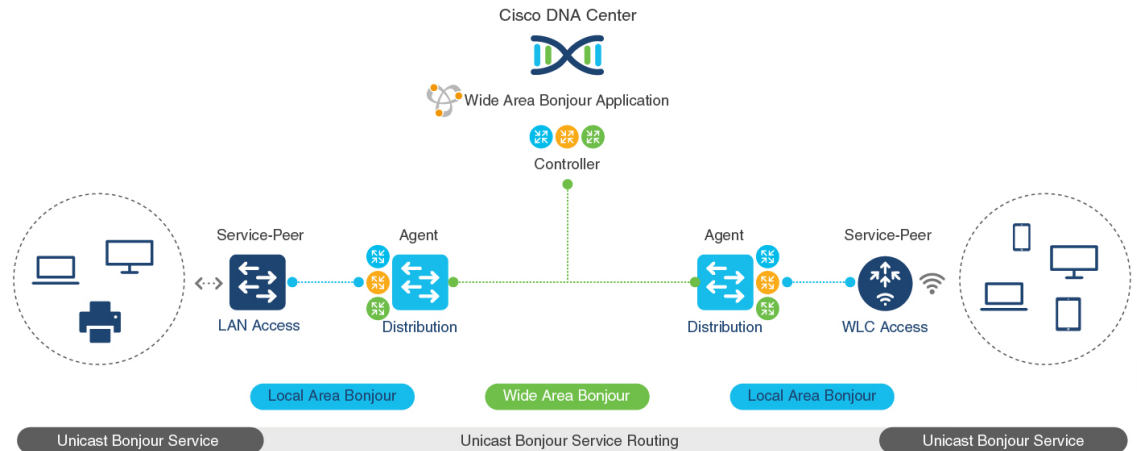
Information about VRF-Aware Local Area Bonjour Services

The Cisco DNA Service for Bonjour solution provides end-to-end service-routing for enterprise-grade Wired and Wireless networks. The enterprise network builds secure and segmented networks that protect IT-managed infrastructure and shares services and resources among trusted and untrusted user group. The physical infrastructure can be logically virtualized into a private networking space that supports secure communication services within closed user groups and conditionally extends boundary services based on business and technical demands.

VRF-Aware Local Area Bonjour gateway services allow to dynamically discover and distribute mDNS services on the same VRF segmented Layer 3 overlay networks based on policy. You can also build an Extranet network using the mDNS location-filter policy that supports proxy services among multiple logical VRF or a global IP routing domain on a local system. The Layer 3 VRF segmented networks can also be configured to route in overlay using any next-generation overlay networks such as Cisco SD-Access, BGP EVPN VXLAN or classic technologies such as Multi-VRF, MPLS.

[Figure 67: Cisco DNA Service for Bonjour with VRF-Aware Services](#) illustrates the Cisco DNA Service for Bonjour solution configured with VRF-Aware services for enterprise-grade Wired and Wireless networks.

