

EVPN Feature Guide for EX9200 Ethernet Switches



Modified: 2017-11-30

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

EVPN Feature Guide for EX9200 Ethernet Switches
Copyright © 2017 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	xi
	Documentation and Release Notes	xi
	Supported Platforms	xi
	Using the Examples in This Manual	xi
	Merging a Full Example	xii
	Merging a Snippet	xii
	Documentation Conventions	xiii
	Documentation Feedback	xv
	Requesting Technical Support	xv
	Self-Help Online Tools and Resources	xv
	Opening a Case with JTAC	xvi
Part 1	Understanding Ethernet VPNs	
Chapter 1	Overview	3
	EVPN Overview for Switches	3
	Understanding EVPN with VXLAN Data Plane Encapsulation	5
	Understanding EVPN	5
	Understanding VXLAN	8
	EVPN-VXLAN Integration Overview	8
	Understanding Contrail Virtual Networks Use with EVPN-VXLAN	10
	EVPN-VXLAN Support for VXLAN Underlay	11
	EVPN-VXLAN Packet Format	12
	Q & A	12
	EVPN-VXLAN Constraints on QFX Series Switches	13
	EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression	14
Chapter 2	EVPN Standards	17
	Supported EVPN Standards	17
Chapter 3	Distributing Routes in VPNs	19
	Configuring an IGP on the PE and P Routers on EX9200 Switches	19
	Configuring IBGP Sessions Between PE Routers in VPNs on EX9200 Switches	19
	Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches	21
	Using LDP for VPN Signaling	21

Part 2	Configuring EVPNs	
Chapter 4	Configuring Routing Instances for EVPN	25
	Configuring EVPN Routing Instances on EX9200 Switches	25
	Tracing EVPN Traffic and Operations	27
	Example: Configuring an EVPN Control Plane and VXLAN Data Plane	29
Chapter 5	Configuring Integrated Bridging and Routing Using EVPNs	99
	An EVPN with IRB Solution on EX9200 Switches Overview	99
	Need for an EVPN IRB Solution	100
	Implementing the EVPN IRB Solution	100
	Benefits of Implementing the EVPN IRB Solution	102
	Gateway MAC and IP Synchronization	102
	Layer 3 VPN Interworking	103
	IPv6 Support for IRB Interfaces with EVPN Using Neighborhood Discovery Protocol (NDP)	104
	Configuring an EVPN with IRB Solution on EX9200 Switches	104
	Example: Configuring an EVPN with IRB Solution on EX9200 Switches	106
Chapter 6	Configuring EVPN Multihoming	119
	EVPN Multihoming Overview	119
	Introduction to EVPN Multihoming	119
	Understanding EVPN Multihoming Concepts	121
	EVPN Multihoming Mode of Operation	123
	EVPN Multihoming Implementation	124
	New BGP NLRIs	124
	New Extended Communities	127
	New EVPN Route Types	129
	Update to the MAC Forwarding Table	130
	Traffic Flow	131
	Aliasing	133
	EVPN Active-Active Multihoming and Multichassis Link Aggregation	134
	EVPN Active-Active Multihoming and IRB	135
	Sample Configuration	135
	Designated Forwarder Election	136
	DF Election Roles	136
	DF Election as Per RFC 7432	136
	Preference-Based DF Election	138
	DF Election for Virtual Switch	140
	Handling Failover	140
	ESIs on Physical, Aggregated Ethernet, and Logical Interfaces	141
	Convergence in an EVPN Network	141
	Configuring EVPN Active-Standby Multihoming	144
	Example: Configuring EVPN Active-Active Multihoming	147
	Example: Configuring an ESI on a Logical Interface With EVPN Multihoming	195
Chapter 7	Configuring MAC Mobility	207
	Overview of MAC Mobility	207
	Changing Duplicate MAC Address Detection Settings	209

Chapter 8	Configuring EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG	211
	Understanding EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG	211
	BUM Traffic Handling	213
	Split Horizon	214
	MAC Learning	215
	Handling Down Link Between Cascade and Uplink Ports in Junos Fusion Enterprise	215
	Layer 3 Gateway Support	216
	Example: EVPN-MPLS Interworking With Junos Fusion Enterprise	216
	Example: EVPN-MPLS Interworking With an MC-LAG Topology	232
Part 3	Configuration Statements and Operational Commands	
Chapter 9	Configuration Statements	253
	bgp-peer	254
	decapsulate-inner-vlan	255
	duplicate-mac-detection	256
	encapsulate-inner-vlan	257
	encapsulation (Physical Interface)	258
	encapsulation vxlan	264
	esi	265
	evpn	267
	extended-vlan-list	269
	extended-vni-list	270
	global-mac-ip-limit	271
	global-mac-ip-table-aging-time	272
	ingress-node-replication (EVPN)	273
	instance-type	274
	interface (EVPN Routing Instances)	277
	interface (Routing Instances)	278
	interface-mac-ip-limit	279
	interface-mac-limit (VPLS)	280
	ip-prefix-routes	282
	l2-learning	285
	mac-statistics	286
	mac-ip-table-size	287
	mac-table-size	288
	mclag	289
	multicast-mode (EVPN)	290
	no-arp-suppression	291
	no-mac-learning	292
	packet-action	294
	routing-instances	297
	switch-options	298
	traceoptions (Protocols EVPN)	299
	vlan-id (routing instance)	301
	vlangs	302

Chapter 10

vni-options	303
vni	304
Operational Commands	305
clear evpn duplicate-mac-suppression	306
clear evpn nd-table	307
clear ethernet-switching evpn nd-statistics	308
clear ethernet-switching evpn nd-table	309
show ethernet-switching evpn nd-statistics	310
show evpn arp-table	312
show evpn database	315
show evpn flood	319
show evpn instance	320
show evpn ip-prefix-database	329
show evpn l3-context	336
show evpn mac-table	339
show evpn nd-table	340
show evpn peer-gateway-macs	343
show vlans evpn nd-table	344

List of Figures

Part 1	Understanding Ethernet VPNs	
Chapter 1	Overview	3
	Figure 1: EVPN Overview	6
	Figure 2: EVPN-VXLAN Integration Overview	10
	Figure 3: EVPN-VXLAN Packet Format	12
Part 2	Configuring EVPNs	
Chapter 5	Configuring Integrated Bridging and Routing Using EVPNs	99
	Figure 4: Inter-Subnet Traffic Forwarding	101
Chapter 6	Configuring EVPN Multihoming	119
	Figure 5: CE Device Multihomed to Two PE Routers	121
	Figure 6: Simple EVPN Topology	121
	Figure 7: Active-Active EVPN Multihoming	124
	Figure 8: EVPN Active-Active Multihoming Topology	149
	Figure 9: EVPN-MPLS Topology with Multihoming Active-Standby	196
	Figure 10: EVPN-MPLS Topology with Multihoming Active-Active	197
Chapter 7	Configuring MAC Mobility	207
	Figure 11: Mac Mobility in an EVPN Network	208
Chapter 8	Configuring EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG	211
	Figure 12: EVPN-MPLS Interworking with Junos Fusion Enterprise	212
	Figure 13: EVPN-MPLS Interworking with MC-LAG	212
	Figure 14: EVPN-MPLS Interworking with Junos Fusion Enterprise	218
	Figure 15: EVPN-MPLS Interworking With an MC-LAG Topology	233

List of Tables

	About the Documentation	xi
	Table 1: Notice Icons	xiii
	Table 2: Text and Syntax Conventions	xiv
Part 2	Configuring EVPNs	
Chapter 6	Configuring EVPN Multihoming	119
	Table 3: Priority for NLRI Route Type	141
	Table 4: EVPN Multihoming Active-Standby: Configuring the Connection with CE1 on PE1 and PE2	196
	Table 5: Multihoming Active-Standby: Configuring Logical Interface on PE1 that Provides Services to a Different VLAN	197
	Table 6: Multihoming Active-Active: Configuring the Connection with CE1 on PE1 and PE2	197
	Table 7: EVPN Multihoming Active-Active: Configuring Interface on PE1 that Provides Services to a Different VLAN	198
Chapter 7	Configuring MAC Mobility	207
	Table 8: MAC Advertisement Routes on PE Devices	208
Chapter 8	Configuring EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG	211
	Table 9: BUM Traffic: Issues and Resolutions	214
	Table 10: BUM Traffic: Split Horizon-Related Issue and Resolution	214
	Table 11: MAC Learning: EVPN and MC-LAG Synchronization Issue and Implementation Details	215
	Table 12: Key Junos Fusion Enterprise and EVPN (BGP and MPLS) Attributes Configured on PE1, PE2, and PE3	219
	Table 13: Key MC-LAG and EVPN (BGP and MPLS) Attributes Configured on PE1, PE2, and PE3	234
Part 3	Configuration Statements and Operational Commands	
Chapter 10	Operational Commands	305
	Table 14: show ethernet-switching evpn nd-statistics Output Fields	310
	Table 15: show evpn arp-table Output Fields	312
	Table 16: show evpn database Output Fields	316
	Table 17: show evpn instance Output Fields	321
	Table 18: show evpn ip-prefix-database Output Fields	330
	Table 19: show evpn l3-context Output Fields	336
	Table 20: show evpn nd-table Output Fields	340
	Table 21: show vlans evpn nd-table Output Fields	344

About the Documentation

- Documentation and Release Notes on page xi
- Supported Platforms on page xi
- Using the Examples in This Manual on page xi
- Documentation Conventions on page xiii
- Documentation Feedback on page xv
- Requesting Technical Support on page xv

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Supported Platforms

For the features described in this document, the following platforms are supported:

- EX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see [CLI Explorer](#).

Documentation Conventions

Table 1 on page xiii defines notice icons used in this guide.

Table 1: Notice Icons







Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xiv defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies guide names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS CLI User Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [community-ids]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

GUI Conventions

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <http://www.juniper.net/techpubs/feedback/>.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <https://prsearch.juniper.net/>
- Find product documentation: <http://www.juniper.net/documentation/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes:
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:
<http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum:
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://entitlementsearch.juniper.net/entitlementsearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Understanding Ethernet VPNs

- [Overview on page 3](#)
- [EVPN Standards on page 17](#)
- [Distributing Routes in VPNs on page 19](#)

CHAPTER 1

Overview

- [EVPN Overview for Switches on page 3](#)
- [Understanding EVPN with VXLAN Data Plane Encapsulation on page 5](#)
- [EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression on page 14](#)

EVPN Overview for Switches

An Ethernet VPN (EVPN) enables you to connect a group of dispersed customer sites using a Layer 2 virtual bridge. As with other types of VPNs, an EVPN comprises customer edge (CE) devices (host, router, or switch) connected to provider edge (PE) devices. The PE devices can include an MPLS edge switch (MES) that acts at the edge of the MPLS infrastructure. For the initial deployment of EVPNs using Juniper Networks equipment, you can configure an EX9200 switch to act as an MES. You can deploy multiple EVPNs within the network, each providing network connectivity to customers while ensuring that traffic sharing that network remains private.

The MESs are interconnected within the network by using label-switched paths (LSPs). The MPLS infrastructure allows you to take advantage of the MPLS functionality provided by the Junos operating system (Junos OS), including fast reroute, node and link protection, and standby secondary paths. For EVPNs, learning between MESs takes place in the control plane rather than in the data plane (as is the case with traditional network bridging). The control plane provides greater control over the learning process, allowing you to restrict which devices discover information about the network. You can also apply policies on the MESs, allowing you to carefully control how network information is distributed and processed. EVPNs utilize the BGP control plane infrastructure, providing greater scale and the ability to isolate groups of devices (hosts, servers, virtual machines, and so on) from each other.

The MESs attach an MPLS label to each MAC address learned from the CE devices. This label and MAC address combination is advertised to the other MESs in the control plane. Control plane learning enables load balancing and improves convergence times in the event of certain types of network failures. The learning process between the MESs and the CE devices is completed using the method best suited to each CE device (data plane learning, IEEE 802.1, LLDP, 802.1aq, and so on).

The policy attributes of an EVPN are similar to an IP VPN (for example, Layer 3 VPNs). Each EVPN routing instance requires that you configure a route distinguisher and one or

more route targets. A CE device attaches to an EVPN routing instance on an MES through an Ethernet interface that might be configured for one or more VLANs.

The following features are available for EVPNs:

- Ethernet connectivity between data centers spanning metropolitan area networks (MANs) and WANs
- One or more VLANs for each MAC VPN
- Automatic route distinguishers
- Dual-homed EVPN connection with active standby multihoming
- Starting with Junos OS Releases 16.1R4 and 16.2R2, the active-active mode for EVPN multihoming is supported.
- Starting with Junos OS Release 17.3R1, both pure type-5 routes and standard type-5 routes are supported on EX9200 switches. Use this feature, which advertises IP prefixes through EVPN, when the Layer 2 domain does not exist at the remote data centers or metro network peering points. For more information about how to configure, see [ip-prefix-routes](#).
- Starting with Junos OS OS Release 17.3R1, VXLAN encapsulation is supported. Previously, only MPLS encapsulation is supported.

The following features are not supported for EVPNs:

- Graceful restart, graceful Routing Engine switchover (GRES), and nonstop active routing (NSR)

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1, both pure type-5 routes and standard type-5 routes are supported on EX9200 switches.
17.3R1	Starting with Junos OS OS Release 17.3R1, VXLAN encapsulation is supported. Previously, only MPLS encapsulation is supported.
16.1R4	Starting with Junos OS Releases 16.1R4 and 16.2R2, the active-active mode for EVPN multihoming is supported.

Related Documentation

- [Supported EVPN Standards on page 17](#)
- [Example: Configuring EVPN Active-Active Multihoming on page 147](#)

Understanding EVPN with VXLAN Data Plane Encapsulation

Ethernet VPNs (EVPNs) enable you to connect groups of dispersed customer sites using Layer 2 virtual bridges, and Virtual Extensible LANs (VXLANs) allow you to stretch Layer 2 connectivity over an intervening Layer 3 network, while providing network segmentation like a VLAN, but without the scaling limitation of traditional VLANs. EVPN with VXLAN encapsulation handles Layer 2 connectivity at the scale required by cloud server providers and replaces limiting protocols like Spanning Tree Protocol (STP), freeing up your Layer 3 network to use more robust routing protocols. EVPN with VXLAN data plane encapsulation can be used with and without Juniper Networks Contrail virtualization software—use Contrail with EVPN VXLAN data plane encapsulation when you have an environment that includes both virtual and base-metal devices.



NOTE: MX Series Routers interoperate with both QFX5100 Series Switches and QFX5100 Virtual Chassis to provide full EVPN with VXLAN functionality. Junos OS Release 14.2R4 and Junos OS Release 16.1 support MX Series Routers, and Junos OS Release 14.1X53-D30 and Junos OS Release 16.1 support QFX5100 Switches.

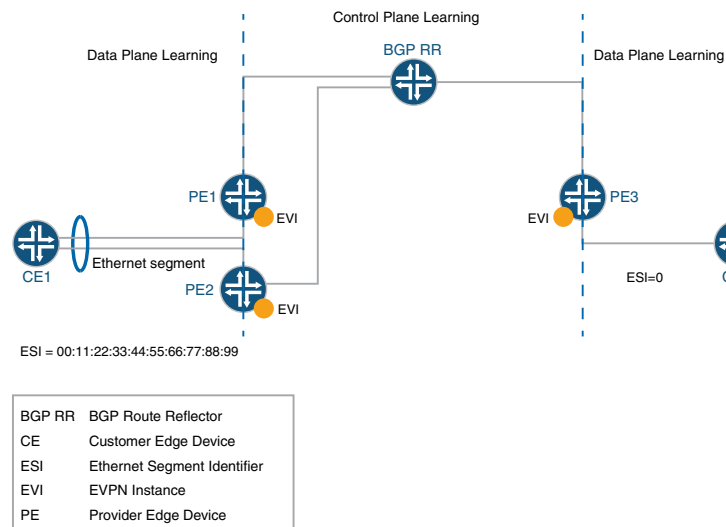
EX9200 switches are supported starting with Junos OS Release 17.3R1. Previously, EX9200 switches supported only MPLS encapsulation.

- [Understanding EVPN on page 5](#)
- [Understanding VXLAN on page 8](#)
- [EVPN-VXLAN Integration Overview on page 8](#)
- [Understanding Contrail Virtual Networks Use with EVPN-VXLAN on page 10](#)
- [EVPN-VXLAN Support for VXLAN Underlay on page 11](#)
- [EVPN-VXLAN Packet Format on page 12](#)
- [Q & A on page 12](#)
- [EVPN-VXLAN Constraints on QFX Series Switches on page 13](#)

Understanding EVPN

Ethernet VPN (EVPN) is a standards-based technology that provides virtual multipoint bridged connectivity between different Layer 2 domains over an IP or IP/MPLS backbone network. Like other VPN technologies, such as IP VPN and virtual private LAN service (VPLS), EVPN instances are configured on provider edge (PE) routers to maintain logical service separation between customers. The PE routers connect to customer edge (CE) devices, which can be routers, switches, or hosts. The PE routers then exchange reachability information using Multiprotocol BGP (MP-BGP) and encapsulated traffic is forwarded between PE routers. Because elements of the architecture are common with other VPN technologies, you can seamlessly introduce and integrate EVPN into existing service environments, as shown in [Figure 1 on page 6](#).

Figure 1: EVPN Overview



The EVPN is used as a Layer 2 overlay solution to provide Layer 2 connection over an IP underlay for the endpoints within a virtual network whenever Layer 2 connectivity is required by an end station such as bare-metal server (BMS). Otherwise, Layer 3 routing is used through VRF tables between Contrail vRouters and MX Series routers. EVPN technology offers multitenancy, flexible services that can be extended on demand, frequently using compute resources of different physical data centers for a single service (Layer 2 extension).

EVPN's MP-BGP control plane enables you to dynamically move live virtual machines from one data center to another, also known as virtual machine (VM) motion. After you move a VM to a destination server or hypervisor, it transmits a gratuitous ARP, which updates the Layer 2 forwarding table of the PE device at the destination data center. The PE device then transmits a MAC route update to all remote PE devices which in turn update their forwarding tables. An EVPN tracks the movement of the VM, which is also known as MAC mobility.

EVPN also has mechanisms that detect and stop MAC flapping, and prevent the looping of broadcast, unknown unicast, and multicast (BUM) traffic in an all-active multi-homed topology.

The EVPN technology, similar to Layer 3 MPLS VPN, includes the concept of routing MAC addresses using IP/MPLS core. EVPN provides the following benefits:

- Ability to have an active multihomed edge device
- Aliasing
- Fast convergence
- Load balancing across dual-active links

- MAC address mobility
- Multitenancy

In addition, EVPN uses these techniques:

- *Multihoming* provides redundancy in the event that an access link or one of the PE routing devices fails. In either case, traffic flows from the CE device towards the PE router, using the remaining active links. For traffic in the other direction, the remote PE router updates its forwarding table to send traffic to the remaining active PE routers connected to the multihomed Ethernet segment. EVPN provides a fast convergence mechanism, which reduces traffic restoration time so that the time it takes to make this adjustment is independent of the number of media access control (MAC) addresses learned by the PE router. All-active multihoming enables a CE device to connect to two or more PE routers such that traffic is forwarded using all of the links between the devices. This multihoming enables the CE device to load-balance traffic to multiple PE routers. More importantly, multihoming enables a remote PE router to load-balance traffic to the multihomed PE routers across the core network. This load balancing of traffic flows between data centers is known as *aliasing*, which causes different signals to become indistinguishable—they become aliases of one another. Aliasing is used with digital audio and digital images.
- *Split horizon* prevents the looping of broadcast, unknown unicast, and multicast (BUM) traffic in a network. The split horizon basic principle is simple: Information about the routing for a particular packet is never sent back in the direction from which it was received.
- *Local link bias* conserves bandwidth by using local links to forward unicast traffic exiting a Virtual Chassis or Virtual Chassis Fabric (VCF) that has a link aggregation group (LAG) bundle composed of member links on different member switches in the same Virtual Chassis or VCF. A local link is a member link in the LAG bundle that is on the member switch that received the traffic.
- EVPN with VXLAN encapsulation is used for Layer 2 connectivity between virtual machines and a QFX5100 switch within a Layer 2 domain where the QFX5100 switch is physically connected to top-of-rack.

For Layer 2 connectivity, traffic load balance in the core is achieved using the all-active multihome feature provided by EVPN. Each QFX5100 switch is also multihomed to the top-of-rack switches.

You can use QFX5100 switches to provision for Layer 2 and Layer 3 gateways in a Contrail user interface by using NETCONF. However, on MX series or QFX series devices VTEP services is also provided when needed and configured as such. The NETCONF XML management protocol is an XML-based protocol that client applications use to request and change configuration information on routing, switching, and security devices. The NETCONF XML management protocol uses an XML based data encoding for the configuration data and remote procedure calls. The NETCONF protocol defines basic operations that are equivalent to configuration mode commands in the command-line interface (CLI). Applications use the protocol operations to display, edit, and commit configuration statements similarly to how administrators use CLI configuration mode commands to perform those same operations.

Understanding VXLAN

Virtual extensible LANs (VXLANs) introduced an overlay scheme that expands the Layer 2 network address space from 4K to 16 million, largely solving the scaling issues seen in VLAN-based environments.

Network overlays are created by encapsulating traffic and tunneling the traffic over a physical network. You can use a number of tunneling protocols in the data center to create network overlays—the most common protocol is VXLAN. VXLAN tunneling protocol encapsulates Layer 2 Ethernet frames in Layer 3 UDP packets. This encapsulation enables you to create virtual Layer 2 subnets or segments that can span physical Layer 3 networks.

In a VXLAN overlay network, a VXLAN network identifier (VNI) uniquely identifies each Layer 2 subnet or segment. A VNI segments traffic the same way that an IEEE 802.1Q VLAN ID segments traffic. As is the case with VLAN, virtual machines on the same VNI can communicate directly with each other, whereas virtual machines on different VNIs need a router to communicate with each other.

The entity that performs the encapsulation and decapsulation is called a VXLAN tunnel endpoint (VTEP). In the virtual network, VTEPs typically reside in hypervisor hosts, such as kernel-based virtual machine (KVM) hosts.

Each VTEP has two interfaces.

- One interface is a switching interface that faces the virtual machines in the host and provides communication between VMs on the local LAN segment.
- The other is an IP interface that faces the Layer 3 network.