Figure 67: Cisco DNA Service for Bonjour with VRF-Aware Services



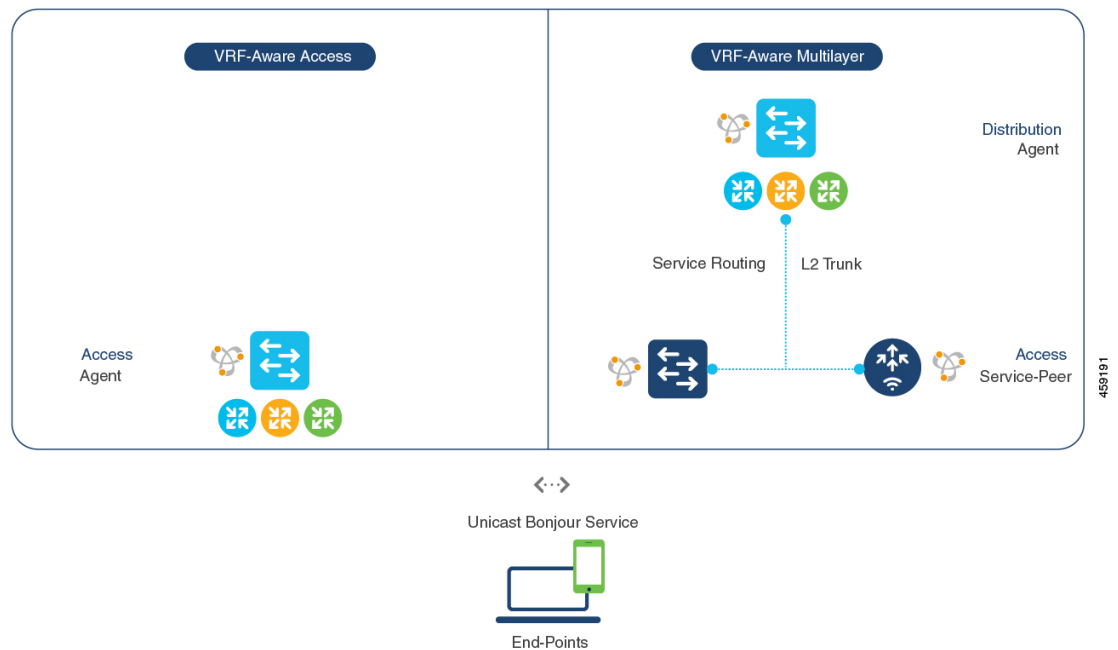
459190

Gateway Modes for VRF-Aware Bonjour Services

VRF is a Layer 3 specific virtual routing function and therefore it is implemented on Layer 3 Ethernet switches with first-hop IP gateways that can directly or remotely attach mDNS endpoints.

Figure 68: [Gateway Modes for VRF-Aware Services](#) illustrates the Cisco Catalyst 9000 Series switch in SDG-Agent mode supporting VRF-Aware services in Layer 3 access mode and in multi-layer network deployment mode. In multi-layer network deployment mode, the gateway to the distribution layer provides a Layer 2 or Layer 3 boundary to a downstream Layer 2 Cisco Catalyst 9000 Series switch and Cisco Catalyst 9800 WLC for local proxy service with local VLANs.

Figure 68: Gateway Modes for VRF-Aware Services



- **VRF-Aware Routed Access:** The Cisco Catalyst 9000 Series switch can be deployed as an IP gateway for directly attached Wired or Wireless mDNS endpoints. The Cisco Wireless SSID can be configured as fabric-enabled or as FlexConnect with local switching that provides local termination point to a first-hop Ethernet switch that supports Layer 3 overlay networks such as Cisco SD-Access or BGP EVPN based-fabric networks. A Cisco Catalyst 9000 Series switch configured in SDG-Agent mode provides unicast-based mDNS gateway services to directly attached Wired and Wireless endpoints within the same or different virtual routing network space or a default global IP network.
- **VRF-Aware Multilayer:** The Cisco Catalyst 9000 Series Switch can be deployed as an IP gateway for remotely attached Wired or Wireless mDNS endpoints through an intermediate Layer 2 Cisco Catalyst 9000 Series switch or Cisco Catalyst 9800 Series WLC. A Cisco Catalyst 9000 Series switch, configured in SDG-Agent mode and in the distribution layer, provides VRF-Aware mDNS gateway services, while the Layer 2 Ethernet switch and Cisco WLC in Unicast mode provides local proxy services to directly attached Wired and Wireless endpoints within the same or different VLAN.

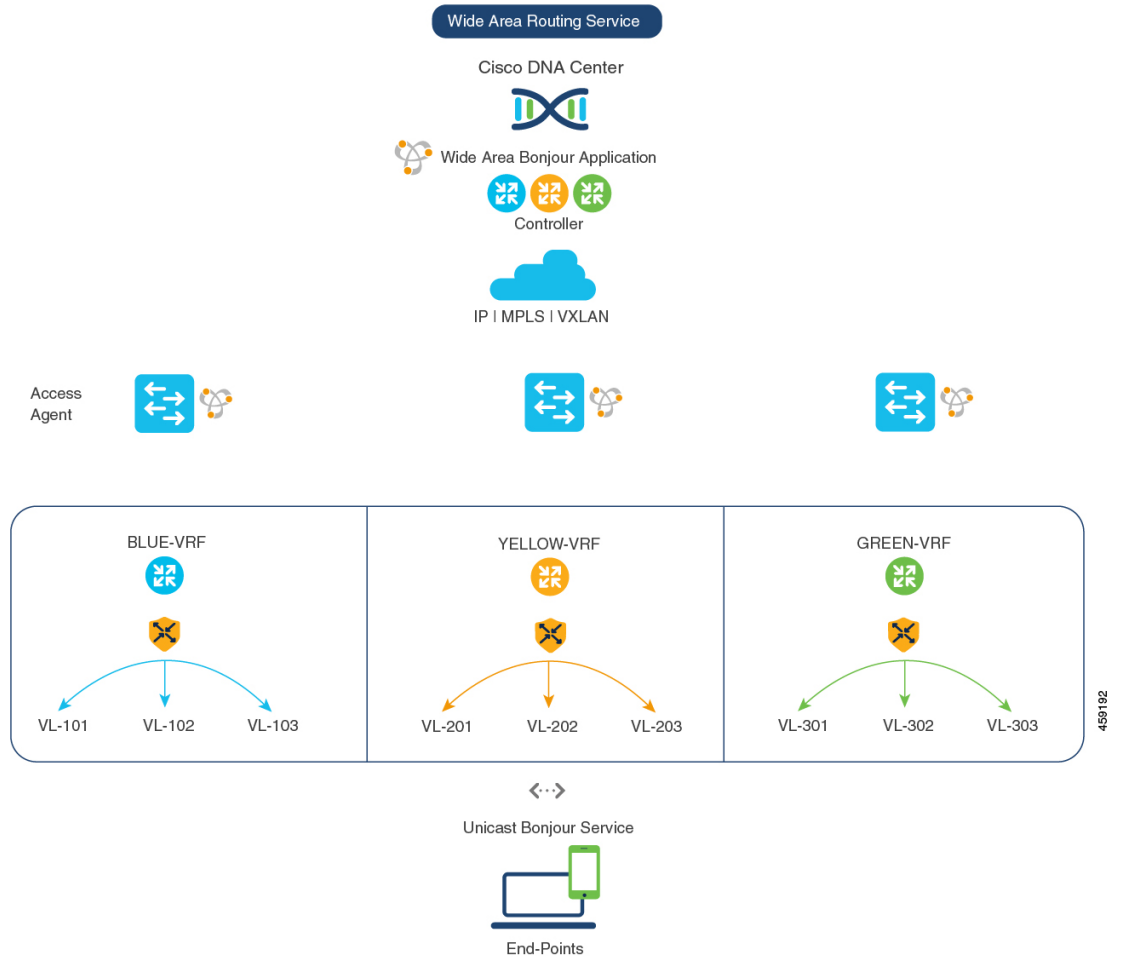
Understanding VRF-Aware Wide Area Bonjour Services