Each VTEP has a unique IP address that is used for routing the UDP packets between VTEPs. For example, when VTEP1 receives an Ethernet frame from VM1 addressed to VM3, it uses the VNI and the destination MAC to look up in its forwarding table, which VTEP sends the packet to. It then adds a VXLAN header that contains the VNI to the Ethernet frame and encapsulates the frame in a Layer 3 UDP packet and routes the packet to VTEP2 over the Layer 3 network. VTEP2 de-encapsulates the original Ethernet frame and forwards it to VM3. VM1 and VM3 cannot detect the VXLAN tunnel and the Layer 3 network between them.

EVPN-VXLAN Integration Overview

VXLAN defines a tunneling scheme to overlay Layer 2 networks on top of Layer 3 networks. This tunneling scheme allows for optimal forwarding of Ethernet frames with support for multipathing of unicast and multicast traffic with the use of UDP/IP encapsulation for tunneling, and is mainly used for the intra-data center site connectivity.

A unique characteristic of EVPN is that MAC address learning between PE routers occurs in the control plane. The local PE router detects a new MAC address from a CE device and then, using MP-BGP, advertises the address to all the remote PE routers. This method differs from existing Layer 2 VPN solutions such as VPLS, which learn by flooding unknown unicast in the data plane. This control plane MAC learning method is the key enabler of the many useful features that EVPN provides.

Because MAC learning is handled in the control plane, EVPN has the flexibility to support different data plane encapsulation technologies between PE routers. This flexibility is important because not every backbone network might be running MPLS, especially in enterprise networks.

EVPN addresses many of the challenges faced by network operators building data centers to offer cloud and virtualization services. The main application of EVPN is Data Center Interconnect (DCI), the ability to extend Layer 2 connectivity between different data centers that are deployed to improve the performance of delivering application traffic to end users and for disaster recovery.

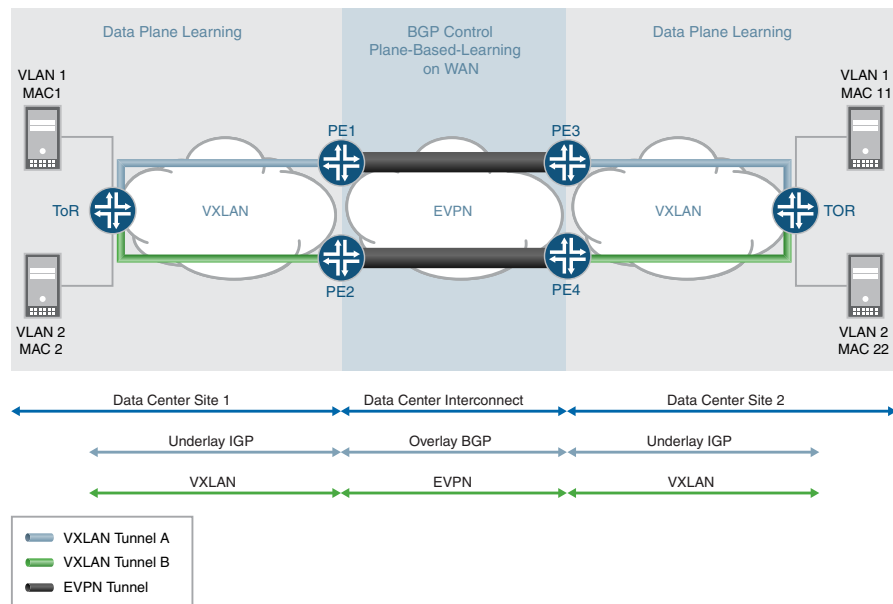
Although various DCI technologies are available, EVPN has an advantage over the other MPLS technologies because of its unique features, such as active/active redundancy, aliasing, and mass MAC withdrawal. As a result, to provide a solution for DCI, VXLAN is integrated with EVPN.

As shown in [“Overview” on page 3](#), every VXLAN, which is connected to the MPLS or IP core, runs an independent instance of the interior gateway protocol (IGP) control plane. Each PE router participates in the IGP control plane instance of its VXLAN. Each customer is a data center, so each has its own virtual router for VXLAN underlay.

Each PE node can terminate the VXLAN data plane encapsulation where the VXLAN network identifier (VNI) is mapped to a bridge domain. The PE router performs data plane learning on the traffic received from the VXLAN.

Each PE node implements EVPN to distribute the client MAC addresses learned over the VXLAN tunnel into BGP. Each PE node encapsulates the VXLAN or Ethernet frames with MPLS when sending the packets over the MPLS core and with the VXLAN tunnel header when sending the packets over the VXLAN network.

Figure 2: EVPN-VXLAN Integration Overview



- Every VXLAN network, which is connected to the MPLS or IP core, runs an independent instance of the IGP control plane. Each PE device participates in the IGP control plane instance of its VXLAN network, where each customer is a data center and has its own virtual router for the VXLAN underlay.
- Each PE node may terminate the VXLAN data plane encapsulation, where each VXLAN network identifier (VNI) is mapped to a bridge domain. The PE router performs data plane learning on the traffic received from the VXLAN network.
- Each PE node implements EVPN to distribute the client MAC addresses learned over the VXLAN tunnel into the BGP.
- Each PE node encapsulates the VXLAN or Ethernet frames with MPLS when sending the packets over the MPLS core and with the VXLAN tunnel header when sending the packets over the VXLAN network.

Understanding Contrail Virtual Networks Use with EVPN-VXLAN

Juniper Networks Contrail virtualization software is a software-defined networking (SDN) solution that automates and orchestrates the creation of highly scalable virtual networks. These virtual networks enable you to harness the power of the cloud—for new services, increased business agility, and revenue growth. MX Series routers can use EVPN-VXLAN to provide both Layer 2 and Layer 3 connectivity for end stations within a Contrail virtual network (VN).



NOTE: Contrail does not support the provisioning of Layer 2 and Layer 3 gateways on QFX Series switches.

The Contrail software for virtual networks provides both Layer 2 and Layer 3 connectivity. With Contrail, Layer 3 routing is preferred over Layer 2 bridging whenever possible. Layer 3 routing is used through virtual routing and forwarding (VRF) tables between Contrail vRouters and physical MX Series routers. MX Series routers provide Layer 3 gateway functionality between virtual networks.

Contrail enables you to use EVPN-VXLAN when your network includes both virtual and bare-metal devices.

Two types of encapsulation methods are used in virtual networks.

- MPLS-over-GRE (generic routing encapsulation) is used for Layer 3 routing between Contrail and MX Series routers.
- EVPN-VXLAN is used for Layer 2 connectivity between virtual machines and QFX Series switches within a Layer 2 domain where the QFX Series switch is physically connected to top-of-rack switches. For Layer 2 connectivity, traffic load balancing in the core is achieved using the multihoming all-active feature provided by EVPN. Each QFX switch is also multihomed to top-of-rack switches. Starting with Junos OS Release 17.3R1, EX9200 switches also support EVPN-VXLAN with Contrail.



NOTE: MPLS core is not supported on switches—only MX Series routers support this feature.

You cannot simultaneously mix EVPN-VXLAN with Open vSwitch Database (OVSDB)-VXLAN on QFX Series switches. After a switch is set to OVSDB-managed, the controller treats all ports as managed by OVSDB.

EVPN-VXLAN Support for VXLAN Underlay

Starting in Junos OS Release 14.1, MX Series routers support Virtual Extensible LAN (VXLAN) gateways. Each VXLAN gateway supports the following functionalities:

- 32,000 VXLANs with one VXLAN per bridge domain
- 8,000 VXLAN tunnel endpoints (VTEPs)
- 32,000 multicast groups
- Switching functionality with traditional Layer 2 networks and VPLS networks
- Inter-VXLAN routing and VXLAN-only bridging domain with IRB
- Virtual switches
- VXLAN with VRF functionality
- Configurable load balancing
- Statistics for remote VTEP

Starting in Junos OS Release 17.3, support is extended to VXLAN gateway implementation using an IPv6 underlay.

The following service types are supported with the IPv6 underlay support:

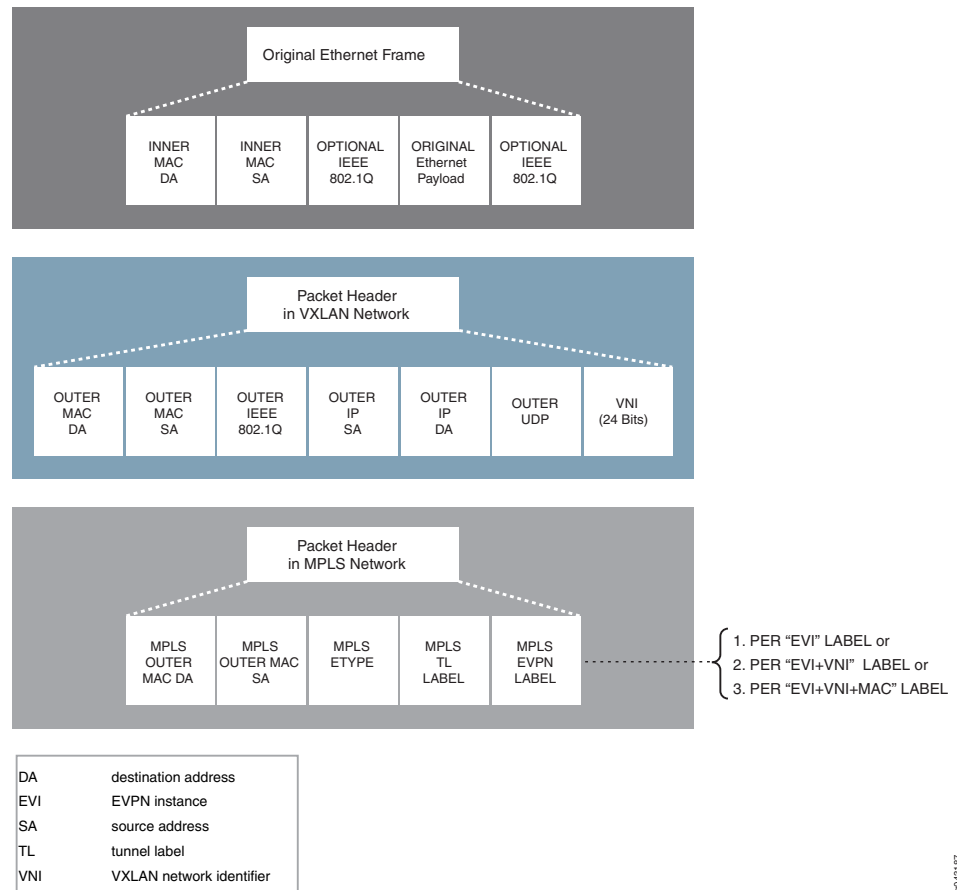
- VLAN-based service
- VLAN-bundle service
- Port-based service
- VLAN-aware service

Both IPv4 and IPv6 EVPN-VXLAN underlays support the Type 2 MAC address with IP address advertisement and the proxy MAC address with IP address advertisement.

EVPN-VXLAN Packet Format

The EVPN-VXLAN packet format is shown in [Figure 3 on page 12](#).

Figure 3: EVPN-VXLAN Packet Format



Q & A

Q: What is the difference between EVPN/VXLAN and MPLS/EVPN?

A: With EVPN/VxLAN the host entries are created by L2 and there is no existing infrastructure method to export to L3. In the MPLS/EVPN solution host entries are created

by RPD, are in the RIB and can be exported into a VRF or inet.0. Therefore, the ability to advertise /32 host routes is not there with EVPN/VxLAN, so the classic VMTO does not work with VxLAN. You can take advantage of the A/A L2 functionality with VxLAN.

Q: Can I simultaneously use EVPN-VXLAN with OVSDb-VXLAN on QFX switches?

A: No, you cannot mix EVPN-VXLAN with OVSDb-VXLAN. Once a switch is set to OVSDb-managed, the controller expects all ports to be OVSDb managed.



NOTE: MPLS core is not supported on switches—only MX Series routers support this feature.

EVPN-VXLAN Constraints on QFX Series Switches

QFX Series switches do not support the following features in an EVPN-VXLAN overlay network:

- Connectivity fault management (CFM)
- Internet Group Management Protocol (IGMP) snooping
- Multichassis link aggregation groups (MC-LAGs)
- Q-in-Q tunneling
- Spanning Tree Protocol (STP) (any variant)

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1, EX9200 switches also support EVPN-VXLAN with Contrail.

Related Documentation

- [EVPN Multihoming Overview on page 119](#)
- [Understanding EVPN Pure Route Type-5 on QFX Switches](#)
- [Understanding VXLANs](#)
- [EVPN-over-VXLAN Supported Functionality](#)
- [EVPN Over VXLAN Encapsulation Configuration Overview for QFX5100 Series](#)

EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression

Starting with Junos OS Release 17.2R1, MX Series routers and EX9200 that function as provider edge (PE) devices in an Ethernet VPN-MPLS (EVPN-MPLS) or Ethernet VPN-Virtual Extensible LAN (EVPN-VXLAN) environment support proxy Address Resolution Protocol (ARP) and ARP suppression and Network Discovery Protocol (NDP) and NDP suppression. Starting with Junos OS Release 17.3R1, QFX10000 Series switches that function as provider edge (PE) devices in an Ethernet VPN-Virtual Extensible LAN (EVPN-VXLAN) environment support proxy Address Resolution Protocol (ARP) and ARP suppression and Network Discovery Protocol (NDP) and NDP suppression. This feature reduces the flooding of ARP and NDP messages in the EVPN network, resulting in a more efficient use of core bandwidth.

By default, proxy ARP and ARP suppression and NDP and NDP suppression are enabled. For these features to work properly In Junos OS Releases 17.2 and 17.3, the configuration of an integrated and routing (IRB) interface on the PE device is required. Starting with Junos OS Release 17.4R1, proxy ARP and ARP suppression and NDP suppression are also supported on non-IRB interfaces. Junos OS 17.4R1 also introduces the ability to limit the number of media-access-control (MAC)-IP address bindings that can be learned. This limit can be configured globally or for a specific routing instance, bridge domain (MX routers), VLAN (EX9200 switches), or interface. After the specified limit is reached, no additional entries are added to the MAC-IP binding database. You can also specify a timeout interval for MAC-IP address bindings.

Proxy ARP and Network Discovery Protocol snooping is enabled by default for all EVPN-MPLS or EVPN-VXLAN bridge domains and VLANs. ARP or Network Discovery Protocol (NDP) packets generated from a local CE are snooped. ARP and NDP packets generated from a remote PE through core-facing interfaces, however, are not snooped.

Both IRB and non-IRB interfaces configured on a PE device deliver ARP requests and NDP requests from both local (CE) devices. When a PE device receives an ARP request or NDP request from a CE device, the PE device searches its MAC-IP address bindings database for the requested IP address. If the PE device finds the MAC-IP address binding in its database, it responds to the request. If the device does not find the MAC-IP address binding, it swaps the source MAC address in the with the MAC address of the IRB interface on which the request was received and sends the request to all interfaces.

Even when a PE device responds to an ARP request or NDP request, ARP packets and NDP might still be flooded across the WAN. ARP suppression and NDP suppression prevent this flooding from occurring.

You can disable the suppression of ARP packets and NDP packets by specifying the **no-arp-suppression** configuration statement. However, if you do so, be aware of the following implications:

- ARP and NDP packets always get flooded.
- The PE does not respond to ARP or NDP requests.
- Then PE does not learn the IP address from the ARP or NDP request.

Therefore, we recommend that ARP suppression and NDP suppression remain enabled.

Proxy ARP and ARP suppression and NDP and NDP suppression are supported in the following scenarios:

- single-homed devices in active mode—EVPN-MPLS and EVPN-VXLAN
- multihomed devices in active-active mode—EVPN-MPLS and EVPN-VXLAN
- multihomed devices in single-active mode—EVPN-MPLS only

In a multihoming active-active situation, the database of MAC-IP address bindings are synchronized between the PE devices that act as designated forwarder (DF) and non-designated forwarder (non-DF).

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, proxy ARP and ARP suppression and NDP suppression are also supported on non-IRB interfaces.
17.4R1	Junos OS 17.4R1 also introduces the ability to limit the number of media-access-control (MAC)-IP address bindings that can be learned.
17.3R1	Starting with Junos OS Release 17.3R1, QFX10000 Series switches that function as provider edge (PE) devices in an Ethernet VPN-Virtual Extensible LAN (EVPN-VXLAN) environment support proxy Address Resolution Protocol (ARP) and ARP suppression and Network Discovery Protocol (NDP) and NDP suppression.
17.2R1	Starting with Junos OS Release 17.2R1, MX Series routers and EX9200 that function as provider edge (PE) devices in an Ethernet VPN-MPLS (EVPN-MPLS) or Ethernet VPN-Virtual Extensible LAN (EVPN-VXLAN) environment support proxy Address Resolution Protocol (ARP) and ARP suppression and Network Discovery Protocol (NDP) and NDP suppression.

Related Documentation

- [global-mac-ip-limit on page 271](#)
- [interface-mac-ip-limit on page 279](#)
- [mac-ip-table-size on page 287](#)
- [no-arp-suppression on page 291](#)

CHAPTER 2

EVPN Standards

- [Supported EVPN Standards on page 17](#)

Supported EVPN Standards

Junos OS supports the following RFCs and Internet drafts that define standards for EVPNs:

- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*
- RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*
- RFC 7432, *BGP MPLS-Based Ethernet VPN*

The following features are not supported:

- Automatic derivation of Ethernet segment (ES) values. Only static ES configurations are supported.
- Host proxy ARP.
- Internet draft draft-ietf-spring-segment-routing-13, *Segment Routing Architecture*
- Internet draft draft-ietf-spring-segment-routing-mpls-11, *Segment Routing with MPLS data plane*
- Internet draft draft-ietf-isis-segment-routing-extensions-13, *IS-IS Extensions for Segment Routing*

Related Documentation

- [EVPN Overview](#)
- [Accessing Standards Documents on the Internet](#)

CHAPTER 3

Distributing Routes in VPNs

- [Configuring an IGP on the PE and P Routers on EX9200 Switches on page 19](#)
- [Configuring IBGP Sessions Between PE Routers in VPNs on EX9200 Switches on page 19](#)
- [Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches on page 21](#)

Configuring an IGP on the PE and P Routers on EX9200 Switches

For Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, VPLS, EVPNs, and Layer 2 circuits to function properly, the service provider's PE and P routers must be able to exchange routing information. For this to happen, you must configure either an IGP (such as OSPF or IS-IS) or static routes on these routers. You configure the IGP on the master instance of the routing protocol process at the **[edit protocols]** hierarchy level, not within the routing instance used for the VPN—that is, not at the **[edit routing-instances]** hierarchy level.

When you configure the PE router, do not configure any summarization of the PE router's loopback addresses at the area boundary. Each PE router's loopback address should appear as a separate route.

- Related Documentation**
- [Understanding IS-IS Configuration](#)
 - [Example: Configuring IS-IS](#)
 - [OSPF Feature Guide](#)

Configuring IBGP Sessions Between PE Routers in VPNs on EX9200 Switches

You must configure an IBGP session between the PE routers to allow the PE routers to exchange information about routes originating and terminating in the VPN. The PE routers rely on this information to determine which labels to use for traffic destined for remote sites.

Configure an IBGP session for the VPN as follows:

```
[edit protocols]
bgp {
  group group-name {
```

```
type internal;
local-address ip-address;
family evpn {
    signaling;
}
family (inet-vpn | inet6-vpn) {
    unicast;
}
family l2vpn {
    signaling;
}
neighbor ip-address;
}
```

The IP address in the **local-address** statement is the address of the loopback interface on the local PE router. The IBGP session for the VPN runs through the loopback address. (You must also configure the loopback interface at the **[edit interfaces]** hierarchy level.)

The IP address in the **neighbor** statement is the loopback address of the neighboring PE router.

The **family** statement allows you to configure the IBGP session for Layer 2 VPNs, VPLS, EVPNs or for Layer 3 VPNs.

- To configure an IBGP session for Layer 2 VPNs and VPLS, include the **signaling** statement at the **[edit protocols bgp group *group-name* family l2vpn]** hierarchy level:

```
[edit protocols bgp group group-name family l2vpn]
signaling;
```

- To configure an IBGP session for EVPNs, include the **signaling** statement at the **[edit protocols bgp group *group-name* family evpn]** hierarchy level:

```
[edit protocols bgp group group-name family evpn]
signaling;
```

- To configure an IPv4 IBGP session for Layer 3 VPNs, configure the **unicast** statement at the **[edit protocols bgp group *group-name* family inet-vpn]** hierarchy level:

```
[edit protocols bgp group group-name family inet-vpn]
unicast;
```

- To configure an IPv6 IBGP session for Layer 3 VPNs, configure the **unicast** statement at the **[edit protocols bgp group *group-name* family inet6-vpn]** hierarchy level:

```
[edit protocols bgp group group-name family inet6-vpn]
unicast;
```



NOTE: You can configure both **family inet** and **family inet-vpn** or both **family inet6** and **family inet6-vpn** within the same peer group. This allows you to enable support for both IPv4 and IPv4 VPN routes or both IPv6 and IPv6 VPN routes within the same peer group.

- Related Documentation**
- [Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches on page 21](#)

Configuring a Signaling Protocol and LSPs for VPNs on EX9200 Switches

For VPNs to function, you must enable the LDP signaling protocol on the provider edge (PE) routers and on the provider (P) routers.

To enable the LDP signaling protocol, perform the steps in the following section:

- [Using LDP for VPN Signaling on page 21](#)

Using LDP for VPN Signaling

To use LDP for VPN signaling, perform the following steps on the PE and provider (P) routers:

1. Configure LDP on the interfaces in the core of the network by including the **ldp** statement at the **[edit protocols]** hierarchy level.