The VRF-Aware service discovery and distribution can be implemented across multiple switches in SDG-Agent mode on an IP, MPLS, or VXLAN-enabled network with Wide Area Bonjour. The Cisco DNA-Center Wide Area Bonjour application supports granular and policy-based routing services that allow discovery and distribution of mDNS services dynamically over overlay networks. You can build a global policy combining one or more source and receiver SDG-Agents that allow distributing or advertising services from a specific IPv4 or even an IPv6 network mapped to the VRF.

The network wide and distributed switches in SDG-Agent mode transport locally discovered or requested mDNS service information over lightweight unicast routing services to a centralized Cisco DNA-Center controller in an underlay IPv4 network. These switches must be configured with a unified service-export policy for local networks mapped to one or more VRFs or to a global IP routing domain.

Figure 69: VRF-Aware Wide Area Bonjour Services illustrates VRF-Aware Wide Area Bonjour services for IP, MPLS, or VXLAN enabled overlay networks.

Figure 69: VRF-Aware Wide Area Bonjour Services



The *Configuring Wide Area Bonjour* module lists the configuration procedures in detail.

Understanding VRF-Aware Service on Multilayered Wired and Wireless Networks

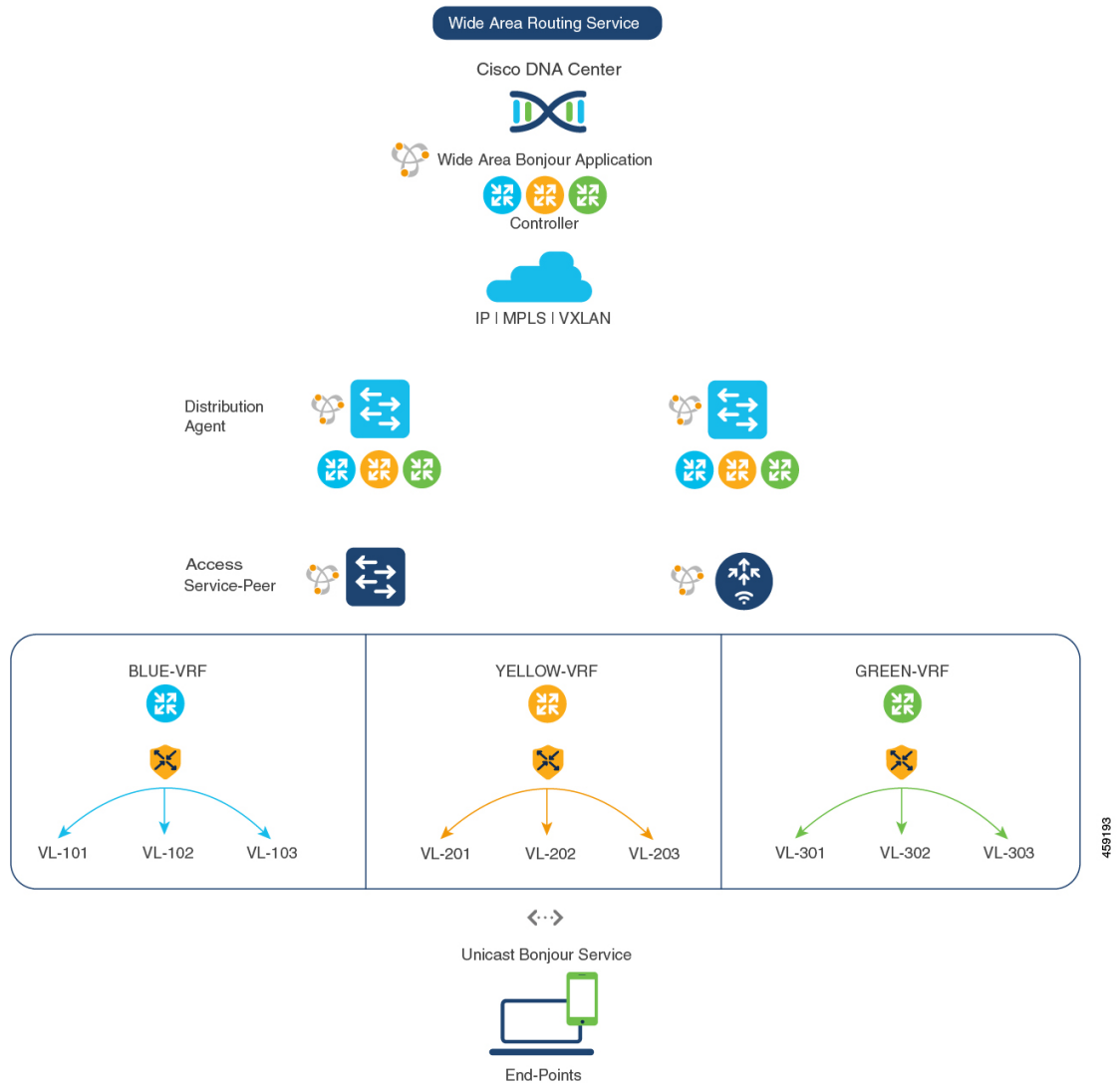
The Cisco Catalyst 9000 Series switches support VRF-Aware service for multilayered Wired and central-switching Wireless-enabled networks. The Layer 2 or Layer 3 network boundary to the Cisco Catalyst 9000 Series switches is extended at the distribution layer with an intermediate Layer 2 Cisco Catalyst 9000 Series switch or Cisco Catalyst 9800 Series WLC and directly attached to the Wired and central-switching Wireless endpoints. As the IP gateway shifts, the Cisco Catalyst 9000 Series switches in the distribution layer must be configured in SDG-Agent mode and the downstream Layer 2 switch and WLC network devices must be configured in Service-Peer mode to support mDNS proxy services to locally attached endpoints.