You need to configure LDP only on the interfaces between PE routers or between PE and P routers. You can think of these as the “core-facing” interfaces. You do not need to configure LDP on the interface between the PE and customer edge (CE) routers.

```
[edit]
protocols {
  ldp {
    interface type-fpc/pic/port;
  }
}
```

2. Configure the MPLS address family on the interfaces on which you enabled LDP (the interfaces you configured in Step 1) by including the **family mpls** statement at the **[edit interfaces *type-fpc/pic/port* unit *logical-unit-number*]** hierarchy level.

```
[edit]
interfaces {
  type-fpc/pic/port {
    unit logical-unit-number {
      family mpls;
    }
  }
}
```

3. Configure OSPF or IS-IS on each PE and P router.

You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

- To configure OSPF, include the **ospf** statement at the **[edit protocols]** hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

```
[edit]
```

```
protocols {
  ospf {
    area 0.0.0.0 {
      interface type-fpc/pic/port;
    }
  }
}
```

- To configure IS-IS, include the **isis** statement at the **[edit protocols]** hierarchy level and configure the loopback interface and International Organization for Standardization (ISO) family at the **[edit interfaces]** hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, lo0), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the **address** statement, **address** is the NET.

```
[edit]
interfaces {
  lo0 {
    unit logical-unit-number {
      family iso {
        address address;
      }
    }
  }
  type-fpc/pic/port {
    unit logical-unit-number {
      family iso;
    }
  }
}
protocols {
  isis {
    interface all;
  }
}
```

For more information about configuring OSPF and IS-IS, see the *OSPF Feature Guide* and *IS-IS Feature Guide*.

Related Documentation

- [EVPN Overview for Switches on page 3](#)

PART 2

Configuring EVPNs

- [Configuring Routing Instances for EVPN on page 25](#)
- [Configuring Integrated Bridging and Routing Using EVPNs on page 99](#)
- [Configuring EVPN Multihoming on page 119](#)
- [Configuring MAC Mobility on page 207](#)
- [Configuring EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG on page 211](#)

Configuring Routing Instances for EVPN

- [Configuring EVPN Routing Instances on EX9200 Switches on page 25](#)
- [Tracing EVPN Traffic and Operations on page 27](#)
- [Example: Configuring an EVPN Control Plane and VXLAN Data Plane on page 29](#)

Configuring EVPN Routing Instances on EX9200 Switches

To configure an EVPN routing instance, complete the following configuration on the PE router (or on the MPLS edge switch) within the EVPN service provider's network:

1. Configure the EVPN routing instance name using the **routing-instances** statement at the **[edit]** hierarchy level:

```
routing-instances routing-instance-name {...}
```

2. Configure the **evpn** option for the **instance-type** statement at the **[edit routing-instances routing-instance-name]** hierarchy level:

```
instance-type evpn;
```

3. Configure the interfaces for handling EVPN traffic between the MES and the CE device using the **interface** statement at the **[edit routing-instances routing-instance-name]** hierarchy level:

```
interface interface-name;
```

4. Configure a VLAN identifier for the EVPN routing instance using the **vlan-id** statement at the **[edit routing-instances routing-instance-name]** hierarchy level:

```
vlan-id (vlan-id | all | none);
```

5. Configure a route distinguisher on a PE router by including the **route-distinguisher** statement:

```
route-distinguisher (as-number:number | ip-address:number);
```

Each routing instance that you configure on a PE router must have a unique route distinguisher associated with it. VPN routing instances need a route distinguisher to help BGP to distinguish between potentially identical network layer reachability

information (NLRI) messages received from different VPNs. If you configure different VPN routing instances with the same route distinguisher, the commit fails.

For a list of the hierarchy levels at which you can include this statement, see the statement summary for this statement.

The route distinguisher is a 6-byte value that you can specify in one of the following formats:

- **as-number:number**, where **as-number** is an autonomous system (AS) number (a 2-byte value) and **number** is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number.
 - **ip-address:number**, where **ip-address** is an IP address (a 4-byte value) and **number** is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the **router-id** statement, which is a nonprivate address in your assigned prefix range.
6. Configure either import and export policies for the EVPN routing table, or configure the default policies using the **vrf-target** statement configured at the **[edit routing-instances routing-instance-name]** hierarchy level.

See *Configuring Policies for the VRF Table on PE Routers in VPNs*.

7. Configure each EVPN interface for the EVPN routing instance:
- a. Configure interface encapsulation for the CE facing interfaces at the **[edit interfaces interface-name encapsulation]** hierarchy level. . Supported encapsulations for EX9200 switches are: (**extended-vlan-bridge** | **flexible-ethernet-services** | **vlan-bridge**).
 - b. Configure **vlan-bridge** encapsulation on the logical interface at the **[edit interfaces interface-name flexible-vlan-tagging encapsulation flexible-ethernet-services unit 0 encapsulation]** hierarchy level.
 - c. (Optional) Allow the EVPN to establish a connection to the CE device even if the CE device interface encapsulation and the EVPN interface encapsulations do not match by including the **ignore-encapsulation-mismatch** statement at the **[edit routing-instances routing-instance-name protocols evpn interface interface-name]** hierarchy level.
8. Specify the maximum number of media access control (MAC) addresses that can be learned by the EVPN routing instance by including the **interface-mac-limit** statement.

You can configure the same limit for all interfaces configured for a routing instance by including this statement at the **[edit routing-instances routing-instance-name protocols evpn]** hierarchy level. You can also configure a limit for a specific interface by including this statement at the **[edit routing-instances routing-instance-name protocols evpn interface interface-name]** hierarchy level.

By default, packets with new source MAC addresses are forwarded after the MAC address limit is reached. You can alter this behavior by including the **packet-action**

drop statement at either the `[edit routing-instances routing-instance-name protocols evpn interface-mac-limit]` or the `[edit routing-instances routing-instance-name protocols evpn interface interface-name]` hierarchy level. If you configure this statement, packets from new source MAC addresses are dropped once the configured MAC address limit is reached.

9. Enable MAC accounting for the EVPN by including the **mac-statistics** statement at the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level.
10. Specify the number of addresses that can be stored in the MAC routing table using the **mac-table-size** statement at the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level.

You can optionally configure the **packet-action drop** option to specify that packets for new source MAC addresses be dropped once the MAC address limit is reached. If you do not configure this option, packets for new source MAC addresses are forwarded.

11. Disable MAC learning by including the **no-mac-learning** statement at either the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level to apply this behavior to all of the devices configured for an EVPN routing instance or at the `[edit routing-instances routing-instance-name protocols evpn interface interface-name]` hierarchy level to apply this behavior to just one of the CE devices.

Related Documentation

- [Configuring Policies for the VRF Table on PE Routers in VPNs](#)
- [Configuring Routing Instances on PE Routers in VPNs](#)
- [Tracing EVPN Traffic and Operations on page 27](#)

Tracing EVPN Traffic and Operations

To configure the EVPN routing instance to trace a variety of different parameters related to EVPN operation:

1. Specify the name of one or more EVPN trace files using the **file** option for the **traceoptions** statement at the `[edit routing-instances routing-instance-name protocols evpn]` hierarchy level:

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
}
```

The **file** option includes the following sub-options:

- ***filename***—Specify the name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory `/var/log`.
- ***files number***—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum **size**, it is renamed **trace-file.0**, then **trace-file.1**, and

so on, until the specified maximum **number** of trace files specified is reached. Then the oldest trace file is overwritten.

- **size size**—(Optional) Maximum size of each trace file. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.
- **world-readable | no-world-readable**—(Optional) Enable unrestricted file access or restrict file access to the user who created the file.

2. Specify the **flag** option for the **traceoptions** statement:

```
traceoptions {  
    flag flag <flag-modifier> <disable>;  
}
```

The **flag** option allows you to specify the scope of the trace by including one of the following sub-options:

- **all**—All EVPN tracing options
- **error**—Error conditions
- **general**—General events
- **mac-database**—MAC route database in the EVPN routing instance
- **nlri**—EVPN advertisements received or sent by means of the BGP
- **normal**—Normal events
- **oam**—OAM messages
- **policy**—Policy processing
- **route**—Routing information
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing
- **topology**—EVPN topology changes caused by reconfiguration or advertisements received from other PE routers using BGP

You can also specify one of the following modifiers for any of the traceoptions flags:

- **detail**—Provide detailed trace information.
- **disable**—Disable this trace flag.
- **receive**—Trace received packets.
- **send**—Trace sent packets.

Related Documentation

- *Configuring EVPN Routing Instances*
- [traceoptions on page 299](#)

Example: Configuring an EVPN Control Plane and VXLAN Data Plane

This example shows how to configure EVPN and VXLAN on a network to support Data Center Interconnect (DCI), allow for optimal forwarding of Ethernet frames, provide network segmentation on a broad scale, enable control plane-based MAC learning, and many other advantages.

- [Requirements on page 29](#)
- [Overview on page 29](#)
- [Configuration on page 29](#)
- [Verification on page 55](#)

Requirements

This example uses the following hardware and software components:

- Two Juniper Networks MX Series routers to act as IP gateways for the EVPN overlay
- Four Juniper Networks QFX5100 switches. Two of these switches act as PE devices in the EVPN topology, and the other two switches act as pure IP transport for the overlay.
- Junos OS Release 16.1 or later.
- Starting with Junos OS Release 17.3R1, EVPN-VXLAN is also supported on EX9200 switches. Previously, only MPLS encapsulation was supported. In this example, the EX9200 switch would function as an IP gateway for the EVPN overlay. There are some configuration differences between MX Series routers and EX9200 switches, See [EX9200 Configuration on page 37](#) for more information about specific EX9200 configuration.

Overview

Ethernet VPNs (EVPNs) enable you to connect groups of dispersed customer sites using Layer 2 virtual bridges, and Virtual Extensible LANs (VXLANs) enable you to stretch Layer 2 connection over an intervening Layer 3 network, while providing network segmentation like a VLAN, but without the scaling limitation of traditional VLANs. EVPN with VXLAN encapsulation handles Layer 2 connectivity at the scale required by cloud server providers and replaces limiting protocols like STP, freeing up your Layer 3 network to use more robust routing protocols.

This example configuration shows how to configure EVPN with VXLAN encapsulation. For the purposes of this example, the MX Series routers are named Core-1 and Core-2. The QFX5100 switches are named Leaf-1, Leaf-2, Spine-1, and Spine-2. The core routers act as IP gateways for the EVPN overlay, the leaf switches act as PE devices in the EVPN topology, and the spine switches act as pure IP transport for the overlay.

Configuration

- [Configuring Leaf-1 on page 40](#)
- [Configuring Leaf-2 on page 43](#)

- [Configuring Spine-1 on page 46](#)
- [Configuring Spine-2 on page 47](#)
- [Configuring Core-1 on page 49](#)
- [Configuring Core-2 on page 52](#)

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Leaf-1

```
set system host-name leaf-1
set chassis aggregated-devices ethernet device-count 2
set interfaces xe-0/0/2 unit 0 family inet address 10.0.0.9/31
set interfaces xe-0/0/4 unit 0 family inet address 10.0.0.13/31
set interfaces xe-0/0/32 ether-options 802.3ad ae0
set interfaces xe-0/0/33 ether-options 802.3ad ae1
set interfaces xe-0/0/34 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/34 unit 0 family ethernet-switching vlan members v300
set interfaces xe-0/0/34 unit 0 family ethernet-switching vlan members v400
set interfaces ae0 esi 00:01:01:01:01:01:01:01
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members v100
set interfaces ae1 esi 00:02:02:02:02:02:02:02
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode access
set interfaces ae1 unit 0 family ethernet-switching vlan members v200
set interfaces lo0 unit 0 family inet address 192.0.2.0/24
set routing-options router-id 192.0.2.0
set routing-options autonomous-system 65402
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 10.0.0.8 description spine-1
set protocols bgp group underlay neighbor 10.0.0.12 description spine-2
set protocols bgp group EVPN_VXLAN_CORE type external
set protocols bgp group EVPN_VXLAN_CORE multihop ttl 255
set protocols bgp group EVPN_VXLAN_CORE multihop no-nexthop-change
set protocols bgp group EVPN_VXLAN_CORE local-address 192.0.2.0
set protocols bgp group EVPN_VXLAN_CORE family evpn signaling
set protocols bgp group EVPN_VXLAN_CORE peer-as 65400
set protocols bgp group EVPN_VXLAN_CORE local-as 65403
set protocols bgp group EVPN_VXLAN_CORE neighbor 10.255.255.0 description core-1
set protocols bgp group EVPN_VXLAN_CORE neighbor 10.255.255.1 description core-2
set protocols bgp group EVPN_VXLAN_LEAF type internal
set protocols bgp group EVPN_VXLAN_LEAF local-address 192.0.2.0
set protocols bgp group EVPN_VXLAN_LEAF family evpn signaling
```

```

set protocols bgp group EVPN_VXLAN_LEAF export LEAF-PREPEND
set protocols bgp group EVPN_VXLAN_LEAF neighbor 10.255.255.5 description leaf-2
set protocols evpn encapsulation vxlan
set protocols evpn extended-vni-list 1100
set protocols evpn extended-vni-list 1200
set protocols evpn extended-vni-list 1300
set protocols evpn extended-vni-list 1400
set protocols evpn multicast-mode ingress-replication
set protocols evpn vni-routing-options vni 1100 vrf-target export target:1:100
set protocols evpn vni-routing-options vni 1200 vrf-target export target:1:200
set protocols evpn vni-routing-options vni 1300 vrf-target export target:1:300
set protocols evpn vni-routing-options vni 1400 vrf-target export target:1:400
set protocols l2-learning traceoptions file l2ald.log
set protocols l2-learning traceoptions file size 10m
set protocols l2-learning traceoptions level all
set protocols l2-learning traceoptions flag all
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement LEAF-PREPEND then as-path-prepend 65402
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0
prefix-length-range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance
per-packet
set policy-options policy-statement vrf-imp term t1 from community com100
set policy-options policy-statement vrf-imp term t1 then accept
set policy-options policy-statement vrf-imp term t2 from community com200
set policy-options policy-statement vrf-imp term t2 then accept
set policy-options policy-statement vrf-imp term t3 from community com300
set policy-options policy-statement vrf-imp term t3 then accept
set policy-options policy-statement vrf-imp term t4 from community com400
set policy-options policy-statement vrf-imp term t4 then accept
set policy-options policy-statement vrf-imp term t5 then reject
set policy-options community com100 members target:1:100
set policy-options community com200 members target:1:200
set policy-options community com300 members target:1:300
set policy-options community com400 members target:1:400
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 192.0.2.0:1
set switch-options vrf-import vrf-imp
set switch-options vrf-target target:9999:9999
set vlans v100 vlan-id 100
set vlans v100 vxlan vni 1100
set vlans v100 vxlan ingress-node-replication
set vlans v200 vlan-id 200
set vlans v200 vxlan vni 1200
set vlans v200 vxlan ingress-node-replication
set vlans v300 vlan-id 300
set vlans v300 vxlan vni 1300
set vlans v300 vxlan ingress-node-replication
set vlans v400 vlan-id 400
set vlans v400 vxlan vni 1400
set vlans v400 vxlan ingress-node-replication

```

Leaf-2

```

set system host-name leaf-2
set chassis aggregated-devices ethernet device-count 2
set interfaces xe-0/0/3 unit 0 family inet address 10.0.0.11/31

```

```
set interfaces xe-0/0/5 unit 0 family inet address 10.0.0.15/31
set interfaces xe-0/0/36 ether-options 802.3ad ae0
set interfaces xe-0/0/37 ether-options 802.3ad ae1
set interfaces xe-0/0/38 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/38 unit 0 family ethernet-switching vlan members v300
set interfaces xe-0/0/38 unit 0 family ethernet-switching vlan members v100
set interfaces ae0 esi 00:01:01:01:01:01:01:01
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp passive
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae0 unit 0 family ethernet-switching interface-mode access
set interfaces ae0 unit 0 family ethernet-switching vlan members v100
set interfaces ae1 esi 00:02:02:02:02:02:02:02
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options lacp passive
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:01:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode access
set interfaces ae1 unit 0 family ethernet-switching vlan members v200
set interfaces lo0 unit 0 family inet address 10.255.255.5/32
set routing-options router-id 10.255.255.5
set routing-options autonomous-system 65402
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 10.0.0.10 description spine-1
set protocols bgp group underlay neighbor 10.0.0.14 description spine-2
set protocols bgp group EVPN_VXLAN_CORE type external
set protocols bgp group EVPN_VXLAN_CORE multihop ttl 255
set protocols bgp group EVPN_VXLAN_CORE multihop no-nexthop-change
set protocols bgp group EVPN_VXLAN_CORE local-address 10.255.255.5
set protocols bgp group EVPN_VXLAN_CORE family evpn signaling
set protocols bgp group EVPN_VXLAN_CORE peer-as 65400
set protocols bgp group EVPN_VXLAN_CORE neighbor 10.255.255.0 description core-1
set protocols bgp group EVPN_VXLAN_CORE neighbor 10.255.255.1 description core-2
set protocols bgp group EVPN_VXLAN_LEAF type internal
set protocols bgp group EVPN_VXLAN_LEAF local-address 10.255.255.5
set protocols bgp group EVPN_VXLAN_LEAF family evpn signaling
set protocols bgp group EVPN_VXLAN_LEAF export LEAF-PREPEND
set protocols bgp group EVPN_VXLAN_LEAF neighbor 192.0.2.0 description leaf-1
set protocols evpn encapsulation vxlan
set protocols evpn extended-vni-list 1100
set protocols evpn extended-vni-list 1200
set protocols evpn extended-vni-list 1300
set protocols evpn extended-vni-list 1400
set protocols evpn multicast-mode ingress-replication
set protocols evpn vni-routing-options vni 1100 vrf-target export target:1:100
set protocols evpn vni-routing-options vni 1200 vrf-target export target:1:200
set protocols evpn vni-routing-options vni 1300 vrf-target export target:1:300
set protocols evpn vni-routing-options vni 1400 vrf-target export target:1:400
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement LEAF-PREPEND then as-path-prepend 65402
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0
prefix-length-range /32-/32
```



```

set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance
per-packet
set policy-options policy-statement vrf-imp term t1 from community com100
set policy-options policy-statement vrf-imp term t1 then accept
set policy-options policy-statement vrf-imp term t2 from community com200
set policy-options policy-statement vrf-imp term t2 then accept
set policy-options policy-statement vrf-imp term t3 from community com300
set policy-options policy-statement vrf-imp term t3 then accept
set policy-options policy-statement vrf-imp term t4 from community com400
set policy-options policy-statement vrf-imp term t4 then accept
set policy-options policy-statement vrf-imp term t5 then reject
set policy-options community com100 members target:1:100
set policy-options community com200 members target:1:200
set policy-options community com300 members target:1:300
set policy-options community com400 members target:1:400
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 10.255.255.5:1
set switch-options vrf-import vrf-imp
set switch-options vrf-target target:9999:9999
set vlans v100 vlan-id 100
set vlans v100 vxlan vni 1100
set vlans v100 vxlan ingress-node-replication
set vlans v200 vlan-id 200
set vlans v200 vxlan vni 1200
set vlans v200 vxlan ingress-node-replication
set vlans v300 vlan-id 300
set vlans v300 vxlan vni 1300
set vlans v300 vxlan ingress-node-replication
set vlans v400 vlan-id 400
set vlans v400 vxlan vni 1400
set vlans v400 vxlan ingress-node-replication

```

Spine-1

```

set system host-name spine-1
set interfaces xe-0/0/0 unit 0 family inet address 10.0.0.1/31
set interfaces xe-0/0/1 unit 0 family inet address 10.0.0.5/31
set interfaces xe-0/0/2 unit 0 family inet address 10.0.0.8/31
set interfaces xe-0/0/3 unit 0 family inet address 10.0.0.10/31
set interfaces lo0 unit 0 family inet address 10.255.255.2/32
set routing-options router-id 10.255.255.2
set routing-options autonomous-system 65401
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay-leaf type external
set protocols bgp group underlay-leaf advertise-peer-as
set protocols bgp group underlay-leaf family inet unicast loops 2
set protocols bgp group underlay-leaf export lo0
set protocols bgp group underlay-leaf peer-as 65402
set protocols bgp group underlay-leaf multipath
set protocols bgp group underlay-leaf neighbor 10.0.0.9 description leaf-1
set protocols bgp group underlay-leaf neighbor 10.0.0.11 description leaf-2
set protocols bgp group underlay-core type external
set protocols bgp group underlay-core advertise-peer-as
set protocols bgp group underlay-core family inet unicast loops 2
set protocols bgp group underlay-core export lo0
set protocols bgp group underlay-core peer-as 65400
set protocols bgp group underlay-core multipath
set protocols bgp group underlay-core neighbor 10.0.0.0 description core-1
set protocols bgp group underlay-core neighbor 10.0.0.4 description core-2

```

```

set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0
prefix-length-range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance
per-packet

```

Spine-2

```

set system host-name spine-2
set interfaces xe-0/0/0 unit 0 family inet address 10.0.0.3/31
set interfaces xe-0/0/1 unit 0 family inet address 10.0.0.7/31
set interfaces xe-0/0/4 unit 0 family inet address 10.0.0.12/31
set interfaces xe-0/0/5 unit 0 family inet address 10.0.0.14/31
set interfaces lo0 unit 0 family inet address 10.255.255.3/32
set routing-options router-id 10.255.255.3
set routing-options autonomous-system 65401
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay-leaf type external
set protocols bgp group underlay-leaf advertise-peer-as
set protocols bgp group underlay-leaf family inet unicast loops 2
set protocols bgp group underlay-leaf export lo0
set protocols bgp group underlay-leaf peer-as 65402
set protocols bgp group underlay-leaf multipath
set protocols bgp group underlay-leaf neighbor 10.0.0.13 description leaf-1
set protocols bgp group underlay-leaf neighbor 10.0.0.15 description leaf-2
set protocols bgp group underlay-core type external
set protocols bgp group underlay-core advertise-peer-as
set protocols bgp group underlay-core family inet unicast loops 2
set protocols bgp group underlay-core export lo0
set protocols bgp group underlay-core peer-as 65400
set protocols bgp group underlay-core multipath
set protocols bgp group underlay-core neighbor 10.0.0.2 description core-1
set protocols bgp group underlay-core neighbor 10.0.0.6 description core-2
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0
prefix-length-range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance
per-packet

```

Core-1

```

set system host-name core-1
set interfaces ge-1/0/0 unit 0 family inet address 10.0.0.0/31
set interfaces ge-1/0/1 unit 0 family inet address 10.0.0.2/31
set interfaces irb unit 1100 family inet address 172.16.0.2/24
virtual-gateway-address 100.0.0.1
set interfaces irb unit 1200 family inet address 172.16.0.3/24
virtual-gateway-address 200.0.0.1
set interfaces irb unit 1300 family inet address 10.10.10.2/24
virtual-gateway-address 10.10.10.1
set interfaces irb unit 1400 family inet address 10.10.10.2/24
virtual-gateway-address 10.10.10.1

```

```

set interfaces lo0 unit 0 family inet address 10.255.255.0/32
set routing-options router-id 10.255.255.0
set routing-options autonomous-system 65400
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 10.0.0.1 description spine-1
set protocols bgp group underlay neighbor 10.0.0.3 description spine-2
set protocols bgp group EVPN_VXLAN type external
set protocols bgp group EVPN_VXLAN local-address 10.255.255.0
set protocols bgp group EVPN_VXLAN family evpn signaling
set protocols bgp group EVPN_VXLAN peer-as 65402
set protocols bgp group EVPN_VXLAN multipath
set protocols bgp group EVPN_VXLAN neighbor 192.0.2.0 description leaf-1
set protocols bgp group EVPN_VXLAN neighbor 192.0.2.0 multihop ttl 255
set protocols bgp group EVPN_VXLAN neighbor 10.255.255.5 description leaf-2
set protocols bgp group EVPN_VXLAN neighbor 10.255.255.5 multihop ttl 255
set protocols bgp group EVPN_VXLAN_TEMP type external
set protocols bgp group EVPN_VXLAN_TEMP local-address 10.255.255.0
set protocols bgp group EVPN_VXLAN_TEMP family evpn signaling
set protocols bgp group EVPN_VXLAN_TEMP peer-as 65403
set protocols bgp group EVPN_VXLAN_TEMP multipath
set protocols bgp group EVPN_VXLAN_TEMP neighbor 192.0.2.0 description leaf-1
set protocols bgp group EVPN_VXLAN_TEMP neighbor 192.0.2.0 multihop ttl 255
set protocols bgp group EVPN_VXLAN_TEMP neighbor 10.255.255.5 description leaf-2
set protocols bgp group EVPN_VXLAN_TEMP neighbor 10.255.255.5 multihop ttl 255
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement VS_VLAN100_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN100_IMP term ESI then accept
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 from community
comm-VS_VLAN100
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 then accept
set policy-options policy-statement VS_VLAN200_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN200_IMP term ESI then accept
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 from community
comm-VS_VLAN200
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 then accept
set policy-options policy-statement VS_VLAN300_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN300_IMP term ESI then accept
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 from community
comm-VS_VLAN300
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 then accept
set policy-options policy-statement VS_VLAN400_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN400_IMP term ESI then accept
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 from community
comm-VS_VLAN400
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 then accept
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0
prefix-length-range /32-/32

```

```
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance
per-packet
set policy-options community comm-VS_VLAN100 members target:1:100
set policy-options community comm-VS_VLAN200 members target:1:200
set policy-options community comm-VS_VLAN300 members target:1:300
set policy-options community comm-VS_VLAN400 members target:1:400
set policy-options community comm-leaf_esi members target:9999:9999
set routing-instances VRF_Tenant_A instance-type vrf
set routing-instances VRF_Tenant_A interface irb.1100
set routing-instances VRF_Tenant_A route-distinguisher 10.255.255.0:1100
set routing-instances VRF_Tenant_A vrf-target target:10:100
set routing-instances VRF_Tenant_A routing-options auto-export
set routing-instances VRF_Tenant_B instance-type vrf
set routing-instances VRF_Tenant_B interface irb.1200
set routing-instances VRF_Tenant_B route-distinguisher 10.255.255.0:1200
set routing-instances VRF_Tenant_B vrf-target target:10:200
set routing-instances VRF_Tenant_C instance-type vrf
set routing-instances VRF_Tenant_C interface irb.1300
set routing-instances VRF_Tenant_C route-distinguisher 10.255.255.0:1300
set routing-instances VRF_Tenant_C vrf-target target:10:300
set routing-instances VRF_Tenant_D instance-type vrf
set routing-instances VRF_Tenant_D interface irb.1400
set routing-instances VRF_Tenant_D route-distinguisher 10.255.255.0:1400
set routing-instances VRF_Tenant_D vrf-target target:10:400
set routing-instances VS_VLAN100 vtep-source-interface lo0.0
set routing-instances VS_VLAN100 instance-type virtual-switch
set routing-instances VS_VLAN100 route-distinguisher 10.255.255.0:100
set routing-instances VS_VLAN100 vrf-import VS_VLAN100_IMP
set routing-instances VS_VLAN100 vrf-target target:1:100
set routing-instances VS_VLAN100 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN100 protocols evpn extended-vni-list 1100
set routing-instances VS_VLAN100 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN100 bridge-domains bd1100 vlan-id 100
set routing-instances VS_VLAN100 bridge-domains bd1100 routing-interface irb.1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan vni 1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan
ingress-node-replication
set routing-instances VS_VLAN200 vtep-source-interface lo0.0
set routing-instances VS_VLAN200 instance-type virtual-switch
set routing-instances VS_VLAN200 route-distinguisher 10.255.255.0:200
set routing-instances VS_VLAN200 vrf-import VS_VLAN200_IMP
set routing-instances VS_VLAN200 vrf-target target:1:200
set routing-instances VS_VLAN200 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN200 protocols evpn extended-vni-list 1200
set routing-instances VS_VLAN200 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN200 bridge-domains bd1200 vlan-id 200
set routing-instances VS_VLAN200 bridge-domains bd1200 routing-interface irb.1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan vni 1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan
ingress-node-replication
set routing-instances VS_VLAN300 vtep-source-interface lo0.0
set routing-instances VS_VLAN300 instance-type virtual-switch
set routing-instances VS_VLAN300 route-distinguisher 10.255.255.0:300
set routing-instances VS_VLAN300 vrf-import VS_VLAN300_IMP
set routing-instances VS_VLAN300 vrf-target target:1:300
set routing-instances VS_VLAN300 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN300 protocols evpn extended-vni-list 1300
set routing-instances VS_VLAN300 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN300 bridge-domains bd1300 vlan-id 300
set routing-instances VS_VLAN300 bridge-domains bd1300 routing-interface irb.1300
```

```

set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan vni 1300
set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan
ingress-node-replication
set routing-instances VS_VLAN400 vtep-source-interface lo0.0
set routing-instances VS_VLAN400 instance-type virtual-switch
set routing-instances VS_VLAN400 route-distinguisher 10.255.255.0:400
set routing-instances VS_VLAN400 vrf-import VS_VLAN400_IMP
set routing-instances VS_VLAN400 vrf-target target:1:400
set routing-instances VS_VLAN400 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN400 protocols evpn extended-vni-list 1400
set routing-instances VS_VLAN400 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN400 bridge-domains bd1400 vlan-id 400
set routing-instances VS_VLAN400 bridge-domains bd1400 routing-interface irb.1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan vni 1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan
ingress-node-replication

```

EX9200 Configuration

On EX9200 switches, the **vlan**s statement is used instead of **bridge-domains**, and the **l3-interface** statement is used instead of **routing-interface**.

The following example shows how to configure these statements. All other configuration shown for MX Series routers in this example applies to EX9200 switches.

```

set routing-instances VS_VLAN300 vlans vlan1300 vlan-id 300
set routing-instances VS_VLAN300 vlans vlan1300 l3-interface irb.1300

```



NOTE: In this example, wherever **bridge-domains** or **routing-interface** statements are used, to configure on EX9200 switches, use **vlans** and **l3-interface** instead.

Core-2

```

set system host-name core-2
set chassis network-services enhanced-ip
set interfaces ge-1/0/0 unit 0 family inet address 10.0.0.4/31
set interfaces ge-1/0/1 unit 0 family inet address 10.0.0.6/31
set interfaces irb unit 1100 family inet address 172.16.0.4/24
virtual-gateway-address 100.0.0.1
set interfaces irb unit 1200 family inet address 172.16.0.5/24
virtual-gateway-address 200.0.0.1
set interfaces irb unit 1300 family inet address 10.10.10.3/24
virtual-gateway-address 10.10.10.1
set interfaces irb unit 1400 family inet address 10.10.10.3/24
virtual-gateway-address 10.10.10.1
set interfaces lo0 unit 0 family inet address 10.255.255.1/32
set routing-options router-id 10.255.255.1
set routing-options autonomous-system 65400
set routing-options forwarding-table export load-balance
set routing-options forwarding-table ecmp-fast-reroute
set protocols bgp group underlay type external
set protocols bgp group underlay advertise-peer-as
set protocols bgp group underlay family inet unicast loops 2
set protocols bgp group underlay export lo0

```

```
set protocols bgp group underlay peer-as 65401
set protocols bgp group underlay multipath
set protocols bgp group underlay neighbor 10.0.0.5 description spine-1
set protocols bgp group underlay neighbor 10.0.0.7 description spine-2
set protocols bgp group EVPN_VXLAN type external
set protocols bgp group EVPN_VXLAN local-address 10.255.255.1
set protocols bgp group EVPN_VXLAN family evpn signaling
set protocols bgp group EVPN_VXLAN peer-as 65402
set protocols bgp group EVPN_VXLAN multipath
set protocols bgp group EVPN_VXLAN neighbor 192.0.2.0 description leaf-1
set protocols bgp group EVPN_VXLAN neighbor 192.0.2.0 multihop ttl 255
set protocols bgp group EVPN_VXLAN neighbor 10.255.255.5 description leaf-2
set protocols bgp group EVPN_VXLAN neighbor 10.255.255.5 multihop ttl 255
set protocols lldp port-id-subtype interface-name
set protocols lldp interface all
set policy-options policy-statement VS_VLAN100_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN100_IMP term ESI then accept
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 from community
comm-VS_VLAN100
set policy-options policy-statement VS_VLAN100_IMP term VS_VLAN100 then accept
set policy-options policy-statement VS_VLAN200_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN200_IMP term ESI then accept
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 from community
comm-VS_VLAN200
set policy-options policy-statement VS_VLAN200_IMP term VS_VLAN200 then accept
set policy-options policy-statement VS_VLAN300_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN300_IMP term ESI then accept
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 from community
comm-VS_VLAN300
set policy-options policy-statement VS_VLAN300_IMP term VS_VLAN300 then accept
set policy-options policy-statement VS_VLAN400_IMP term ESI from community
comm-leaf_esi
set policy-options policy-statement VS_VLAN400_IMP term ESI then accept
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 from community
comm-VS_VLAN400
set policy-options policy-statement VS_VLAN400_IMP term VS_VLAN400 then accept
set policy-options policy-statement lo0 from family inet
set policy-options policy-statement lo0 from protocol direct
set policy-options policy-statement lo0 from route-filter 0.0.0.0/0
prefix-length-range /32-/32
set policy-options policy-statement lo0 then accept
set policy-options policy-statement load-balance term 1 then load-balance
per-packet
set policy-options community comm-VS_VLAN100 members target:1:100
set policy-options community comm-VS_VLAN200 members target:1:200
set policy-options community comm-VS_VLAN300 members target:1:300
set policy-options community comm-VS_VLAN400 members target:1:400
set policy-options community comm-leaf_esi members target:9999:9999
set routing-instances VRF_Tenant_A instance-type vrf
set routing-instances VRF_Tenant_A interface irb.1100
set routing-instances VRF_Tenant_A route-distinguisher 10.255.255.1:1100
set routing-instances VRF_Tenant_A vrf-target target:10:100
set routing-instances VRF_Tenant_A routing-options auto-export
set routing-instances VRF_Tenant_B instance-type vrf
set routing-instances VRF_Tenant_B interface irb.1200
set routing-instances VRF_Tenant_B route-distinguisher 10.255.255.1:1200
set routing-instances VRF_Tenant_B vrf-target target:10:200
set routing-instances VRF_Tenant_C instance-type vrf
```

```

set routing-instances VRF_Tenant_C interface irb.1300
set routing-instances VRF_Tenant_C route-distinguisher 10.255.255.1:1300
set routing-instances VRF_Tenant_C vrf-target target:10:300
set routing-instances VRF_Tenant_D instance-type vrf
set routing-instances VRF_Tenant_D interface irb.1400
set routing-instances VRF_Tenant_D route-distinguisher 10.255.255.1:1400
set routing-instances VRF_Tenant_D vrf-target target:10:400
set routing-instances VS_VLAN100 vtep-source-interface lo0.0
set routing-instances VS_VLAN100 instance-type virtual-switch
set routing-instances VS_VLAN100 route-distinguisher 10.255.255.1:100
set routing-instances VS_VLAN100 vrf-import VS_VLAN100_IMP
set routing-instances VS_VLAN100 vrf-target target:1:100
set routing-instances VS_VLAN100 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN100 protocols evpn extended-vni-list 1100
set routing-instances VS_VLAN100 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN100 protocols evpn default-gateway
no-gateway-community
set routing-instances VS_VLAN100 bridge-domains bd1100 vlan-id 100
set routing-instances VS_VLAN100 bridge-domains bd1100 routing-interface irb.1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan vni 1100
set routing-instances VS_VLAN100 bridge-domains bd1100 vxlan
ingress-node-replication
set routing-instances VS_VLAN200 vtep-source-interface lo0.0
set routing-instances VS_VLAN200 instance-type virtual-switch
set routing-instances VS_VLAN200 route-distinguisher 10.255.255.1:200
set routing-instances VS_VLAN200 vrf-import VS_VLAN200_IMP
set routing-instances VS_VLAN200 vrf-target target:1:200
set routing-instances VS_VLAN200 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN200 protocols evpn extended-vni-list 1200
set routing-instances VS_VLAN200 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN200 bridge-domains bd1200 vlan-id 200
set routing-instances VS_VLAN200 bridge-domains bd1200 routing-interface irb.1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan vni 1200
set routing-instances VS_VLAN200 bridge-domains bd1200 vxlan
ingress-node-replication
set routing-instances VS_VLAN300 vtep-source-interface lo0.0
set routing-instances VS_VLAN300 instance-type virtual-switch
set routing-instances VS_VLAN300 route-distinguisher 10.255.255.1:300
set routing-instances VS_VLAN300 vrf-import VS_VLAN300_IMP
set routing-instances VS_VLAN300 vrf-target target:1:300
set routing-instances VS_VLAN300 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN300 protocols evpn extended-vni-list 1300
set routing-instances VS_VLAN300 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN300 bridge-domains bd1300 vlan-id 300
set routing-instances VS_VLAN300 bridge-domains bd1300 routing-interface irb.1300
set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan vni 1300
set routing-instances VS_VLAN300 bridge-domains bd1300 vxlan
ingress-node-replication
set routing-instances VS_VLAN400 vtep-source-interface lo0.0
set routing-instances VS_VLAN400 instance-type virtual-switch
set routing-instances VS_VLAN400 route-distinguisher 10.255.255.1:400
set routing-instances VS_VLAN400 vrf-import VS_VLAN400_IMP
set routing-instances VS_VLAN400 vrf-target target:1:400
set routing-instances VS_VLAN400 protocols evpn encapsulation vxlan
set routing-instances VS_VLAN400 protocols evpn extended-vni-list 1400
set routing-instances VS_VLAN400 protocols evpn multicast-mode ingress-replication
set routing-instances VS_VLAN400 bridge-domains bd1400 vlan-id 400
set routing-instances VS_VLAN400 bridge-domains bd1400 routing-interface irb.1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan vni 1400
set routing-instances VS_VLAN400 bridge-domains bd1400 vxlan
ingress-node-replication

```

Configuring Leaf-1

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

Configure the QFX5100 switch called Leaf-1:

1. Set the system hostname.

```
[edit]
user@leaf-1# set system host-name leaf-1
```

2. Configure routing options.

```
[edit routing-options]
user@leaf-1# set router-id 192.0.2.0
user@leaf-1# set autonomous-system 65402
user@leaf-1# set forwarding-table export load-balance
user@leaf-1# set forwarding-table ecmp-fast-reroute
```

3. Configure load balancing.

```
[edit policy-options policy-statement load-balance]
user@leaf-1# set term 1 then load-balance per-packet
```

4. Make sure lo0 is exported and thus advertised into the underlay, then configure **family inet unicast loops 2**. Doing this is necessary because of the design choice to re-use the same autonomous system (AS) number within a tier.

```
[edit protocols bgp group underlay]
user@leaf-1# set type external
user@leaf-1# set advertise-peer-as
user@leaf-1# set family inet unicast loops 2
user@leaf-1# set export lo0
user@leaf-1# set peer-as 65401
user@leaf-1# set multipath
user@leaf-1# set neighbor 10.0.0.8 description spine-1
user@leaf-1# set neighbor 10.0.0.12 description spine-2
```

5. Configure policy options for the loopback address.

```
[edit policy-options policy-statement lo0]
user@leaf-1# set from family inet
user@leaf-1# set from protocol direct
user@leaf-1# set from route-filter 0.0.0.0/0 prefix-length-range /32-/32
user@leaf-1# set then accept
```

6. Configure switch options. At the **[switch-options]** hierarchy level, the virtual tunnel endpoint interface is lo0.0, which must be reachable through the underlay routing protocol. The route distinguisher must be unique across all switches in the network

to ensure all route advertisements within MP-BGP are globally unique. The VRF table target on the QFX Series switch is, at a minimum, the community with which the switch sends all ESI (Type-1) routes. The **vrf-import vrf-imp** statement defines the target community list, which is imported into the default-switch.evpn.0 instance from bgp.evpn.0.

```
[edit switch-options]
user@leaf-1# set vtep-source-interface lo0.0
user@leaf-1# set route-distinguisher 192.0.2.0:1
user@leaf-1# set vrf-import vrf-imp
user@leaf-1# set vrf-target target:9999:9999
```

7. Configure the VRF table import policy.

```
[edit policy-options policy-statement vrf-imp]
user@leaf-1# set term t1 from community com100
user@leaf-1# set vrf-imp term t1 then accept
user@leaf-1# set vrf-imp term t2 from community com200
user@leaf-1# set vrf-imp term t2 then accept
user@leaf-1# set vrf-imp term t3 from community com300
user@leaf-1# set vrf-imp term t3 then accept
user@leaf-1# set vrf-imp term t4 from community com400
user@leaf-1# set vrf-imp term t4 then accept
user@leaf-1# set vrf-imp term t5 then reject
```

8. Configure the community policy options.

```
[edit policy-options community]
user@leaf-1# set com100 members target:1:100
user@leaf-1# set com200 members target:1:200
user@leaf-1# set com300 members target:1:300
user@leaf-1# set com400 members target:1:400
```

9. Configure the extended virtual network identifier list to establish which VXLAN network identifiers you want to be part of the EVPN and VXLAN MP-BGP domain. Next, set up the ingress replication; this EVPN and VXLAN ingress-replication is used instead of a multicast underlay. Then configure different route targets for each VXLAN network identifier instance under **vni-routing-options**.

```
[edit protocols evpn]
user@leaf-1# set encapsulation vxlan
user@leaf-1# set extended-vni-list 1100
user@leaf-1# set extended-vni-list 1200
user@leaf-1# set extended-vni-list 1300
user@leaf-1# set extended-vni-list 1400
user@leaf-1# set multicast-mode ingress-replication
user@leaf-1# set vni-routing-options vni 1100 vrf-target export target:1:100
user@leaf-1# set vni-routing-options vni 1200 vrf-target export target:1:200
user@leaf-1# set vni-routing-options vni 1300 vrf-target export target:1:300
user@leaf-1# set vni-routing-options vni 1400 vrf-target export target:1:400
```

10. Map locally significant VLAN IDs to globally significant VXLAN network identifiers.

```
[edit vlans]
```

```

user@leaf-1# set v100 vlan-id 100
user@leaf-1# set v100 vxlan vni 1100
user@leaf-1# set v100 vxlan ingress-node-replication
user@leaf-1# set v200 vlan-id 200
user@leaf-1# set v200 vxlan vni 1200
user@leaf-1# set v200 vxlan ingress-node-replication
user@leaf-1# set v300 vlan-id 300
user@leaf-1# set v300 vxlan vni 1300
user@leaf-1# set v300 vxlan ingress-node-replication
user@leaf-1# set v400 vlan-id 400
user@leaf-1# set v400 vxlan vni 1400
user@leaf-1# set v400 vxlan ingress-node-replication

```

11. Configure the EVPN MP-BGP sessions.

```

[edit protocols bgp group EVPN_VXLAN_CORE]

user@leaf-1# set type external
user@leaf-1# set multihop ttl 255
user@leaf-1# set multihop no-nexthop-change
user@leaf-1# set local-address 192.0.2.0
user@leaf-1# set family evpn signaling
user@leaf-1# set peer-as 65400
user@leaf-1# set local-as 65403
user@leaf-1# set neighbor 10.255.255.0 description core-1
user@leaf-1# set neighbor 10.255.255.1 description core-2
user@leaf-1# set type internal
user@leaf-1# set local-address 192.0.2.0
user@leaf-1# set family evpn signaling
user@leaf-1# set export LEAF-PREPEND
user@leaf-1# set neighbor 10.255.255.5 description leaf-2

```

12. Configure the Gigabit Ethernet interfaces.

```

[edit interfaces]

user@leaf-1# set xe-0/0/2 unit 0 family inet address 10.0.0.9/31
user@leaf-1# set xe-0/0/4 unit 0 family inet address 10.0.0.13/31
user@leaf-1# set xe-0/0/32 ether-options 802.3ad ae0
user@leaf-1# set xe-0/0/33 ether-options 802.3ad ae1
user@leaf-1# set xe-0/0/34 unit 0 family ethernet-switching interface-mode trunk
user@leaf-1# set xe-0/0/34 unit 0 family ethernet-switching vlan members v300
user@leaf-1# set xe-0/0/34 unit 0 family ethernet-switching vlan members v400

```

13. Configure two LACP-enabled LAG interfaces. The ESI value is globally unique across the entire EVPN domain. The **all-active** configuration statement ensures that all PE routers to which this multihomed tenant is attached to can forward traffic from the CE device, such that all CE links are actively used.

```

[edit interfaces]

user@leaf-1# set ae0 esi 00:01:01:01:01:01:01:01:01
user@leaf-1# set ae0 esi all-active
user@leaf-1# set e0 aggregated-ether-options lacp active
user@leaf-1# set ae0 aggregated-ether-options lacp system-id 00:00:00:01:01:01
user@leaf-1# set ae0 unit 0 family ethernet-switching interface-mode access
user@leaf-1# set ae0 unit 0 family ethernet-switching vlan members v100

```

```

user@leaf-1# set ae1 esi 00:02:02:02:02:02:02:02:02
user@leaf-1# set ae1 esi all-active
user@leaf-1# set ae1 aggregated-ether-options lacp active
user@leaf-1# set ae1 aggregated-ether-options lacp system-id 00:00:00:01:01:01
user@leaf-1# set ae1 unit 0 family ethernet-switching interface-mode access
user@leaf-1# set ae1 unit 0 family ethernet-switching vlan members v200

```

14. Configure the loopback interface address.