The VRF-Aware service configured on a switch, in SDG-Agent mode and in the distribution layer, follows configuration and operation guidelines for Wired and central-switching Wireless as described in [Understanding VRF-Aware Wide Area Bonjour Services, on page 630](#). The Layer 2 switch and WLC network devices remains transparent to VRF-Aware services and continues to provide local proxy services to locally attached users in the same or different VLANs.

The VRF-Aware service discovery and distribution can be implemented across multiple switches in SDG-Agent mode on an IP, MPLS, or VXLAN-enabled network with Wide Area Bonjour. The Cisco DNA-Center Wide Area Bonjour application supports granular and policy-based routing services that allow discovery and distribution of mDNS services dynamically for overlay networks. You can build a global policy combining one or more source and receiver SDG-Agent that allow distributing or advertising services from a specific IPv4 or even an IPv6 network mapped to the VRF.

Figure 70: VRF-Aware on Multilayered Wired and Wireless Network illustrates end-to-end VRF-Aware on multilayered Wired and Wireless networks across Wide Area Bonjour domain with Cisco DNA-Center.

Figure 70: VRF-Aware on Multilayered Wired and Wireless Network



How to configure Intra-Virtual Network Proxy Service on Local Area Bonjour Domain

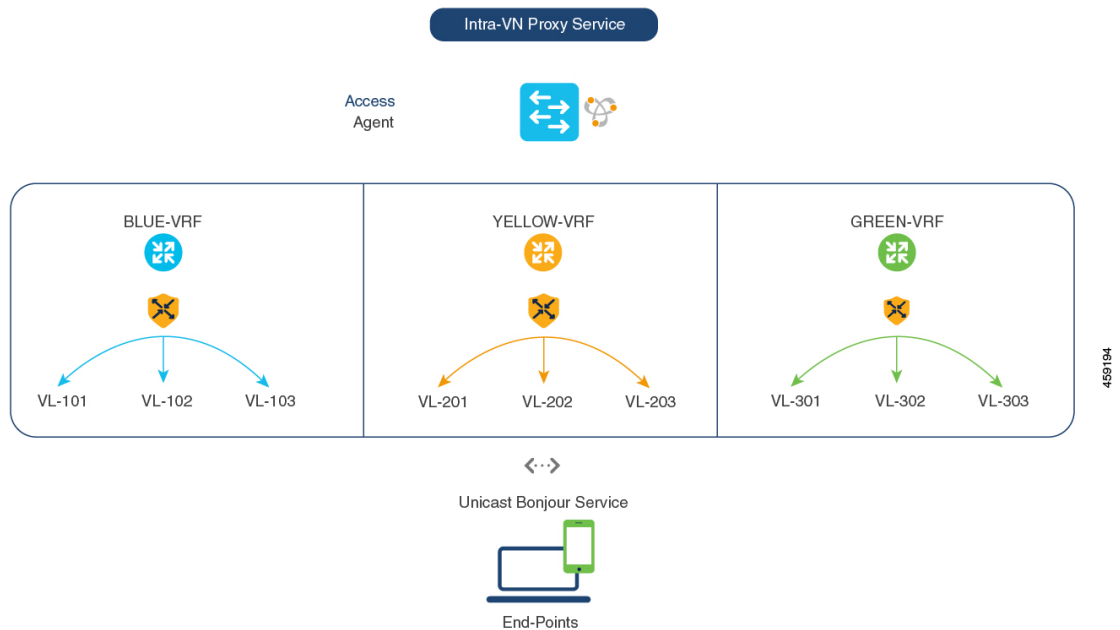
Intra-Virtual Network (Intra-VN) Proxy Service is a policy-based VRF-Aware service discovery and distribution implemented on the IP VRF of a switch in SDG-Agent mode connected to multiple IP networks.