```

[edit interfaces]
user@leaf-1# set lo0 unit 0 family inet address 192.0.2.0/24

```

Configuring Leaf-2

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

The configuration of Leaf-2 is very similar to the configuration of Leaf-1. Configure the QFX5100 switch called Leaf-2:

1. Set the system hostname.

```

[edit]
user@leaf-2# set system host-name leaf-2

```

2. Configure routing options.

```

[edit routing-options]
user@leaf-2# set router-id 10.255.255.5
user@leaf-2# set autonomous-system 65402
user@leaf-2# set forwarding-table export load-balance
user@leaf-2# set forwarding-table ecmp-fast-reroute

```

3. Configure load balancing.

```

[edit policy-options policy-statement load-balance]
user@leaf-2# set term 1 then load-balance per-packet

```

4. Make sure lo0 is exported and thus advertised into the underlay, then configure **family inet unicast loops 2**. Doing this is necessary because of the design choice to re-use the same autonomous system (AS) number within a tier.

```

[edit protocols bgp group underlay]
user@leaf-2# set type external
user@leaf-2# set advertise-peer-as
user@leaf-2# set family inet unicast loops 2
user@leaf-2# set export lo0
user@leaf-2# set peer-as 65401
user@leaf-2# set multipath
user@leaf-2# set neighbor 10.0.0.10 description spine-1
user@leaf-2# set neighbor 10.0.0.14 description spine-2

```

5. Configure policy options for the loopback address.

```
[edit policy-options policy-statement lo0]
user@leaf-2# set from family inet
user@leaf-2# set from protocol direct
user@leaf-2# set from route-filter 0.0.0.0/0 prefix-length-range /32-/32
user@leaf-2# set then accept
```

6. Configure switch options. At the **[switch-options]** hierarchy level, the virtual tunnel endpoint interface is lo0.0, which must be reachable through the underlay routing protocol. The route distinguisher must be unique across all switches in the network to ensure all route advertisements within MP-BGP are globally unique. The VRF table target on the QFX Series switch is, at a minimum, the community with which the switch sends all ESI (Type-1) routes. The **vrf-import vrf-imp** statement defines the target community list, which is imported into the default-switch.evpn.0 instance from bgp.evpn.0.

```
[edit switch-options]
user@leaf-2# set vtep-source-interface lo0.0
user@leaf-2# set route-distinguisher 10.255.255.5:1
user@leaf-2# set vrf-import vrf-imp
user@leaf-2# set vrf-target target:9999:9999
```

7. Configure the VRF table import policy.

```
[edit policy-options policy-statement vrf-imp]
user@leaf-2# set term t1 from community com100
user@leaf-2# set vrf-imp term t1 then accept
user@leaf-2# set vrf-imp term t2 from community com200
user@leaf-2# set vrf-imp term t2 then accept
user@leaf-2# set vrf-imp term t3 from community com300
user@leaf-2# set vrf-imp term t3 then accept
user@leaf-2# set vrf-imp term t4 from community com400
user@leaf-2# set vrf-imp term t4 then accept
user@leaf-2# set vrf-imp term t5 then reject
```

8. Configure the community policy options.

```
[edit policy-options community]
user@leaf-2# set com100 members target:1:100
user@leaf-2# set com200 members target:1:200
user@leaf-2# set com300 members target:1:300
user@leaf-2# set com400 members target:1:400
```

9. Configure the extended virtual network identifier list to establish which VXLAN network identifiers you want to be part of the EVPN and VXLAN MP-BGP domain. Next, set up the ingress replication; this EVPN and VXLAN ingress-replication is used instead of a multicast underlay. Then configure different route targets for each VXLAN network identifier instance under **vni-routing-options**.

```
[edit protocols evpn]
user@leaf-2# set encapsulation vxlan
user@leaf-2# set extended-vni-list 1100
```

```

user@leaf-2# set extended-vni-list 1200
user@leaf-2# set extended-vni-list 1300
user@leaf-2# set extended-vni-list 1400
user@leaf-2# set multicast-mode ingress-replication
user@leaf-2# set vni-routing-options vni 1100 vrf-target export target:1:100
user@leaf-2# set vni-routing-options vni 1200 vrf-target export target:1:200
user@leaf-2# set vni-routing-options vni 1300 vrf-target export target:1:300
user@leaf-2# set vni-routing-options vni 1400 vrf-target export target:1:400

```

10. Map locally significant VLAN IDs to globally significant VXLAN network identifiers.

```

[edit vlans]

user@leaf-2# set v100 vlan-id 100
user@leaf-2# set v100 vxlan vni 1100
user@leaf-2# set v100 vxlan ingress-node-replication
user@leaf-2# set v200 vlan-id 200
user@leaf-2# set v200 vxlan vni 1200
user@leaf-2# set v200 vxlan ingress-node-replication
user@leaf-2# set v300 vlan-id 300
user@leaf-2# set v300 vxlan vni 1300
user@leaf-2# set v300 vxlan ingress-node-replication
user@leaf-2# set v400 vlan-id 400
user@leaf-2# set v400 vxlan vni 1400
user@leaf-2# set v400 vxlan ingress-node-replication

```

11. Configure the EVPN MP-BGP sessions.

```

[edit protocols bgp group EVPN_VXLAN_CORE]

user@leaf-2# set type external
user@leaf-2# set multihop ttl 255
user@leaf-2# set multihop no-nexthop-change
user@leaf-2# set local-address 10.255.255.5
user@leaf-2# set family evpn signaling
user@leaf-2# set peer-as 65400
user@leaf-2# set neighbor 10.255.255.0 description core-1
user@leaf-2# set neighbor 10.255.255.1 description core-2
user@leaf-2# set type internal
user@leaf-2# set local-address 10.255.255.5
user@leaf-2# set family evpn signaling
user@leaf-2# set export LEAF-PREPEND
user@leaf-2# set neighbor 192.0.2.0 description leaf-2

```

12. Configure the Gigabit Ethernet interfaces.

```

[edit interfaces]

user@leaf-2# set xe-0/0/3 unit 0 family inet address 10.0.0.11/31
user@leaf-2# set xe-0/0/5 unit 0 family inet address 10.0.0.15/31
user@leaf-2# set xe-0/0/36 ether-options 802.3ad ae0
user@leaf-2# set xe-0/0/37 ether-options 802.3ad ae1
user@leaf-2# set xe-0/0/38 unit 0 family ethernet-switching interface-mode trunk
user@leaf-2# set xe-0/0/38 unit 0 family ethernet-switching vlan members v300
user@leaf-2# set xe-0/0/38 unit 0 family ethernet-switching vlan members v100

```

13. Configure two LACP-enabled LAG interfaces. The ESI value is globally unique across the entire EVPN domain. The **all-active** configuration statement ensures that all PE routers to which this multihomed tenant is attached to can forward traffic from the CE device, such that all CE links are actively used.

```
[edit interfaces]
```

```
user@leaf-2# set ae0 esi 00:01:01:01:01:01:01:01:01
user@leaf-2# set ae0 esi all-active
user@leaf-2# set ae0 aggregated-ether-options lacp passive
user@leaf-2# set ae0 aggregated-ether-options lacp system-id 00:00:00:01:01:01
user@leaf-2# set ae0 unit 0 family ethernet-switching interface-mode access
user@leaf-2# set ae0 unit 0 family ethernet-switching vlan members v100
user@leaf-2# set ae1 esi 00:02:02:02:02:02:02:02:02
user@leaf-2# set ae1 esi all-active
user@leaf-2# set ae1 aggregated-ether-options lacp passive
user@leaf-2# set ae1 aggregated-ether-options lacp system-id 00:00:00:01:01:01
user@leaf-2# set ae1 unit 0 family ethernet-switching interface-mode access
user@leaf-2# set ae1 unit 0 family ethernet-switching vlan members v200
```

14. Configure the loopback interface address.

```
[edit interfaces]
```

```
user@leaf-2# set lo0 unit 0 family inet address 10.255.255.5/32
```

Configuring Spine-1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

The configuration of the spine switches is similar to the leaf switches. To configure Spine-1:

1. Set the system hostname.

```
[edit]
```

```
user@spine-1# set system host-name spine-1
```

2. Configure the routing options.

```
[edit routing-options]
```

```
user@spine-1# set router-id 10.255.255.2
user@spine-1# set autonomous-system 65401
user@spine-1# set forwarding-table export load-balance
user@spine-1# set forwarding-table ecmp-fast-reroute
```

3. Configure load balance policy options.

```
[edit policy-options policy-statement load-balance]
```

```
user@spine-1# set term 1 then load-balance per-packet
```

4. Configure the underlay Leaf group with the **advertise-peer-as** statement to enable Spine-1 to bypass EBGp rules and re-advertise a route to the same autonomous system (AS) number.

```
[edit protocols bgp group underlay-leaf]
user@spine-1# set type external
user@spine-1# set advertise-peer-as
user@spine-1# set family inet unicast loops 2
user@spine-1# set export lo0
user@spine-1# set peer-as 65402
user@spine-1# set multipath
user@spine-1# set neighbor 10.0.0.9 description leaf-1
user@spine-1# set neighbor 10.0.0.11 description leaf-2
```

5. Configure the underlay Core group. The MX Series Core routers are not EVPN peered, and do not require reachability to the other's loopback, so including the **advertise-peer-as** configuration statement is optional.

```
[edit protocols bgp group underlay-core]
user@spine-1# set type external
user@spine-1# set advertise-peer-as
user@spine-1# set family inet unicast loops 2
user@spine-1# set export lo0
user@spine-1# set peer-as 65400
user@spine-1# set multipath
user@spine-1# set neighbor 10.0.0.0 description core-1
user@spine-1# set neighbor 10.0.0.4 description core-2
```

6. Configure policy options for the loopback address.

```
[edit policy-options policy-statement lo0]
user@spine-1# set from family inet
user@spine-1# set from protocol direct
user@spine-1# set from route-filter 0.0.0.0/0 prefix-length-range /32-/32
user@spine-1# set then accept
```

Configuring Spine-2

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

The configuration of the spine switches is similar to the leaf switches. To configure Spine-2:

1. Set the system hostname.

```
[edit]
user@spine-2# set system host-name spine-2
```

2. Configure the routing options.

```
[edit routing-options]
```

```
user@spine-2# set router-id 10.255.255.3
user@spine-2# set autonomous-system 65401
user@spine-2# set forwarding-table export load-balance
user@spine-2# set forwarding-table ecmp-fast-reroute
```

3. Configure load balance policy options.

```
[edit policy-options policy-statement load-balance]
user@spine-2# set term 1 then load-balance per-packet
```

4. Configure the underlay Leaf group with the **advertise-peer-as** statement to enable Spine-2 to bypass EBGp rules and re-advertise a route to the same autonomous system (AS) number.

```
[edit protocols bgp group underlay-leaf]
user@spine-2# set type external
user@spine-2# set advertise-peer-as
user@spine-2# set family inet unicast loops 2
user@spine-2# set export lo0
user@spine-2# set peer-as 65402
user@spine-2# set multipath
user@spine-2# set neighbor 10.0.0.13 description leaf-1
user@spine-2# set neighbor 10.0.0.15 description leaf-2
```

5. Configure the underlay Core group. The MX Series Core routers are not EVPN peered, and do not require reachability to the other's loopback, so including the **advertise-peer-as** configuration statement is optional.

```
[edit protocols bgp group underlay-core]
user@spine-2# set type external
user@spine-2# set advertise-peer-as
user@spine-2# set family inet unicast loops 2
user@spine-2# set export lo0
user@spine-2# set peer-as 65400
user@spine-2# set multipath
user@spine-2# set neighbor 10.0.0.2 description core-1
user@spine-2# set neighbor 10.0.0.6 description core-2
```

6. Configure policy options for the loopback address.

```
[edit policy-options policy-statement lo0]
user@spine-2# set from family inet
user@spine-2# set from protocol direct
user@spine-2# set from route-filter 0.0.0.0/0 prefix-length-range /32-/32
user@spine-2# set then accept
```


Configuring Core-1

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Core-1:

1. Set the system hostname.

```
[edit]
```

```
user@core-1# set system host-name core-1
```

2. Configure the router ID and other routing options.

```
[edit routing-options]
```

```
user@core-1# set router-id 10.255.255.0
```

```
user@core-1# set autonomous-system 65400
```

```
user@core-1# set forwarding-table export load-balance
```

```
user@core-1# set forwarding-table ecmp-fast-reroute
```

3. Configure the load balance policy options.

```
[edit policy-options policy-statement load-balance]
```

```
user@core-1# set term 1 then load-balance per-packet
```

4. Configure the BGP underlay group.

```
[edit protocols bgp group underlay]
```

```
user@core-1# set type external
```

```
user@core-1# set advertise-peer-as
```

```
user@core-1# set family inet unicast loops 2
```

```
user@core-1# set export lo0
```

```
user@core-1# set peer-as 65401
```

```
user@core-1# set multipath
```

```
user@core-1# set neighbor 10.0.0.1 description spine-1
```

```
user@core-1# set neighbor 10.0.0.3 description spine-2
```

5. Configure the loopback address policy options.

```
[edit policy-options policy-statement lo0]
```

```
user@core-1# set from family inet
```

```
user@core-1# set from protocol direct
```

```
user@core-1# set lo0 from route-filter 0.0.0.0/0 prefix-length-range /32-/32
```

```
user@core-1# set lo0 then accept
```

6. A large portion of Core-1's configuration takes place in the **[routing-instance]** hierarchy. Configure the virtual routers and configure a unique VRF table import policy for each virtual switch.

```
[edit routing-instances]
```

```
user@core-1# set VRF_Tenant_A instance-type vrf
user@core-1# set VRF_Tenant_A interface irb.1100
user@core-1# set VRF_Tenant_A route-distinguisher 10.255.255.0:1100
user@core-1# set VRF_Tenant_A vrf-target target:10:100
user@core-1# set VRF_Tenant_A routing-options auto-export
user@core-1# set VRF_Tenant_B instance-type vrf
user@core-1# set VRF_Tenant_B interface irb.1200
user@core-1# set VRF_Tenant_B route-distinguisher 10.255.255.0:1200
user@core-1# set VRF_Tenant_B vrf-target target:10:200
user@core-1# set VRF_Tenant_C instance-type vrf
user@core-1# set VRF_Tenant_C interface irb.1300
user@core-1# set VRF_Tenant_C route-distinguisher 10.255.255.0:1300
user@core-1# set VRF_Tenant_C vrf-target target:10:300
user@core-1# set VRF_Tenant_D instance-type vrf
user@core-1# set VRF_Tenant_D interface irb.1400
user@core-1# set VRF_Tenant_D route-distinguisher 10.255.255.0:1400
user@core-1# set VRF_Tenant_D vrf-target target:10:400
user@core-1# set VS_VLAN100 vtep-source-interface lo0.0
user@core-1# set VS_VLAN100 instance-type virtual-switch
user@core-1# set VS_VLAN100 route-distinguisher 10.255.255.0:100
user@core-1# set VS_VLAN100 vrf-import VS_VLAN100_IMP
user@core-1# set VS_VLAN100 vrf-target target:1:100
user@core-1# set VS_VLAN100 protocols evpn encapsulation vxlan
user@core-1# set VS_VLAN100 protocols evpn extended-vni-list 1100
user@core-1# set VS_VLAN100 protocols evpn multicast-mode ingress-replication
user@core-1# set VS_VLAN100 bridge-domains bd1100 vlan-id 100
user@core-1# set VS_VLAN100 bridge-domains bd1100 routing-interface irb.1100
user@core-1# set VS_VLAN100 bridge-domains bd1100 vxlan vni 1100
user@core-1# set VS_VLAN100 bridge-domains bd1100 vxlan
ingress-node-replication
user@core-1# set VS_VLAN200 vtep-source-interface lo0.0
user@core-1# set VS_VLAN200 instance-type virtual-switch
user@core-1# set VS_VLAN200 route-distinguisher 10.255.255.0:200
user@core-1# set VS_VLAN200 vrf-import VS_VLAN200_IMP
user@core-1# set VS_VLAN200 vrf-target target:1:200
user@core-1# set VS_VLAN200 protocols evpn encapsulation vxlan
user@core-1# set VS_VLAN200 protocols evpn extended-vni-list 1200
user@core-1# set VS_VLAN200 protocols evpn multicast-mode ingress-replication
user@core-1# set VS_VLAN200 bridge-domains bd1200 vlan-id 200
user@core-1# set VS_VLAN200 bridge-domains bd1200 routing-interface irb.1200
user@core-1# set VS_VLAN200 bridge-domains bd1200 vxlan vni 1200
user@core-1# set VS_VLAN200 bridge-domains bd1200 vxlan
ingress-node-replication
user@core-1# set VS_VLAN300 vtep-source-interface lo0.0
user@core-1# set VS_VLAN300 instance-type virtual-switch
user@core-1# set VS_VLAN300 route-distinguisher 10.255.255.0:300
user@core-1# set VS_VLAN300 vrf-import VS_VLAN300_IMP
user@core-1# set VS_VLAN300 vrf-target target:1:300
user@core-1# set VS_VLAN300 protocols evpn encapsulation vxlan
user@core-1# set VS_VLAN300 protocols evpn extended-vni-list 1300
user@core-1# set VS_VLAN300 protocols evpn multicast-mode ingress-replication
user@core-1# set VS_VLAN300 bridge-domains bd1300 vlan-id 300
user@core-1# set VS_VLAN300 bridge-domains bd1300 routing-interface irb.1300
user@core-1# set VS_VLAN300 bridge-domains bd1300 vxlan vni 1300
user@core-1# set VS_VLAN300 bridge-domains bd1300 vxlan
ingress-node-replication
user@core-1# set VS_VLAN400 vtep-source-interface lo0.0
user@core-1# set VS_VLAN400 instance-type virtual-switch
user@core-1# set VS_VLAN400 route-distinguisher 10.255.255.0:400
user@core-1# set VS_VLAN400 vrf-import VS_VLAN400_IMP
user@core-1# set VS_VLAN400 vrf-target target:1:400
```

```

user@core-1# set VS_VLAN400 protocols evpn encapsulation vxlan
user@core-1# set VS_VLAN400 protocols evpn extended-vni-list 1400
user@core-1# set VS_VLAN400 protocols evpn multicast-mode ingress-replication
user@core-1# set VS_VLAN400 bridge-domains bd1400 vlan-id 400
user@core-1# set VS_VLAN400 bridge-domains bd1400 routing-interface irb.1400
user@core-1# set VS_VLAN400 bridge-domains bd1400 vxlan vni 1400
user@core-1# set VS_VLAN400 bridge-domains bd1400 vxlan
ingress-node-replication

```

7. Configure the individual policy statements.

```
[edit policy-options policy-statement]
```

```

user@core-1# set VS_VLAN100_IMP term ESI from community comm-leaf_es1
user@core-1# set VS_VLAN100_IMP term ESI then accept
user@core-1# set VS_VLAN100_IMP term VS_VLAN100 from community comm-VS_VLAN100
user@core-1# set VS_VLAN100_IMP term VS_VLAN100 then accept
user@core-1# set VS_VLAN200_IMP term ESI from community comm-leaf_es1
user@core-1# set VS_VLAN200_IMP term ESI then accept
user@core-1# set VS_VLAN200_IMP term VS_VLAN200 from community comm-VS_VLAN200
user@core-1# set VS_VLAN200_IMP term VS_VLAN200 then accept
user@core-1# set VS_VLAN300_IMP term ESI from community comm-leaf_es1
user@core-1# set VS_VLAN300_IMP term ESI then accept
user@core-1# set VS_VLAN300_IMP term VS_VLAN300 from community comm-VS_VLAN300
user@core-1# set VS_VLAN300_IMP term VS_VLAN300 then accept
user@core-1# set VS_VLAN400_IMP term ESI from community comm-leaf_es1
user@core-1# set VS_VLAN400_IMP term ESI then accept
user@core-1# set VS_VLAN400_IMP term VS_VLAN400 from community comm-VS_VLAN400
user@core-1# set VS_VLAN400_IMP term VS_VLAN400 then accept
user@core-1# set lo0 from family inet
user@core-1# set lo0 from protocol direct
user@core-1# set lo0 from route-filter 0.0.0.0/0 prefix-length-range /32-/32
user@core-1# set lo0 then accept

```

8. Configure the community policy options. Make sure the **comm-leaf_es1** policy option accepts target 9999:9999 to ensure that all virtual switches import the Type-1 ESI routes from all leaves.

```
[edit policy-options community]
```

```

user@core-1# set comm-VS_VLAN100 members target:1:100
user@core-1# set comm-VS_VLAN200 members target:1:200
user@core-1# set comm-VS_VLAN300 members target:1:300
user@core-1# set comm-VS_VLAN400 members target:1:400
user@core-1# set comm-leaf_es1 members target:9999:9999

```

9. Configure the IRB interfaces. Every IRB has a virtual gateway address, which is a shared MAC address and IP address across Core-1 and Core-2.

```
[edit interfaces irb]
```

```

user@core-1# set unit 1100 family inet address 172.16.0.2/24
virtual-gateway-address 100.0.0.1
user@core-1# set unit 1200 family inet address 172.16.0.3/24
virtual-gateway-address 200.0.0.1
user@core-1# set unit 1300 family inet address 10.10.10.2/24
virtual-gateway-address 10.10.10.1

```

```
user@core-1# set unit 1400 family inet address 10.10.10.2/24
virtual-gateway-address 10.10.10.1
```

10. Configure MP-BGP sessions towards Leaf-1 and Leaf-2.

```
[edit protocols bgp group EVPN_VXLAN]
user@core-1# set type external
user@core-1# set local-address 10.255.255.0
user@core-1# set family evpn signaling
user@core-1# set peer-as 65402
user@core-1# set multipath
user@core-1# set neighbor 192.0.2.0 description leaf-1
user@core-1# set neighbor 192.0.2.0 multihop ttl 255
user@core-1# set neighbor 10.255.255.5 description leaf-2
user@core-1# set neighbor 10.255.255.5 multihop ttl 255
```

Configuring Core-2

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Core-2:

1. Set the system hostname.

```
[edit]
user@core-2# set system host-name core-2
```

2. Configure the router ID and other routing options.

```
[edit routing-options]
user@core-2# set router-id 10.255.255.1
user@core-2# set autonomous-system 65400
user@core-2# set forwarding-table export load-balance
user@core-2# set forwarding-table ecmp-fast-reroute
```

3. Configure the load balance policy options.

```
[edit policy-options policy-statement load-balance]
user@core-2# set term 1 then load-balance per-packet
```

4. Configure the BGP underlay group.

```
[edit protocols bgp group underlay]
user@core-2# set type external
user@core-2# set advertise-peer-as
user@core-2# set family inet unicast loops 2
user@core-2# set export lo0
user@core-2# set peer-as 65401
user@core-2# set multipath
user@core-2# set neighbor 10.0.0.5 description spine-1
user@core-2# set neighbor 10.0.0.7 description spine-2
```

5. Configure the loopback address policy options.

```
[edit policy-options policy-statement lo0]

user@core-2# set from family inet
user@core-2# set from protocol direct
user@core-2# set lo0 from route-filter 0.0.0.0/0 prefix-length-range /32-/32
user@core-2# set lo0 then accept
```

6. A large portion of Core-1's configuration takes place in the **[routing-instance]** hierarchy. Configure the virtual routers and configure a unique VRF table import policy for each virtual switch.

```
[edit routing-instances]

user@core-2# set VRF_Tenant_A instance-type vrf
user@core-2# set VRF_Tenant_A interface irb.1100
user@core-2# set VRF_Tenant_A route-distinguisher 10.255.255.1:1100
user@core-2# set VRF_Tenant_A vrf-target target:10:100
user@core-2# set VRF_Tenant_A routing-options auto-export
user@core-2# set VRF_Tenant_B instance-type vrf
user@core-2# set VRF_Tenant_B interface irb.1200
user@core-2# set VRF_Tenant_B route-distinguisher 10.255.255.1:1200
user@core-2# set VRF_Tenant_B vrf-target target:10:200
user@core-2# set VRF_Tenant_C instance-type vrf
user@core-2# set VRF_Tenant_C interface irb.1300
user@core-2# set VRF_Tenant_C route-distinguisher 10.255.255.1:1300
user@core-2# set VRF_Tenant_C vrf-target target:10:300
user@core-2# set VRF_Tenant_D instance-type vrf
user@core-2# set VRF_Tenant_D interface irb.1400
user@core-2# set VRF_Tenant_D route-distinguisher 10.255.255.1:1400
user@core-2# set VRF_Tenant_D vrf-target target:10:400
user@core-2# set VS_VLAN100 vtep-source-interface lo0.0
user@core-2# set VS_VLAN100 instance-type virtual-switch
user@core-2# set VS_VLAN100 route-distinguisher 10.255.255.1:100
user@core-2# set VS_VLAN100 vrf-import VS_VLAN100_IMP
user@core-2# set VS_VLAN100 vrf-target target:1:100
user@core-2# set VS_VLAN100 protocols evpn encapsulation vxlan
user@core-2# set VS_VLAN100 protocols evpn extended-vni-list 1100
user@core-2# set VS_VLAN100 protocols evpn multicast-mode ingress-replication
user@core-2# set VS_VLAN100 protocols evpn default-gateway
no-gateway-community
user@core-2# set VS_VLAN100 bridge-domains bd1100 vlan-id 100
user@core-2# set VS_VLAN100 bridge-domains bd1100 routing-interface irb.1100
user@core-2# set VS_VLAN100 bridge-domains bd1100 vxlan vni 1100
user@core-2# set VS_VLAN100 bridge-domains bd1100 vxlan
ingress-node-replication
user@core-2# set VS_VLAN200 vtep-source-interface lo0.0
user@core-2# set VS_VLAN200 instance-type virtual-switch
user@core-2# set VS_VLAN200 route-distinguisher 10.255.255.1:200
user@core-2# set VS_VLAN200 vrf-import VS_VLAN200_IMP
user@core-2# set VS_VLAN200 vrf-target target:1:200
user@core-2# set VS_VLAN200 protocols evpn encapsulation vxlan
user@core-2# set VS_VLAN200 protocols evpn extended-vni-list 1200
user@core-2# set VS_VLAN200 protocols evpn multicast-mode ingress-replication
user@core-2# set VS_VLAN200 bridge-domains bd1200 vlan-id 200
user@core-2# set VS_VLAN200 bridge-domains bd1200 routing-interface irb.1200
user@core-2# set VS_VLAN200 bridge-domains bd1200 vxlan vni 1200
user@core-2# set VS_VLAN200 bridge-domains bd1200 vxlan
ingress-node-replication
user@core-2# set VS_VLAN300 vtep-source-interface lo0.0
```

```

user@core-2# set VS_VLAN300 instance-type virtual-switch
user@core-2# set VS_VLAN300 route-distinguisher 10.255.255.1:300
user@core-2# set VS_VLAN300 vrf-import VS_VLAN300_IMP
user@core-2# set VS_VLAN300 vrf-target target:1:300
user@core-2# set VS_VLAN300 protocols evpn encapsulation vxlan
user@core-2# set VS_VLAN300 protocols evpn extended-vni-list 1300
user@core-2# set VS_VLAN300 protocols evpn multicast-mode ingress-replication
user@core-2# set VS_VLAN300 bridge-domains bd1300 vlan-id 300
user@core-2# set VS_VLAN300 bridge-domains bd1300 routing-interface irb.1300
user@core-2# set VS_VLAN300 bridge-domains bd1300 vxlan vni 1300
user@core-2# set VS_VLAN300 bridge-domains bd1300 vxlan
ingress-node-replication
user@core-2# set VS_VLAN400 vtep-source-interface lo0.0
user@core-2# set VS_VLAN400 instance-type virtual-switch
user@core-2# set VS_VLAN400 route-distinguisher 10.255.255.1:400
user@core-2# set VS_VLAN400 vrf-import VS_VLAN400_IMP
user@core-2# set VS_VLAN400 vrf-target target:1:400
user@core-2# set VS_VLAN400 protocols evpn encapsulation vxlan
user@core-2# set VS_VLAN400 protocols evpn extended-vni-list 1400
user@core-2# set VS_VLAN400 protocols evpn multicast-mode ingress-replication
user@core-2# set VS_VLAN400 bridge-domains bd1400 vlan-id 400
user@core-2# set VS_VLAN400 bridge-domains bd1400 routing-interface irb.1400
user@core-2# set VS_VLAN400 bridge-domains bd1400 vxlan vni 1400
user@core-2# set VS_VLAN400 bridge-domains bd1400 vxlan
ingress-node-replication

```

7. Configure the individual policy statements.

```
[edit policy-options policy-statement]
```

```

user@core-2# set VS_VLAN100_IMP term ESI from community comm-leaf_esi
user@core-2# set VS_VLAN100_IMP term ESI then accept
user@core-2# set VS_VLAN100_IMP term VS_VLAN100 from community comm-VS_VLAN100
user@core-2# set VS_VLAN100_IMP term VS_VLAN100 then accept
user@core-2# set VS_VLAN200_IMP term ESI from community comm-leaf_esi
user@core-2# set VS_VLAN200_IMP term ESI then accept
user@core-2# set VS_VLAN200_IMP term VS_VLAN200 from community comm-VS_VLAN200
user@core-2# set VS_VLAN200_IMP term VS_VLAN200 then accept
user@core-2# set VS_VLAN300_IMP term ESI from community comm-leaf_esi
user@core-2# set VS_VLAN300_IMP term ESI then accept
user@core-2# set VS_VLAN300_IMP term VS_VLAN300 from community comm-VS_VLAN300
user@core-2# set VS_VLAN300_IMP term VS_VLAN300 then accept
user@core-2# set VS_VLAN400_IMP term ESI from community comm-leaf_esi
user@core-2# set VS_VLAN400_IMP term ESI then accept
user@core-2# set VS_VLAN400_IMP term VS_VLAN400 from community comm-VS_VLAN400
user@core-2# set VS_VLAN400_IMP term VS_VLAN400 then accept
user@core-2# set lo0 from family inet
user@core-2# set lo0 from protocol direct
user@core-2# set lo0 from route-filter 0.0.0.0/0 prefix-length-range /32-/32
user@core-2# set lo0 then accept

```

8. Configure the community policy options. Make sure the **comm-leaf_esi** policy option accepts target 9999:9999 to ensure that all virtual switches import the Type-1 ESI routes from all leafs.

```
[edit policy-options community]
```

```

user@core-2# set comm-VS_VLAN100 members target:1:100
user@core-2# set comm-VS_VLAN200 members target:1:200
user@core-2# set comm-VS_VLAN300 members target:1:300

```

```

user@core-2# set comm-VS_VLAN400 members target:1:400
user@core-2# set comm-leaf_esi members target:9999:9999

```

9. Configure the IRB interfaces. Every IRB has a virtual gateway address, which is a shared MAC address and IP address across Core-1 and Core-2.

```

[edit interfaces irb]

user@core-2# set unit 1100 family inet address 172.16.0.4/24
virtual-gateway-address 100.0.0.1
user@core-2# set unit 1200 family inet address 172.16.0.5/24
virtual-gateway-address 200.0.0.1
user@core-2# set unit 1300 family inet address 10.10.10.3/24
virtual-gateway-address 10.10.10.1
user@core-2# set unit 1400 family inet address 10.10.10.3/24
virtual-gateway-address 10.10.10.1

```

10. Configure MP-BGP sessions towards Leaf-1 and Leaf-2.

```

[edit protocols bgp group EVPN_VXLAN]

user@core-2# set type external
user@core-2# set local-address 10.255.255.1
user@core-2# set family evpn signaling
user@core-2# set peer-as 65402
user@core-2# set multipath
user@core-2# set neighbor 192.0.2.0 description leaf-1
user@core-2# set neighbor 192.0.2.0 multihop ttl 255
user@core-2# set neighbor 10.255.255.5 description leaf-2
user@core-2# set neighbor 10.255.255.5 multihop ttl 255

```

Verification

After you configure both the underlay and EVPN overlay we recommend that you verify that the configurations work as you intended.

- [Verifying Leaf-1 Configuration on page 56](#)
- [Verifying Leaf-2 Configuration on page 60](#)
- [Verifying Spine-1 Configuration on page 64](#)
- [Verifying Spine-2 Configuration on page 66](#)
- [Verifying Core-1 Configuration on page 68](#)
- [Verifying Core-2 Configuration on page 75](#)
- [Verifying MAC Reachability to a Single-Homed CE Device \(Leaf-1\) on page 82](#)
- [Verifying MAC Reachability to a Single-Homed CE Device \(Type-2\) on page 83](#)
- [Verifying the Imported Route on page 84](#)
- [Verifying the Layer 2 Address Learning Daemon Copy on page 85](#)
- [Verifying the Kernel-Level Forwarding Table on page 86](#)
- [Verifying MAC Reachability to a Multihomed CE Device on page 88](#)

- [Verifying EVPN, Layer 2 Address Learning Daemon, and the Kernel-Forwarding Tables on page 89](#)
- [Verifying an EVPN Anycast Gateway on page 93](#)

[Verifying Leaf-1 Configuration](#)

Purpose Verify that Leaf-1 is properly configured.

Action Verify that the routing options are properly configured.

```
user@leaf-1> show configuration routing-options
router-id 192.0.2.0;
autonomous-system 65402;
forwarding-table {
    export load-balance;
}
```

Verify that the load-balance policy statement is properly configured.

```
user@leaf-1> show configuration policy-options policy-statement
load-balance
term 1 {
    then {
        load-balance per-packet;
    }
}
```

Verify the underlay BGP group configuration.

```
user@leaf-1> show configuration protocols bgp group underlay
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65401;
multipath;
neighbor 10.0.0.8 {
    description spine-1;
}
neighbor 10.0.0.12 {
    description spine-2;
}
```

Verify that the loopback address is properly configured.

```
user@leaf-1> show configuration policy-options policy-statement lo0
from {
    family inet;
    protocol direct;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
}
then accept;
```

Verify the configuration of the switch options.

```
user@leaf-1> show configuration switch-options
vtep-source-interface lo0.0;
route-distinguisher 192.0.2.0:1;
vrf-import vrf-imp;
vrf-target target:9999:9999;
```

Verify the configuration of the VRF table import policy statement.

```
user@leaf-1> show configuration policy-options policy-statement
vrf-imp
term t1 {
    from community com100;
    then accept;
}
term t2 {
    from community com200;
    then accept;
}
term t3 {
    from community com300;
    then accept;
}
term t4 {
    from community com400;
    then accept;
}
term t5 {
    then reject;
}
```

Verify the policy options for GREP.

```
user@leaf-1> show configuration policy-options | grep members
community com100 members target:1:100;
community com200 members target:1:200;
community com300 members target:1:300;
community com400 members target:1:400;
```

Verify that EVPN is properly configured with VXLAN details.

```
user@leaf-1> show configuration protocols evpn
encapsulation vxlan;
extended-vni-list [ 1100 1200 1300 1400 ];
multicast-mode ingress-replication;
vni-routing-options {
    vni 1100 {
        vrf-target export target:1:100;
    }
    vni 1200 {
        vrf-target export target:1:200;
    }
    vni 1300 {
        vrf-target export target:1:300;
    }
    vni 1400 {
        vrf-target export target:1:400;
    }
}
```

Verify the EVPN-VXLAN Core BGP group configuration.

```
user@leaf-1> show configuration protocols bgp group EVPN_VXLAN_CORE
type external;
multihop {
```

```

        ttl 255;
        no-nexthop-change;
    }
    local-address 192.0.2.0;
    family evpn {
        signaling;
    }
    peer-as 65400;
    neighbor 10.255.255.0 {
        description core-1;
    }
    neighbor 10.255.255.1 {
        description core-2;
    }
}

```

Verify the EVPN-VXLAN Leaf BGP group configuration.

```

user@leaf-1> show configuration protocols bgp group EVPN_VXLAN_LEAF
type internal;
local-address 192.0.2.0;
family evpn {
    signaling;
}
neighbor 10.255.255.5 {
    description leaf-2;
}

```

Verify the configuration for the xe-0/0/32 interface.

```

user@leaf-1> show configuration interfaces xe-0/0/32
ether-options {
    802.3ad ae0;
}

```

Verify the configuration for the xe-0/0/33 interface.

```

user@leaf-1> show configuration interfaces xe-0/0/33
ether-options {
    802.3ad ae1;
}

```

Verify the configuration for the ae0 interface.

```

user@leaf-1> show configuration interfaces ae0
esi {
    00:01:01:01:01:01:01:01;
    all-active;
}
aggregated-ether-options {
    lacp {
        active;
        system-id 00:00:00:01:01:01;
    }
}
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {

```

```
        members v100;
    }
}
```

Verify the configuration for the ae1 interface.

```
user@leaf-1> show configuration interfaces ae1
esi {
    00:02:02:02:02:02:02:02:02;
    all-active;
}
aggregated-ether-options {
    lacp {
        active;
        system-id 00:00:00:01:01:01;
    }
}
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members v200;
        }
    }
}
```

Verifying Leaf-2 Configuration

Purpose Verify that Leaf-2 is properly configured.

Action Verify that the routing options are properly configured.

```
user@leaf-2> show configuration routing-options
router-id 10.255.255.5;
autonomous-system 65402;
forwarding-table {
    export load-balance;
}
```

Verify that the load-balance policy statement is properly configured.

```
user@leaf-2> show configuration policy-options policy-statement
load-balance
term 1 {
    then {
        load-balance per-packet;
    }
}
```

Verify the underlay BGP group configuration.

```
user@leaf-2> show configuration protocols bgp group underlay
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65401;
multipath;
neighbor 10.0.0.10 {
    description spine-1;
}
neighbor 10.0.0.14 {
    description spine-2;
}
```

Verify that the loopback address is properly configured.

```
user@leaf-2> show configuration policy-options policy-statement lo0
from {
    family inet;
    protocol direct;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
}
then accept;
```

Verify the configuration of the switch options.

```
user@leaf-2> show configuration switch-options
vtep-source-interface lo0.0;
route-distinguisher 10.255.255.5:1;
vrf-import vrf-imp;
vrf-target target:9999:9999;
```

Verify the configuration of the VRF table import policy statement.

```
user@leaf-2> show configuration policy-options policy-statement
vrf-imp
term t1 {
    from community com100;
    then accept;
}
term t2 {
    from community com200;
    then accept;
}
term t3 {
    from community com300;
    then accept;
}
term t4 {
    from community com400;
    then accept;
}
term t5 {
    then reject;
}
```

Verify the policy options for GREP.

```
user@leaf-2> show configuration policy-options | grep members
community com100 members target:1:100;
community com200 members target:1:200;
community com300 members target:1:300;
community com400 members target:1:400;
```

Verify that EVPN is properly configured with VXLAN details.

```
user@leaf-2> show configuration protocols evpn
encapsulation vxlan;
extended-vni-list [ 1100 1200 1300 1400 ];
multicast-mode ingress-replication;
vni-routing-options {
    vni 1100 {
        vrf-target export target:1:100;
    }
    vni 1200 {
        vrf-target export target:1:200;
    }
    vni 1300 {
        vrf-target export target:1:300;
    }
    vni 1400 {
        vrf-target export target:1:400;
    }
}
```

Verify the EVPN-VXLAN Core BGP group configuration.

```
user@leaf-2> show configuration protocols bgp group EVPN_VXLAN_CORE
type external;
multihop {
```

```

        ttl 255;
        no-nexthop-change;
    }
    local-address 10.255.255.5;
    family evpn {
        signaling;
    }
    peer-as 65400;
    neighbor 10.255.255.0 {
        description core-1;
    }
    neighbor 10.255.255.1 {
        description core-2;
    }
}

```

Verify the EVPN-VXLAN Leaf BGP group configuration.

```

user@leaf-2> show configuration protocols bgp group EVPN_VXLAN_LEAF
type internal;
local-address 192.0.2.0;
family evpn {
    signaling;
}
neighbor 10.255.255.5 {
    description leaf-2;
}

```

Verify the configuration for the xe-0/0/36 interface.

```

user@leaf-2> show configuration interfaces xe-0/0/36
ether-options {
    802.3ad ae0;
}

```

Verify the configuration for the xe-0/0/37 interface.

```

user@leaf-2> show configuration interfaces xe-0/0/37
ether-options {
    802.3ad ae1;
}

```

Verify the configuration for the ae0 interface.

```

user@leaf-2> show configuration interfaces ae0
esi {
    00:01:01:01:01:01:01:01;
    all-active;
}
aggregated-ether-options {
    lacp {
        passive;
        system-id 00:00:00:01:01:01;
    }
}
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {

```

```
        members v100;
    }
}
```

Verify the configuration for the ae1 interface.

```
user@leaf-2> show configuration interfaces ae1
esi {
    00:02:02:02:02:02:02:02:02;
    all-active;
}
aggregated-ether-options {
    lacp {
        passive;
        system-id 00:00:00:01:01:01;
    }
}
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members v200;
        }
    }
}
```

Verifying Spine-1 Configuration

Purpose Verify that Spine-1 is properly configured.

Action Verify that the routing options are properly configured.

```
user@spine-1> show configuration routing-options
```

```
router-id 10.255.255.2;
autonomous-system 65401;
forwarding-table {
    export load-balance;
    ecmp-fast-reroute;
}
```

Verify that the load-balance policy statement is properly configured.

```
user@spine-1> show configuration policy-options policy-statement load-balance
```

```
term 1 {
    then {
        load-balance per-packet;
    }
}
```

Verify the underlay Leaf BGP group configuration.

```
user@spine-1> show configuration protocols bgp group underlay-leaf
```

```
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65402;
multipath;
neighbor 10.0.0.9 {
    description leaf-1;
}
neighbor 10.0.0.11 {
    description leaf-2;
}
```

Verify the underlay Core BGP group configuration.

```
user@spine-1> show configuration protocols bgp group underlay-core
```

```
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65400;
multipath;
neighbor 10.0.0.0 {
    description core-1;
}
```

```
}  
neighbor 10.0.0.4 {  
    description core-2;  
}
```

Verify that the loopback address is properly configured.

```
user@spine-1> show configuration policy-options policy-statement lo0  
  
from {  
    family inet;  
    protocol direct;  
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;  
}  
then accept;
```

Verifying Spine-2 Configuration

Purpose Verify that Spine-2 is properly configured.

Action Verify that the routing options are properly configured.

```
user@spine-2> show configuration routing-options
```

```
router-id 10.255.255.3;
autonomous-system 65401;
forwarding-table {
    export load-balance;
    ecmp-fast-reroute;
}
```

Verify that the load-balance policy statement is properly configured.

```
user@spine-2> show configuration policy-options policy-statement load-balance
```

```
term 1 {
    then {
        load-balance per-packet;
    }
}
```

Verify the underlay Leaf BGP group configuration.

```
user@spine-2> show configuration protocols bgp group underlay-leaf
```

```
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65402;
multipath;
neighbor 10.0.0.13 {
    description leaf-1;
}
neighbor 10.0.0.15 {
    description leaf-2;
}
```

Verify the underlay Core BGP group configuration.

```
user@spine-2> show configuration protocols bgp group underlay-core
```

```
type external;
advertise-peer-as;
family inet {
    unicast {
        loops 2;
    }
}
export lo0;
peer-as 65400;
multipath;
neighbor 10.0.0.2 {
    description core-1;
}
```

```
}  
neighbor 10.0.0.6 {  
    description core-2;  
}
```

Verify that the loopback address is properly configured.

```
user@spine-2> show configuration policy-options policy-statement lo0
```

```
from {  
    family inet;  
    protocol direct;  
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;  
}  
then accept;
```

Verifying Core-1 Configuration

Purpose Verify that Core-1 is properly configured.

Action Verify that the routing options are properly configured.

```
user@core-1> show configuration routing-options
router-id 10.255.255.0;
autonomous-system 65400;
forwarding-table {
    export load-balance;
    ecmp-fast-reroute;
}
```

Verify that the load-balance policy statement is properly configured.

```
user@core-1> show configuration policy-options policy-statement load-balance
term 1 {
    then {
        load-balance per-packet;
    }
}
```

Verify the underlay BGP group configuration.

```
user@core-1> show configuration protocols bgp
group underlay {
    type external;
    advertise-peer-as;
    family inet {
        unicast {
            loops 2;
        }
    }
    export lo0;
    peer-as 65401;
    multipath;
    neighbor 10.0.0.1 {
        description spine-1;
    }
    neighbor 10.0.0.3 {
        description spine-2;
    }
}
```

Verify that the loopback address is properly configured.