Beginning from Cisco IOS XE Bengaluru 17.4.1, the Cisco Catalyst 9000 Series switches support mDNS gateway service as the default on each VRF. You must build a mDNS service policy that implicitly allows required mDNS service types and mapping services to endpoint facing VLANs. The Cisco Catalyst 9000 Series switch can automatically discover VRF associations to a VLAN interface without additional configurations.

The Cisco Catalyst 9000 Series switch in SDG-Agent mode dynamically discovers mDNS services from a local network and automatically builds VRF-aware service information. To enable Layer 3 segmented proxy service by default, the SDG-Agent provides limited mDNS service proxy response to endpoints in other VLANs mapped with the same VRF.

[Figure 71: Intra-VN Service Proxy](#) illustrates VRF-Aware enabled on an Intra-VN proxy service.

Figure 71: Intra-VN Service Proxy



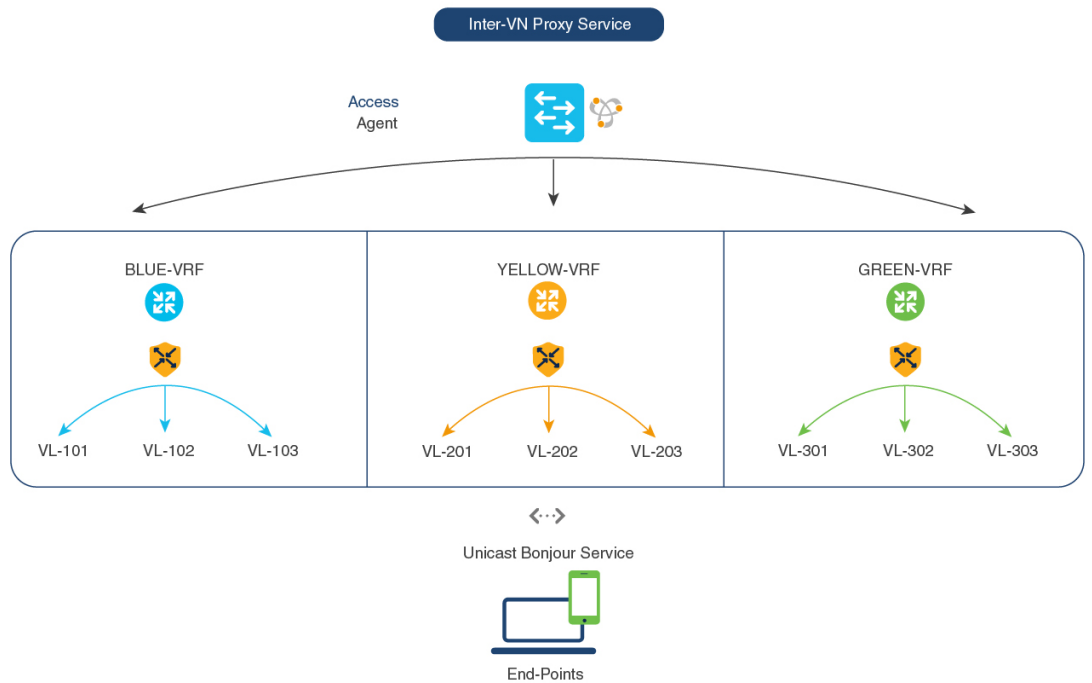
How to configure Inter-Virtual Network Proxy Service on Local Area Bonjour Domain

Inter-VN Proxy Service is a policy-based VRF-Aware service discovery and distribution implemented on multiple IP VRFs or on a global IP routing domain of a switch in SDG-Agent mode connected to multiple IP networks.

Beginning from Cisco IOS XE Bengaluru 17.4.1, the Cisco Catalyst 9000 Series switches support mDNS service discovery and distribution between IP VRFs or on a global routing domain based on the configured mDNS location-filter policy. The existing location-filter configuration on an SDG-Agent permits mDNS service information between configured VLANs and records discovery and distribution on the mapping table. Although configuring inter-VN provides Extranet mDNS proxy services between Wired and Wireless networks, additional methods such as stateful firewall, route-leaking and so on must also be configured to handle the data transfer between Inter-VN or VRF to global IP routing.

[Figure 72: Inter-VN Proxy Service](#) shows Inter-VN proxy service for Extranet network.

Figure 72: Inter-VN Proxy Service



Configuring Inter-Virtual Network Location-Filter

To enable the local service proxy on the switch to discover mDNS services between local VLANs, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	vlan ID Example: Device(config)# vlan 101 Device(config-vlan)# name BLUE-VRF Device(config)# vlan 201 Device(config-vlan)# name YELLOW-VRF Device(config)# vlan 301 Device(config-vlan)# name GREEN-VRF	Configures a VLAN ID in local database for overlay mDNS endpoints.

	Command or Action	Purpose
Step 4	mdns-sd location-filter <i>location-filter-name</i> Example: <pre>Device(config)# mdns-sd location-filter INTER-VN-LOCAL-PROXY</pre>	Configures a unique location-filter.
Step 5	match location {all default ID} vlan [ID] Example: <pre>Device(config-mdns-loc-filter)# match location-group default vlan 101 Device(config-mdns-loc-filter)# match location-group default vlan 201 Device(config-mdns-loc-filter)# match location-group default vlan 301</pre>	Configures the match criteria that mutually distribute permitted services between grouped VLANs.
Step 6	mdns-sd service-list <i>service-list-name</i> {in out} Example: <pre>Device(config)# mdns-sd service-list BLUE-VRF-LIST-OUT out</pre>	Configures mDNS service-list to classify one or more service-types. Unique service-list is required to process incoming mDNS message and the outbound response to the requesting end points.
Step 7	match <i>service-destination-name</i> [message-type {any announcement query}] Example: <pre>Device(config)# mdns-sd service-list BLUE-VRF-LIST-OUT out Device(config-mdns-sl-out)# match APPLE-TV location-filter LOCAL-PROXY</pre>	<p>Associates the location-filter to one or more service-types to enable local proxy between local VLANs. For example, the Apple-TV learned from the YELLOW-VRF VLAN 201 and the GREEN-VRF VLAN 301 will be distributed to the receiver in the BLUE-VRF VLAN 101.</p> <p>The service-list contains implicit deny at the end.</p> <p>The message-type for outbound service-list is not required.</p>
Step 8	mdns-sd service-policy <i>service-policy-name</i> Example: <pre>Device(config)# mdns-sd service-policy BLUE-VRF-POLICY</pre>	Creates a unique mDNS service-policy.
Step 9	service-list <i>service-list-name</i> {in out} Example: <pre>Device(config)# mdns-sd service-policy BLUE-VRF-POLICY Device(config-mdns-ser-policy)# service-list BLUE-VRF-LIST-OUT out</pre>	Configures an mDNS service policy to associate with the service-list for each direction.
Step 10	vlan configuration <i>ID</i> Example: <pre>Device(config)# vlan configuration 101-103</pre>	<p>Enables VLAN configuration for advanced service parameters.</p> <p>One or more VLANs can be created for the same settings. For example, the VLAN</p>

	Command or Action	Purpose
		configuration range 101-110 or 200 allows to configure consecutive and nonconsecutive VLAN IDs.
Step 11	mdns-sd gateway Example: Device (config-vlan) # mdns-sd gateway	Enables the mDNS gateway on the specified VLAN IDs.
Step 12	service-policy BLUE-VRF-POLICY Example: Device (config-vlan-mdns) # service-policy BLUE-VRF-POLICY	Associates an mDNS service-policy with the specified VLAN IDs.
Step 13	end Example: Device (config-vlan-mdns) # end	Returns to privileged EXEC mode.

Verifying VRF-Aware Local Area Bonjour Services

The dynamically discovered VRF-Aware service information can be verified on Cisco Catalyst 9000 Series switch in SDG-Agent mode by including the **vrf** keyword on the existing **show mdns-sd** command. You can verify each VRF-service record information based on the unique VRF name.