```
user@core-1> show configuration policy-options policy-statement lo0
from {
    family inet;
    protocol direct;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
}
then accept;
```

Verify that the routing instances are properly configured.

```
user@core-1> show configuration routing-instances
VRF_Tenant_A {
    instance-type vrf;
    interface irb.1100;
```

```
    route-distinguisher 10.255.255.0:1100;
    vrf-target target:10:100;
}
VRF_Tenant_B {
    instance-type vrf;
    interface irb.1200;
    route-distinguisher 10.255.255.0:1200;
    vrf-target target:10:200;
}
VRF_Tenant_C {
    instance-type vrf;
    interface irb.1300;
    route-distinguisher 10.255.255.0:1300;
    vrf-target target:10:300;
}
VRF_Tenant_D {
    instance-type vrf;
    interface irb.1400;
    route-distinguisher 10.255.255.0:1400;
    vrf-target target:10:400;
}
VS_VLAN100 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.0:100;
    vrf-import VS_VLAN100_IMP;
    vrf-target target:1:100;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1100;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1100 {
            vlan-id 100;
            routing-interface irb.1100;
            vxlan {
                vni 1100;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN200 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.0:200;
    vrf-import VS_VLAN200_IMP;
    vrf-target target:1:200;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1200;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1200 {
            vlan-id 200;
        }
    }
}
```

```

        routing-interface irb.1200;
        vxlan {
            vni 1200;
            ingress-node-replication;
        }
    }
}
VS_VLAN300 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.0:300;
    vrf-import VS_VLAN300_IMP;
    bridge-domains {
        bd1100 {
            vlan-id 100;
            routing-interface irb.1100;
            vxlan {
                vni 1100;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN200 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.0:200;
    vrf-import VS_VLAN200_IMP;
    vrf-target target:1:200;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1200;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1200 {
            vlan-id 200;
            routing-interface irb.1200;
            vxlan {
                vni 1200;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN300 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.0:300;
    vrf-import VS_VLAN300_IMP;
    vrf-target target:1:300;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1300;
            multicast-mode ingress-replication;
        }
    }
}

```

```

    bridge-domains {
        bd1300 {
            vlan-id 300;
            routing-interface irb.1300;
            vxlan {
                vni 1300;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN400 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.0:400;
    vrf-import VS_VLAN400_IMP;
    vrf-target target:1:400;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1400;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1400 {
            vlan-id 400;
            routing-interface irb.1400;
            vxlan {
                vni 1400;
                ingress-node-replication;
            }
        }
    }

    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1300;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1300 {
            vlan-id 300;
            routing-interface irb.1300;
            vxlan {
                vni 1300;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN400 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.0:400;
    vrf-import VS_VLAN400_IMP;
    vrf-target target:1:400;
    protocols {
        evpn {

```



```

        encapsulation vxlan;
        extended-vni-list 1400;
        multicast-mode ingress-replication;
    }
}
bridge-domains {
    bd1400 {
        vlan-id 400;
        routing-interface irb.1400;
        vxlan {
            vni 1400;
            ingress-node-replication;
        }
    }
}

```

Verify that the policy options are properly configured.

```

user@core-1> show configuration policy-options
policy-statement VS_VLAN100_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN100 {
        from community comm-VS_VLAN100;
        then accept;
    }
}
policy-statement VS_VLAN200_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN200 {
        from community comm-VS_VLAN200;
        then accept;
    }
}
policy-statement VS_VLAN300_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN300 {
        from community comm-VS_VLAN300;
        then accept;
    }
}
policy-statement VS_VLAN400_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN400 {
        from community comm-VS_VLAN400;
        then accept;
    }
}
policy-statement lo0 {

```

```
    from {
        family inet;
        protocol direct;
        route-filter 0.0.0.0/0 prefix-length-range /32-/32;
    }
    then accept;
}
policy-statement load-balance {
    term 1 {
        then {
            load-balance per-packet;
        }
    }
}
community comm-VS_VLAN100 members target:1:100;
community comm-VS_VLAN200 members target:1:200;
community comm-VS_VLAN300 members target:1:300;
community comm-VS_VLAN400 members target:1:400;
community comm-leaf-esi members target:9999:9999;
```

Verify the configuration of the IRB interface.

```
user@core-1> show configuration interfaces irb
unit 1100 {
    family inet {
        address 172.16.0.2/24 {
            virtual-gateway-address 100.0.0.1;
        }
    }
}
unit 1200 {
    family inet {
        address 172.16.0.3/24 {
            virtual-gateway-address 200.0.0.1;
        }
    }
}
unit 1300 {
    family inet {
        address 10.10.10.2/24 {
            virtual-gateway-address 10.10.10.1;
        }
    }
}
unit 1400 {
    family inet {
        address 10.10.10.2/24 {
            virtual-gateway-address 10.10.10.1;
        }
    }
}
```

Verify the configuration of the EVPN-VXLAN BGP group.

```
user@core-1> show configuration protocols bgp group EVPN_VXLAN
type external;
local-address 10.255.255.0;
family evpn {
    signaling;
}
```

```
peer-as 65402;
multipath;
neighbor 192.0.2.0 {
    description leaf-1;
    multihop {
        ttl 255;
    }
}
neighbor 10.255.255.5 {
    description leaf-2;
    multihop {
        ttl 255;
    }
}
```

Verifying Core-2 Configuration

Purpose Verify that Core-2 is properly configured.

Action Verify that the routing options are properly configured.

```
user@core-2> show configuration routing-options
router-id 10.255.255.1;
autonomous-system 65400;
forwarding-table {
    export load-balance;
    ecmp-fast-reroute;
}
```

Verify that the load-balance policy statement is properly configured.

```
user@core-2> show configuration policy-options policy-statement load-balance
term 1 {
    then {
        load-balance per-packet;
    }
}
```

Verify the underlay BGP group configuration.

```
user@core-2> show configuration protocols bgp
group underlay {
    type external;
    advertise-peer-as;
    family inet {
        unicast {
            loops 2;
        }
    }
    export lo0;
    peer-as 65401;
    multipath;
    neighbor 10.0.0.5 {
        description spine-1;
    }
    neighbor 10.0.0.7 {
        description spine-2;
    }
}
```

Verify that the loopback address is properly configured.

```
user@core-2> show configuration policy-options policy-statement lo0
from {
    family inet;
    protocol direct;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
}
then accept;
```

Verify that the routing instances are properly configured.

```
user@core-2> show configuration routing-instances
VRF_Tenant_A {
    instance-type vrf;
    interface irb.1100;
```

```

        route-distinguisher 10.255.255.1:1100;
        vrf-target target:10:100;
    }
    VRF_Tenant_B {
        instance-type vrf;
        interface irb.1200;
        route-distinguisher 10.255.255.1:1200;
        vrf-target target:10:200;
    }
    VRF_Tenant_C {
        instance-type vrf;
        interface irb.1300;
        route-distinguisher 10.255.255.1:1300;
        vrf-target target:10:300;
    }
    VRF_Tenant_D {
        instance-type vrf;
        interface irb.1400;
        route-distinguisher 10.255.255.1:1400;
        vrf-target target:10:400;
    }
    VS_VLAN100 {
        vtep-source-interface lo0.0;
        instance-type virtual-switch;
        route-distinguisher 10.255.255.1:100;
        vrf-import VS_VLAN100_IMP;
        vrf-target target:1:100;
        protocols {
            evpn {
                encapsulation vxlan;
                extended-vni-list 1100;
                multicast-mode ingress-replication;
            }
        }
        bridge-domains {
            bd1100 {
                vlan-id 100;
                routing-interface irb.1100;
                vxlan {
                    vni 1100;
                    ingress-node-replication;
                }
            }
        }
    }
    VS_VLAN200 {
        vtep-source-interface lo0.0;
        instance-type virtual-switch;
        route-distinguisher 10.255.255.1:200;
        vrf-import VS_VLAN200_IMP;
        vrf-target target:1:200;
        protocols {
            evpn {
                encapsulation vxlan;
                extended-vni-list 1200;
                multicast-mode ingress-replication;
            }
        }
        bridge-domains {
            bd1200 {
                vlan-id 200;
            }
        }
    }

```

```
        routing-interface irb.1200;
        vxlan {
            vni 1200;
            ingress-node-replication;
        }
    }
}
VS_VLAN300 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.1:300;
    vrf-import VS_VLAN300_IMP;
    bridge-domains {
        bd1100 {
            vlan-id 100;
            routing-interface irb.1100;
            vxlan {
                vni 1100;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN200 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.1:200;
    vrf-import VS_VLAN200_IMP;
    vrf-target target:1:200;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1200;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1200 {
            vlan-id 200;
            routing-interface irb.1200;
            vxlan {
                vni 1200;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN300 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.1:300;
    vrf-import VS_VLAN300_IMP;
    vrf-target target:1:300;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1300;
            multicast-mode ingress-replication;
        }
    }
}
```

```

    bridge-domains {
        bd1300 {
            vlan-id 300;
            routing-interface irb.1300;
            vxlan {
                vni 1300;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN400 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.1:400;
    vrf-import VS_VLAN400_IMP;
    vrf-target target:1:400;
    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1400;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1400 {
            vlan-id 400;
            routing-interface irb.1400;
            vxlan {
                vni 1400;
                ingress-node-replication;
            }
        }
    }

    protocols {
        evpn {
            encapsulation vxlan;
            extended-vni-list 1300;
            multicast-mode ingress-replication;
        }
    }
    bridge-domains {
        bd1300 {
            vlan-id 300;
            routing-interface irb.1300;
            vxlan {
                vni 1300;
                ingress-node-replication;
            }
        }
    }
}
VS_VLAN400 {
    vtep-source-interface lo0.0;
    instance-type virtual-switch;
    route-distinguisher 10.255.255.1:400;
    vrf-import VS_VLAN400_IMP;
    vrf-target target:1:400;
    protocols {
        evpn {

```

```

        encapsulation vxlan;
        extended-vni-list 1400;
        multicast-mode ingress-replication;
    }
}
bridge-domains {
    bd1400 {
        vlan-id 400;
        routing-interface irb.1400;
        vxlan {
            vni 1400;
            ingress-node-replication;
        }
    }
}

```

Verify that the policy options are properly configured.

```

user@core-2> show configuration policy-options
policy-statement VS_VLAN100_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN100 {
        from community comm-VS_VLAN100;
        then accept;
    }
}
policy-statement VS_VLAN200_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN200 {
        from community comm-VS_VLAN200;
        then accept;
    }
}
policy-statement VS_VLAN300_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN300 {
        from community comm-VS_VLAN300;
        then accept;
    }
}
policy-statement VS_VLAN400_IMP {
    term ESI {
        from community comm-leaf_esi;
        then accept;
    }
    term VS_VLAN400 {
        from community comm-VS_VLAN400;
        then accept;
    }
}
policy-statement lo0 {

```



```

    from {
        family inet;
        protocol direct;
        route-filter 0.0.0.0/0 prefix-length-range /32-/32;
    }
    then accept;
}
policy-statement load-balance {
    term 1 {
        then {
            load-balance per-packet;
        }
    }
}
community comm-VS_VLAN100 members target:1:100;
community comm-VS_VLAN200 members target:1:200;
community comm-VS_VLAN300 members target:1:300;
community comm-VS_VLAN400 members target:1:400;
community comm-leaf-esi members target:9999:9999;

```

Verify the configuration of the IRB interface.

```

user@core-2> show configuration interfaces irb
unit 1100 {
    family inet {
        address 172.16.0.4/24 {
            virtual-gateway-address 100.0.0.1;
        }
    }
}
unit 1200 {
    family inet {
        address 172.16.0.5/24 {
            virtual-gateway-address 200.0.0.1;
        }
    }
}
unit 1300 {
    family inet {
        address 10.10.10.3/24 {
            virtual-gateway-address 10.10.10.1;
        }
    }
}
unit 1400 {
    family inet {
        address 10.10.10.3/24 {
            virtual-gateway-address 10.10.10.1;
        }
    }
}

```

Verify the configuration of the EVPN-VXLAN BGP group.

```

user@core-2> show configuration protocols bgp group EVPN_VXLAN
type external;
local-address 10.255.255.1;
family evpn {
    signaling;
}

```

```

peer-as 65402;
multipath;
neighbor 192.0.2.0 {
    description leaf-1;
    multihop {
        ttl 255;
    }
}
neighbor 10.255.255.5 {
    description leaf-2;
    multihop {
        ttl 255;
    }
}

```

Verifying MAC Reachability to a Single-Homed CE Device (Leaf-1)

Purpose Verify MAC reachability to Tenant_C single-homed to Leaf-1. First you need to verify that the MAC address is learned locally on Leaf-1. Leaf-1 generates the Type-2 route only after it has learned the MAC address.

Action Verify that the MAC address is learned locally on Leaf-1.

```
lab@leaf-1> show ethernet-switching table vlan-id 300
```

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
SE - statistics enabled, NM - non configured MAC, R - remote PE MAC,
O - ovsdb MAC)

Ethernet switching table : 5 entries, 5 learned

Routing instance : default-switch

Vlan	MAC	MAC	Logical	Active
name	address	flags	interface	source
v300	00:00:5e:00:01:01	DR	esi.1726	
05:00:00:ff:78:00:00:05:14:00				
v300	00:21:59:c8:24:65	D	xe-0/0/34.0	
v300	00:21:59:c8:24:69	D	vtep.32771	
10.255.255.5				
v300	40:a6:77:9a:43:f0	D	vtep.32769	
10.255.255.0				
v300	40:a6:77:9a:47:f0	D	vtep.32770	
10.255.255.1				

Meaning The output shows that MAC 00:21:59:c8:24:65 is successfully learned towards the Tenant_C CE device on xe-0/0/34.0, and from the output below, we can see that we generate the Type-2 route to Core-1.

Verifying MAC Reachability to a Single-Homed CE Device (Type-2)

Purpose Verify MAC reachability to a single-homed CE device (Type-2)

Action Verify the generation of the Type-2 route to Core-1.

```
lab@leaf-1> show route advertising-protocol bgp 10.255.255.0 evpn-mac-address
00:21:59:c8:24:65

bgp.evpn.0: 77 destinations, 77 routes (77 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref    AS path
  2:192.0.2.0:1::1300::00:21:59:c8:24:65/304
  *                               Self                               I
  2:192.0.2.0:1::1400::00:21:59:c8:24:65/304
  *                               Self                               I

default-switch.evpn.0: 67 destinations, 67 routes (67 active, 0 holddown, 0 hidden)

  Prefix      Nexthop      MED      Lclpref    AS path
  2:192.0.2.0:1::1300::00:21:59:c8:24:65/304
  *                               Self                               I
  2:192.0.2.0:1::1400::00:21:59:c8:24:65/304
  *                               Self                               I

__default_evpn__.evpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
```

Meaning The output shows that the same MAC address is advertised twice, with two different route targets. On xe-0/0/34, Tenant_C and Tenant_D are on the same physical server, running on different virtual machines. Tenant isolation is preserved by advertising these MACs on different VNIs and RTs.

On Core-1, the Type-2 route is received into bgp.evpn.0.

```
lab@core-1> show route receive-protocol bgp 192.0.2.0 evpn-mac-address
00:21:59:c8:24:65 extensive table bgp.evpn.0

[output omitted]
* 2:192.0.2.0:1::1300::00:21:59:c8:24:65/304 (2 entries, 0 announced)
  Import Accepted
  Route Distinguisher: 192.0.2.0:1
  Nexthop: 192.0.2.0
  AS path: 65402 I
  Communities: target:1:300 encapsulation0:0:0:0:vxlan

* 2:192.0.2.0:1::1400::00:21:59:c8:24:65/304 (2 entries, 0 announced)
  Import Accepted
  Route Distinguisher: 192.0.2.0:1
  Nexthop: 192.0.2.0
  AS path: 65402 I
  Communities: target:1:400 encapsulation0:0:0:0:vxlan
```

The output shows two Type-2 routes for 00:21:59:c8:24:65, but with different route targets. The route distinguisher is from Leaf-1, set as 192.0.2.0:1.

Verifying the Imported Route

Purpose Verify that the Type-2 route is imported.

Action On Core-1, verify whether this Type-2 route is successfully imported from the bgp.evpn.0 table into the EVPN switch instance.

Meaning The output shows that, in Tenant_C's virtual switch, the Type-2 route is advertised with the correct target, target:1:1300.

```
lab@core-1> show route table VS_VLAN300.evpn.0 evpn-mac-address 00:21:59:c8:24:65
| grep 00:21:59:c8:24:65
2:192.0.2.0:1::1300::00:21:59:c8:24:65/304
```

You can use the **extensive** option to review the Type-2 route in greater detail.

```
lab@core-1> show route table VS_VLAN300.evpn.0 evpn-mac-address 00:21:59:c8:24:65
extensive
```

[output omitted]

```
2:192.0.2.0:1::1300::00:21:59:c8:24:65/304 (2 entries, 1 announced)
  *BGP    Preference: 170/-101
          Route Distinguisher: 192.0.2.0:1
          Next hop type: Indirect
          Address: 0x2cf479c
          Next-hop reference count: 76
          Source: 192.0.2.0
          Protocol next hop: 192.0.2.0
          Indirect next hop: 0x2 no-forward INH Session ID: 0x0
          State: [Secondary Active Ext]
          Local AS: 65400 Peer AS: 65402
          Age: 4:47 Metric2: 0
          Validation State: unverified
          Task: BGP_65402.192.0.2.0+179
          Announcement bits (1): 0-VS_VLAN300-evpn
          AS path: 65402 I
          Communities: target:1:300 encapsulation0:0:0:0:vxlan
          Import Accepted
          Localpref: 100
          Router ID: 192.0.2.0
          Primary Routing Table bgp.evpn.0
          Indirect next hops: 1
            Protocol next hop: 192.0.2.0
            Indirect next hop: 0x2 no-forward INH Session ID: 0x0
            Indirect path forwarding next hops: 2
              Next hop type: Router
              Next hop: 10.0.0.1 via ge-1/0/0.0
              Session Id: 0x140
              Next hop: 10.0.0.3 via ge-1/0/1.0
              Session Id: 0x141
192.0.2.0/24 Originating RIB: inet.0
  Node path count: 1
  Forwarding nexthops: 2
  Nexthop: 10.0.0.1 via ge-1/0/0.0
    BGP    Preference: 170/-101
```

```

Route Distinguisher: 192.0.2.0:1
Next hop type: Indirect
Address: 0x2cf479c
Next-hop reference count: 76
Source: 10.255.255.5
Protocol next hop: 192.0.2.0
Indirect next hop: 0x2 no-forward INH Session ID: 0x0
State: [Secondary NotBest Ext]
Inactive reason: Not Best in its group - Router ID
Local AS: 65400 Peer AS: 65402
Age: 4:47 Metric2: 0
Validation State: unverified
Task: BGP_65402.10.255.255.5+61407
AS path: 65402 I
Communities: target:1:300 encapsulation0:0:0:0:vxlan
Import Accepted
Localpref: 100
Router ID: 10.255.255.5
Primary Routing Table bgp.evpn.0
Indirect next hops: 1
  Protocol next hop: 192.0.2.0
  Indirect next hop: 0x2 no-forward INH Session ID: 0x0
  Indirect path forwarding next hops: 2
    Next hop type: Router
    Next hop: 10.0.0.1 via ge-1/0/0.0
    Session Id: 0x140
    Next hop: 10.0.0.3 via ge-1/0/1.0
    Session Id: 0x141
192.0.2.0/24 Originating RIB: inet.0
  Node path count: 1
  Forwarding nexthops: 2
  Nexthop: 10.0.0.1 via ge-1/0/0.0

```

The output shows that Core-1 receives two copies. The first is the direct advertisement from Leaf-1 (Source: 192.0.2.0). The second is the indirect advertisement from Leaf-1 -> Leaf-2 -> Core-1 (Source: 10.255.255.5). Because we configured **no-nexthop-change** on Leaf-2, the protocol next hop of 192.0.2.0 is correct.

Verifying the Layer 2 Address Learning Daemon Copy

- | | |
|----------------|--|
| Purpose | Verify the Layer 2 address learning daemon copy. |
| Action | Verify the Layer 2 address learning daemon copy by entering the show bridge-mac table command. |
| Meaning | The output shows that 00:21:59:c8:24:65 is reachable through the vtep.32774 logical interface to Leaf-1. |

```
lab@core-1> show bridge mac-table instance VS_VLAN300
```

```

MAC flags      (S -static MAC, D -dynamic MAC, L -locally learned, C -Control
MAC
               0 -OVSDB MAC, SE -Statistics enabled, NM -Non configured MAC, R -Remote PE
MAC)

```

```

Routing instance : VS_VLAN300
Bridging domain : bd1300, VLAN : 300
  MAC          MAC          Logical          Active
  address      flags        interface      source

00:21:59:c8:24:65  D          vtep.32774    192.0.2.0
00:21:59:c8:24:69  D          vtep.32783    10.255.255.5

```



NOTE: On EX9200 switches, the `show ethernet-switching table-instance instance-name` command corresponds to the `show bridge mac-table instance instance-name` command on MX Series routers

Verifying the Kernel-Level Forwarding Table

- Purpose** Verify the kernel-level forwarding table, next hop identifier, and Layer 2 MAC table and hardware.
- Action** Query the kernel-level forwarding table, correlate the index next hop identifier with the correct virtual network identifier, and review the Layer 2 MAC table and hardware.
- Meaning** Tenant_C's MAC, 00:21:59:c8:24:65, is reachable through index 699.

```

lab@core-1> show route forwarding-table family bridge vpn VS_VLAN300
Routing table: VS_VLAN300.evpn-vxlan
VPLS:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
default          perm  0          dscd          540    1
vtep.32774       intf  0          comp          699    7
vtep.32783       intf  0          comp          714    5

```

```

Routing table: VS_VLAN300.evpn-vxlan
Bridging domain: bd1300.evpn-vxlan
VPLS:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
00:21:59:c8:24:65/48 user  0          comp          699    7
00:21:59:c8:24:69/48 user  0          comp          714    5
0x30003/51       user  0          comp          706    2

```

Correlate index 699 (NH-Id) with the correct virtual network identifier 1300 and remote VTEP-ID of 192.0.2.0.