The following is an example of the command that displays the dynamically discovered mDNS service records in the BLUE-VRF:

```
Device# show mdns-sd cache vrf BLUE-VRF
```

```

                                     mDNS CACHE
-----
[<NAME>]                               [<TYPE>]      [<TTL>/Remaining] [Vlan-Id/If-name]
[Mac Address]           [<RR Record Data>]

RTP-ATV-1._device-info._tcp.local      TXT           4500/4495        511
a018.28f2.9889          (13)'model=J33iAP'
RTP-ATV-1._airplay._tcp.local          PTR           4500/4495        511
a018.28f2.9889          RTP-ATV-1._airplay._tcp.local
RTP-ATV-1._raop._tcp.local             PTR           4500/4495        511
a018.28f2.9889          A01828F29889@RTP-ATV-1._raop._tcp.local
RTP-ATV-1._airplay._tcp.local          SRV           4500/4495        511
a018.28f2.9889          0             0             7000            RTP-ATV-3.local
A01828F29889@RTP-ATV-1._raop._tcp.local SRV           4500/4495        511
a018.28f2.9889          0             0             7000            RTP-ATV-3.local
RTP-ATV-1.local              AAAA          4500/4495        511
a018.28f2.9889          2001:10:153:2:C2F:9445:7062:5C3C
RTP-ATV-1.local              A             4500/4495        511
a018.28f2.9889          10.155.1.17
RTP-ATV-1._airplay._tcp.local          TXT           4500/4495        511
a018.28f2.9889
(208)'deviceid=A0:18:28:F2:98:89''features=0x5A7FFFFF7,0x1E''flags=0x44''model=~::~
A01828F29889@RTP-ATV-1._raop._tcp.local TXT           4500/4495        511

```

```
a018.28f2.9889
(177) 'cn=0,1,2,3'da=true'et=0,3,5'ft=0x5A7FFFF7,0x1E'md=0,1,2'am=AppleTV3,2'~
```

Use the following commands in privileged EXEC mode on a Cisco Catalyst 9000 Series switch configured in SDG-Agent mode to verify various Local Area Bonjour domain mDNS parameters such as service configuration, cache records, statistics, and so on.

Table 63: Commands to Verify VRF-Aware Services

Command	Purpose
show mdns-sd cache {all interface mac name service-peer static type vlan vrf}	<p>Displays all available mDNS cache record that supports multiple variables and provides granular source details. The following variables are available:</p> <ul style="list-style-type: none"> • all: Displays all available cache records discovered from multiple source connections of a system. • interface: Displays the available cache records discovered from a specified Layer 3 interface. • mac: Displays the available cache records discovered from the specified MAC address. • name: Displays the available cache records based on service provider announced name. • service-peer: Displays available cache records discovered from the specified Layer 2 Service-Peer. • static: Displays the locally configured static mDNS cache entries. • type: Displays the available cache records based on the specific mDNS record type (PTR, SRV, TXT, A, or AAAA). • vlan: Displays the available cache records discovered from the specified Layer 2 VLAN ID in unicast mode. • vrf: Displays each VRF available cache records based on the specific mDNS record type (PTR, SRV, TXT, A, or AAAA).
show mdns-sd service-definition {name type}	<p>Displays the built-in and user-defined custom service definitions and provides the mapping from service name to mDNS PTR records.</p> <p>The service-definition can be filtered by name or type.</p>

Command	Purpose
show mdns-sd service-list {direction name}	<p>Displays the configured inbound or outbound service-list that classifies matching service types for a service policy.</p> <p>The service lists can be filtered by name or specific direction.</p>
show mdns-sd service-policy {interface name}	<p>Displays the list of mDNS service policies mapped with inbound or outbound service-lists.</p> <p>The service policies list can be filtered by the associated specified interface or by name.</p>
show mdns-sd statistics {all cache debug interface service-list service-policy services vlan}	<p>Displays the detailed mDNS statistics processed bidirectionally by the system on each mDNS-gateway-enabled VLAN, when mDNS is configured in unicast mode.</p> <p>The keywords for the mDNS statistics provide a detail view on the interface, policy, service-list, and services.</p>
show mdns-sd summary {interface vlan}	<p>Displays the brief information about mDNS gateway and the key configuration status on all VLANs and interfaces of the system.</p>



PART II

Feature History for BGP EVPN VXLAN

- [Feature History for BGP EVPN VXLAN, on page 643](#)



CHAPTER 17

Feature History for BGP EVPN VXLAN

- [Feature History for BGP EVPN VXLAN, on page 643](#)

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). Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
	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. Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
	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. Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
	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. Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
	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. Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
	EVPN VXLAN Distributed Anycast Gateway	

Release	Feature	Feature Information
		<p>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.</p> <p>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.</p> <p>Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p>
	DHCP Relay for IPv4 Traffic in BGP EVPN VXLAN Fabric	<p>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.</p> <p>Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.</p>

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.10.1	EVPN VXLAN Overlay Network for IPv4 Bridged Traffic	Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
	EVPN VXLAN Overlay Network for IPv4 Routed Traffic	Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
	Layer 2 Broadcast, Unknown Unicast, and Multicast (BUM) Traffic Forwarding using Underlay Multicast	Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
	EVPN VXLAN Integrated Routing and Bridging	Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
	EVPN VXLAN Distributed Anycast Gateway	Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
	DHCP Relay for IPv4 Traffic in BGP EVPN VXLAN Fabric	Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
	Leaf Functionality	Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	EVPN VXLAN Overlay Network for IPv6 Bridged Traffic	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	EVPN VXLAN Overlay Network for IPv6 Routed Traffic	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Layer 2 Broadcast, Unknown Unicast, and Multicast (BUM) Traffic Forwarding using Ingress Replication	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>

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> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Border Leaf Functionality	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Autonomous System Number Rewrite	<p>The rewrite-evpn-rt-asn 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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	VRF-Lite Border Leaf Handoff	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	MPLS Layer 3 VPN Border Leaf Handoff	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	IEEE 802.1Q Border Leaf Handoff	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Access Border Leaf Handoff	

Release	Feature	Feature Information
		<p>Access border leaf handoff in a BGP EVPN VXLAN fabric allows Layer 2 external connectivity with an Access network through a border leaf switch.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	VPLS over MPLS Border Leaf Handoff	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.1.1	Spine Functionality	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Border Spine Functionality	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	ARP and IPv6 Neighbor Discovery Flooding Suppression	<p>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.</p> <p>Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.</p>
	Layer 3 Tenant Routed Multicast for IPv4 Traffic	<p>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.</p> <p>You can configure TRM with PIM sparse mode (PIM-SM) using anycast RP mode and TRM with PIM source specific mode (PIM-SSM).</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	VRF-Lite Border Spine Handoff	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	MPLS Layer 3 VPN Border Spine Handoff	<p>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.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>

Release	Feature	Feature Information
	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. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
	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. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
	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. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
	Configuring VXLAN-Aware Flexible Netflow	VXLAN-aware Flexible Netflow captures the VXLAN flow information for both bridged and routed traffic. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
	BGP EVPN VXLAN MIB support	Support was introduced for the MIB. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
Cisco IOS XE Amsterdam 17.2.1	ARP and IPv6 Neighbor Discovery Flooding Suppression	Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
	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. Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.3.1	Layer 3 Tenant Routed Multicast for IPv6 Traffic	<p>Layer 3 TRM for IPv6 traffic enables multicast forwarding for IPv6 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.</p> <p>Support was introduced to configure TRM with PIM-SSM and TRM with PIM-SM.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Enhanced RP Functionality for Layer 3 TRM for IPv4 and IPv6 traffic	<p>Enhancement was introduced for the configuration of an RP in the overlay network for TRM with PIM-SM. The enhancement allows you to configure an RP on a single or multiple VTEPs inside the BGP EVPN VXLAN fabric or on a device outside the fabric.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Interworking of Layer 3 TRM with MVPN Networks for IPv4 Traffic	<p>Interworking of Layer 3 TRM with MVPN Networks allows you to forward IPv4 Layer 3 multicast traffic between sources and receivers of an EVPN VXLAN network and an MVPN network.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Broadcast, Unknown Unicast, and Multicast Traffic Rate Limiting	<p>BUM Traffic Rate Limiting allows you to use a policer and set the flood rate limit of the BUM traffic in the network to a predefined value.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
Cisco IOS XE Amsterdam 17.3.2a	VNI Support Enhancement	<p>Support for up to 512 VNIs (Layer 2 and Layer 3 VNIs combined) per VTEP was introduced.</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>
	Route Type 2 (RT 2) to Route Type 5 (RT 5) Host Route Reorigination for Overlay Networks with a Centralized Gateway (Asymmetric IRB)	<p>RT 2 to RT 5 reorigination allows a Layer 2 and Layer 3 VXLAN network to import IP addresses from another Layer 2-only VXLAN network. A centralized gateway VTEP can reoriginate any RT 2 host routes, that it learns from the Layer 2 EVPN segment (or MAC VRF), as RT 5 routes in the Layer 3 EVPN segment (or IP VRF).</p> <p>Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.</p>

Release	Feature	Feature Information
Cisco IOS XE Bengaluru 17.6.1	Private VLANs (PVLANS) in a BGP EVPN VXLAN Fabric	BGP EVPN VXLAN allows you to extend a PVLAN across the VTEPs in an EVPN fabric overlay. The extension uses isolated and community VLANs to provide features like subnet management and sub segmentation of a broadcast domain in the fabric. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
	Route Leaking between EVPN and Global Routing Table	The route leaking feature provides the import and export of routes between an EVPN-enabled VRF and the Global Routing Table (GRT). By enabling the import and export of the EVPN learned routes into the GRT, the border node provides access to services such as DNS or DHCP to the hosts in the fabric. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
	EVPN VXLAN Multi-homing in Single-active Redundancy Mode	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. 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. 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. Support for this feature was introduced on all the models of the Cisco Catalyst 9500 Series Switches.
Cisco IOS XE Bengaluru 17.6.2	Data MDT Support for L3 TRM	Data MDTs are purpose built underlay MDTs to provide optimized forwarding in the MVPN and EVPN core.

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>.