```

lab@core-1> show l2-learning vxlan-tunnel-end-point remote
Logical System Name      Id  SVTEP-IP      IFL  L3-Idx
[default]
RVTEP-IP                IFL-Idx  NH-Id
192.0.2.0                351      697
  VNID                   MC-Group-IP
  1100                   0.0.0.0
RVTEP-IP                IFL-Idx  NH-Id
10.255.255.5            356      711

```

```

VNID      MC-Group-IP
1100      0.0.0.0
RVTEP-IP  IFL-Idx  NH-Id
192.0.2.0 352      698
VNID      MC-Group-IP
1200      0.0.0.0
RVTEP-IP  IFL-Idx  NH-Id
10.255.255.5 355      710
VNID      MC-Group-IP
1200      0.0.0.0
RVTEP-IP  IFL-Idx  NH-Id
192.0.2.0 353      699
VNID      MC-Group-IP
1300      0.0.0.0
RVTEP-IP  IFL-Idx  NH-Id
10.255.255.5 357      714
VNID      MC-Group-IP
1300      0.0.0.0
RVTEP-IP  IFL-Idx  NH-Id
192.0.2.0 354      700
VNID      MC-Group-IP
1400      0.0.0.0
RVTEP-IP  IFL-Idx  NH-Id
10.255.255.5 358      709
VNID      MC-Group-IP
1400      0.0.0.0

```



NOTE: On EX9200 switches, the `show ethernet-switching` command corresponds to the `show l2-learning` command on MX Series routers.

The output shows that 00:21:59:c8:24:65 is successfully programmed in Packet Forwarding Engine hardware.

```

# show l2 manager mac-table
[output omitted]
route table name : VS_VLAN300.7
mac counters
  maximum count
  0          2
mac table information
mac address      BD      learn  Entry  entry  ha1    hardware info
                  Index  vlan   Flags  ifl    ifl    pfe  mask  ifl
-----
00:21:59:c8:24:65 4      0      0x0014 vtep.32774 vtep.32774 0  -D   src
unknown dest vtep.32774
00:21:59:c8:24:69 4      0      0x0014 vtep.32783 vtep.32783 0  -D   src
unknown dest vtep.32783
Displayed 2 entries for routing instance VS_VLAN300.7

```



NOTE: On EX9200 switches, the `show ethernet-switching` command corresponds to the `show l2-learning` command on MX Series routers.

Verifying MAC Reachability to a Multihomed CE Device

Purpose Verify MAC reachability to the multihomed Tenant_B CE device on Leaf-1 and Leaf-2.

Action Verify that Leaf-1 and Leaf-2 are advertising both Type-1 and Type-2 reachability towards the multihomed CE device.

```
lab@leaf-1> show ethernet-switching table vlan-id 200
```

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static

SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)

Ethernet switching table : 4 entries, 4 learned

Routing instance : default-switch

Vlan	MAC	MAC	Logical	Active
name	address	flags	interface	source
v200	00:00:5e:00:01:01	DR	esi.1727	
05:00:00:ff:78:00:00:04:b0:00				
v200	00:21:59:c8:24:64	DL	ae1.0	
v200	40:a6:77:9a:43:f0	D	vtep.32769	
10.255.255.0				
v200	40:a6:77:9a:47:f0	D	vtep.32770	
10.255.255.1				

```
lab@leaf-2> show ethernet-switching table vlan-id 200
```

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static

SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)

Ethernet switching table : 5 entries, 5 learned

Routing instance : default-switch

Vlan	MAC	MAC	Logical	Active
name	address	flags	interface	source
v200	00:00:5e:00:01:01	DR	esi.1727	
05:00:00:ff:78:00:00:04:b0:00				
v200	00:21:59:c8:24:41	DL	ae1.0	
v200	00:21:59:c8:24:68	DL	ae1.0	
v200	40:a6:77:9a:43:f0	D	vtep.32770	
10.255.255.0				
v200	40:a6:77:9a:47:f0	D	vtep.32769	
10.255.255.1				

Verify that interface ae1 belongs on ESI 00:02:02:02:02:02:02:02:02.

```
lab@leaf-1> show configuration interfaces ae1 esi
00:02:02:02:02:02:02:02:02;
all-active;
```


Meaning The output shows that 00:21:59:c8:24:64 represents the physical MAC of the Tenant_B NIC physically attached to Leaf-1. 00:21:59:c8:24:68 represents the physical MAC of the Tenant_B NIC physically attached to Leaf-2. 00:21:59:c8:24:41 is a virtual MAC that, at the time of the capture, has been learned only on Leaf-2.

Verifying EVPN, Layer 2 Address Learning Daemon, and the Kernel-Forwarding Tables

Purpose Verify the Tenant B's EVPN table, and Core-1's Layer 2 address learning daemon table and kernel-forwarding table.

Action In Core-1, display the Tenant B's EVPN table.

```
lab@core-1> show route table VS_VLAN200.evpn.0
```

```
VS_VLAN200.evpn.0: 18 destinations, 31 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
1:192.0.2.0:0::0101010101010101::FFFF:FFFF/304
    *[BGP/170] 23:27:33, localpref 100, from 192.0.2.0
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
1:192.0.2.0:0::0202020202020202::FFFF:FFFF/304
    *[BGP/170] 23:27:33, localpref 100, from 192.0.2.0
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0

    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
1:192.0.2.0:1::0101010101010101::0/304
    *[BGP/170] 23:27:33, localpref 100, from 192.0.2.0
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
1:192.0.2.0:1::0202020202020202::0/304
    *[BGP/170] 23:27:33, localpref 100, from 192.0.2.0
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
        AS path: 65402 I, validation-state: unverified
        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
```

```

1:10.255.255.5:0:0101010101010101:FFFF:FFFF/304
    * [BGP/170] 23:27:33, localpref 100, from 192.0.2.0
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
1:10.255.255.5:0:0202020202020202:FFFF:FFFF/304
    * [BGP/170] 23:27:33, localpref 100, from 192.0.2.0
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
1:10.255.255.5:1:0101010101010101:0/304
    * [BGP/170] 23:27:33, localpref 100, from 192.0.2.0
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
1:10.255.255.5:1:0202020202020202:0/304
    * [BGP/170] 23:27:33, localpref 100, from 192.0.2.0
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 23:27:33, localpref 100, from 10.255.255.5
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
[output omitted]
2:192.0.2.0:1:1200:00:21:59:c8:24:64/304
    * [BGP/170] 4d 10:47:09, localpref 100, from 192.0.2.0
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
      to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 1d 00:44:04, localpref 100, from 10.255.255.5
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
      to 10.0.0.3 via ge-1/0/1.0
2:10.255.255.5:1:1200:00:21:59:c8:24:41/304
    * [BGP/170] 00:14:09, localpref 100, from 192.0.2.0
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
      to 10.0.0.3 via ge-1/0/1.0
    [BGP/170] 00:14:09, localpref 100, from 10.255.255.5
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
      to 10.0.0.3 via ge-1/0/1.0
2:10.255.255.5:1:1200:00:21:59:c8:24:68/304
    * [BGP/170] 1d 00:44:04, localpref 100, from 192.0.2.0
      AS path: 65402 I, validation-state: unverified
      to 10.0.0.1 via ge-1/0/0.0
      to 10.0.0.3 via ge-1/0/1.0
    AS path: 65402 I, validation-state: unverified

```

```

        to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
[BGP/170] 23:27:33, localpref 100, from 10.255.255.5
    AS path: 65402 I, validation-state: unverified
    to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
1:10.255.255.5:1::0202020202020202::0/304
    *[BGP/170] 23:27:33, localpref 100, from 192.0.2.0
    AS path: 65402 I, validation-state: unverified
    to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
[BGP/170] 23:27:33, localpref 100, from 10.255.255.5
    AS path: 65402 I, validation-state: unverified
    to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
2:192.0.2.0:1::1200::00:21:59:c8:24:64/304
    *[BGP/170] 4d 10:47:09, localpref 100, from 192.0.2.0
    AS path: 65402 I, validation-state: unverified
    > to 10.0.0.1 via ge-1/0/0.0
    to 10.0.0.3 via ge-1/0/1.0
[BGP/170] 1d 00:44:04, localpref 100, from 10.255.255.5
    AS path: 65402 I, validation-state: unverified
    > to 10.0.0.1 via ge-1/0/0.0
    to 10.0.0.3 via ge-1/0/1.0
2:10.255.255.5:1::1200::00:21:59:c8:24:41/304
    *[BGP/170] 00:14:09, localpref 100, from 192.0.2.0
    AS path: 65402 I, validation-state: unverified
    > to 10.0.0.1 via ge-1/0/0.0
    to 10.0.0.3 via ge-1/0/1.0
[BGP/170] 00:14:09, localpref 100, from 10.255.255.5
    AS path: 65402 I, validation-state: unverified
    > to 10.0.0.1 via ge-1/0/0.0
    to 10.0.0.3 via ge-1/0/1.0
2:10.255.255.5:1::1200::00:21:59:c8:24:68/304
    *[BGP/170] 1d 00:44:04, localpref 100, from 192.0.2.0
    AS path: 65402 I, validation-state: unverified
    to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
[BGP/170] 4d 10:47:09, localpref 100, from 10.255.255.5
    AS path: 65402 I, validation-state: unverified
    to 10.0.0.1 via ge-1/0/0.0
    > to 10.0.0.3 via ge-1/0/1.0
[output omitted]

```

Display Core-1's Layer 2 address learning daemon table.

```
lab@core-1> show bridge mac-table instance VS_VLAN200
```

```
MAC flags      (S -static MAC, D -dynamic MAC, L -locally learned, C -Control
MAC
    0 -OVSDB MAC, SE -Statistics enabled, NM -Non configured MAC, R -Remote PE
MAC)
```

```
Routing instance : VS_VLAN200
```

```
Bridging domain : bd1200, VLAN : 200
```

MAC address	MAC flags	Logical interface	Active source
00:21:59:c8:24:41	DR	esi.703	
00:02:02:02:02:02:02:02:02			

```

00:21:59:c8:24:64   DR      esi.703
00:02:02:02:02:02:02:02:02
00:21:59:c8:24:68   DR      esi.703
00:02:02:02:02:02:02:02:02

```



NOTE: On EX9200 switches, the `show ethernet-switching table-instance instance-name` command corresponds to the `show bridge mac-table instance instance-name` command on MX Series routers

Display Core-1's kernel forwarding table.

```

lab@core-1> show route forwarding-table vpn VS_VLAN200
Routing table: VS_VLAN200.evpn-vxlan
VPLS:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
default          perm   0                dscd   536     1
vtep.32774       intf   0                comp   702     6
vtep.32778       intf   0                comp   720     6

Routing table: VS_VLAN200.evpn-vxlan
Bridging domain: bd1200.evpn-vxlan
VPLS:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
00:21:59:c8:24:41/48 user    0                indr  1048579  4
                                comp   703     2
00:21:59:c8:24:64/48 user    0                indr  1048579  4
                                comp   703     2
00:21:59:c8:24:68/48 user    0                indr  1048579  4
                                comp   703     2
0x30002/51       user    0                comp   714     2

```

Meaning For the Tenant_B CE device, four different routes are listed for ESI 00:02:02:02:02:02:02:02:02.

- 1:192.0.2.0:0::0202020202020202::FFFF:FFFF/304

This per-Ethernet Segment A-D Type-1 EVPN route originated from Leaf-1. The route distinguisher is taken from global-level **routing-options**. Core-1 receives this Type-1 route, originated from Leaf-1, from both Leaf-1 and Leaf-2.

- 1:192.0.2.0:1::0202020202020202::0/304

This is the per-EVI A-D Type-1 EVPN route. The route distinguisher is taken from the routing instance, or in the case of QFX5100, **switch-options**. Core-1 receives this Type-1 route, originated from Leaf-1, from both Leaf-1 and Leaf-2.

- 1:10.255.255.5:0::0202020202020202::FFFF:FFFF/304

This is the per-Ethernet Segment A-D Type-1 EVPN route originated from Leaf-2. The route distinguisher is taken from global-level **routing-options**. Core-1 receives this Type-1 route, originated from Leaf-2, from both Leaf-2 and Leaf-1.

- 1:10.255.255.5:1::0202020202020202::0/304

This is the per-EVI A-D Type-1 EVPN route. The route distinguisher is taken from the routing instance, or in the case of QFX5100, **switch-options**. Core-1 receives this Type-1 route, originated from Leaf-2, from both Leaf-2 and Leaf-1.

Type-2 routes for the two physical and one virtual MAC belonging to the Tenant_B CE device originated as expected.

From the output we cannot yet determine what VTEPs are being used to forward to ES1 00:02:02:02:02:02:02:02:02:02. To determine the VTEPS, list the VXLAN tunnel endpoint ESIs.

```
lab@core-1> show 12-learning vxlan-tunnel-end-point esi
```

```

ESI                                RTT                                VLNBH  INH          ESI-IFL
LOC-IFL    #RVTEPs
00:01:01:01:01:01:01:01 VS_VLAN200                                704    1048580 esi.704
2
RVTEP-IP                                RVTEP-IFL    VENH    MASK-ID    FLAGS
10.255.255.5                                vtep.32778    720    1    2
192.0.2.0                                vtep.32774    702    0    2
[output omitted]

```



NOTE: On EX9200 switches, the show ethernet-switching command corresponds to the show l2-learning command on MX Series routers.

The output shows active load-balancing on the VTEP interfaces to both Leaf-1 and Leaf-2 for MACs on this ESI, which validates the all-active configuration on Leaf-1 and Leaf-2.

The reference to esi.703 might be confusing. This is an internal unicast next hop that comprises multiple, valid, individual ECMP next hops. You can correlate these individual next hops to the unicast by executing this shell level command:

```
lab@core-1> start shell command "nhinfo -di 703"
[output omitted]
NHs in list: 720, 702,
[output omitted]
```

From the output we can reference index 720 with vtep.32778 to Leaf-2 and index 702 with vtep.32774 to Leaf-1.

Verifying an EVPN Anycast Gateway

Purpose Verifying the EVPN anycast gateway is similar to the multihomed CE device scenario. For this example, verify from the leaf (QFX5100) perspective.

Action For this example, verify the anycast gateway on both Core-1 and Core-2 for VNI 1100 (Vlan 100 on both Leaf-1 and Leaf-2).

```
lab@leaf-1> show route receive-protocol bgp 10.255.255.0

inet.0: 13 destinations, 18 routes (13 active, 0 holddown, 0 hidden)

:vxlan.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)

bgp.evpn.0: 75 destinations, 123 routes (75 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED      Lclpref      AS path
  1:10.255.255.0:0::050000ff780000044c00::FFFF:FFFF/304
  *           10.255.255.0           65400 I
  1:10.255.255.0:0::050000ff78000004b000::FFFF:FFFF/304
  *           10.255.255.0           65400 I
  1:10.255.255.0:0::050000ff780000051400::FFFF:FFFF/304
  *           10.255.255.0           65400 I
  1:10.255.255.0:0::050000ff780000057800::FFFF:FFFF/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:300::1300::00:00:5e:00:01:01/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:300::1300::40:a6:77:9a:43:f0/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:100::1100::00:00:5e:00:01:01/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:100::1100::40:a6:77:9a:43:f0/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:400::1400::00:00:5e:00:01:01/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:400::1400::40:a6:77:9a:43:f0/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:200::1200::00:00:5e:00:01:01/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:200::1200::40:a6:77:9a:43:f0/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:300::1300::00:00:5e:00:01:01::10.10.10.1/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:300::1300::40:a6:77:9a:43:f0::10.10.10.2/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:100::1100::00:00:5e:00:01:01::100.0.0.1/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:100::1100::40:a6:77:9a:43:f0::172.16.0.2/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:400::1400::00:00:5e:00:01:01::10.10.10.1/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:400::1400::40:a6:77:9a:43:f0::10.10.10.2/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:200::1200::00:00:5e:00:01:01::200.0.0.1/304
  *           10.255.255.0           65400 I
  2:10.255.255.0:200::1200::40:a6:77:9a:43:f0::172.16.0.3/304
  *           10.255.255.0           65400 I
[output omitted]
```

Verify the same Type-1, same anycast Type-2, and different physical Type-2 (a.b.c.3) from Core-2.

```
lab@leaf-1> show route receive-protocol bgp 10.255.255.1

inet.0: 13 destinations, 18 routes (13 active, 0 holddown, 0 hidden)
```

```

:vxlan.inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)

bgp.evpn.0: 75 destinations, 123 routes (75 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED      Lclpref      AS path
1:10.255.255.1:0::050000ff780000044c00::FFFF:FFFF/304
*           10.255.255.1           65400 I
1:10.255.255.1:0::050000ff78000004b000::FFFF:FFFF/304
*           10.255.255.1           65400 I
1:10.255.255.1:0::050000ff780000051400::FFFF:FFFF/304
*           10.255.255.1           65400 I
1:10.255.255.1:0::050000ff780000057800::FFFF:FFFF/304
*           10.255.255.1           65400 I
2:10.255.255.1:300::1300::00:00:5e:00:01:01/304
*           10.255.255.1           65400 I
2:10.255.255.1:300::1300::40:a6:77:9a:47:f0/304
*           10.255.255.1           65400 I
2:10.255.255.1:100::1100::00:00:5e:00:01:01/304
*           10.255.255.1           65400 I
2:10.255.255.1:100::1100::40:a6:77:9a:47:f0/304
*           10.255.255.1           65400 I
2:10.255.255.1:400::1400::00:00:5e:00:01:01/304
*           10.255.255.1           65400 I
2:10.255.255.1:400::1400::40:a6:77:9a:47:f0/304
*           10.255.255.1           65400 I
2:10.255.255.1:200::1200::00:00:5e:00:01:01/304
*           10.255.255.1           65400 I
2:10.255.255.1:200::1200::40:a6:77:9a:47:f0/304
*           10.255.255.1           65400 I
2:10.255.255.1:300::1300::00:00:5e:00:01:01::10.10.10.1/304
*           10.255.255.1           65400 I
2:10.255.255.1:300::1300::40:a6:77:9a:47:f0::10.10.10.3/304
*           10.255.255.1           65400 I
2:10.255.255.1:100::1100::00:00:5e:00:01:01::100.0.0.1/304
*           10.255.255.1           65400 I
2:10.255.255.1:100::1100::40:a6:77:9a:47:f0::172.16.0.4/304
*           10.255.255.1           65400 I
2:10.255.255.1:400::1400::00:00:5e:00:01:01::10.10.10.1/304
*           10.255.255.1           65400 I
2:10.255.255.1:400::1400::40:a6:77:9a:47:f0::10.10.10.3/304
*           10.255.255.1           65400 I
2:10.255.255.1:200::1200::00:00:5e:00:01:01::200.0.0.1/304
*           10.255.255.1           65400 I
2:10.255.255.1:200::1200::40:a6:77:9a:47:f0::172.16.0.5/304
[output omitted]

```

Similarly, if we look at the default-switch.evpn.0 table on leaf-1, we see for the ESI for VN11110:

```

lab@leaf-1> show route table default-switch.evpn.0 evpn-esi-value
05:00:00:ff:78:00:00:04:4c:00

default-switch.evpn.0: 66 destinations, 114 routes (66 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

1:10.255.255.0:0::050000ff780000044c00::FFFF:FFFF/304
*[BGP/170] 00:23:37, localpref 100, from 10.255.255.0
AS path: 65400 I, validation-state: unverified
> to 10.0.0.8 via xe-0/0/2.0

```

```

        to 10.0.0.12 via xe-0/0/4.0
[BGP/170] 00:09:29, localpref 100, from 10.255.255.5
AS path: 65400 I, validation-state: unverified
> to 10.0.0.8 via xe-0/0/2.0
  to 10.0.0.12 via xe-0/0/4.0
1:10.255.255.1:0::050000ff78000004c00::FFFF:FFFF/304
*[BGP/170] 00:23:33, localpref 100, from 10.255.255.1
AS path: 65400 I, validation-state: unverified
> to 10.0.0.8 via xe-0/0/2.0
  to 10.0.0.12 via xe-0/0/4.0
[BGP/170] 00:09:29, localpref 100, from 10.255.255.5
AS path: 65400 I, validation-state: unverified
> to 10.0.0.8 via xe-0/0/2.0
  to 10.0.0.12 via xe-0/0/4.0

```

Meaning The output shows that Leaf-1 receives Type-1 advertisements for the autogenerated ESI's for the anycast gateway. The outputs also show the anycast and physical MACs (a.b.c.2) for each bridge domain.

Two unique advertisements are received for this ESI. One from Core-1, and the other from Core-2, with corresponding route distinguishers. Because Leaf-1 is iBGP peered with Leaf-2, the extra copy for each route from Leaf-2 displays with the original protocol-next hop preserved.

Leaf-1's Layer 2 address learning daemon table displays:

```
lab@leaf-1> show ethernet-switching table vlan-id 100
```

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static

SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, 0 - ovsdb MAC)

Ethernet switching table : 5 entries, 5 learned

Routing instance : default-switch

Vlan	MAC	MAC	Logical	Active
name	address	flags	interface	source
v100	00:00:5e:00:01:01	DR	esi.1727	
05:00:00:ff:78:00:00:04:4c:00				
v100	00:21:59:c8:24:63	DL	ae0.0	
v100	00:21:59:c8:24:69	D	vtep.32771	
10.255.255.5				
v100	40:a6:77:9a:43:f0	D	vtep.32769	
10.255.255.0				
v100	40:a6:77:9a:47:f0	D	vtep.32770	
10.255.255.1				

Leaf-1's kernel table displays:

```
lab@leaf-1> show route forwarding-table family ethernet-switching
[output omitted]
```

Routing table: default-switch.bridge


```
VPLS:
Destination      Type RtRef Next hop      Type Index  NhRef Netif
default          perm   0                dscd   1688    1
vtep.32769       intf   0                comp   1724   19
vtep.32770       intf   0                comp   1737   15
vtep.32771       intf   0                comp   1738    9
[output omitted]
```

```
outing table: default-switch.bridge
```

```
Bridging domain: v100.bridge
```

```
VPLS:
Destination      Type RtRef Next hop      Type Index  NhRef Netif
00:00:5e:00:01:01/48 user    0                comp   1724   19
[output omitted]
```

From the output, Leaf-1 has selected index 1724, correlating with vtep.32769, to forward all traffic for the anycast gateway on VNI 1100, and vtep.32769 is the VTEP to Core-1.

```
lab@leaf-1> show ethernet-switching vxlan-tunnel-end-point esi
[output omitted]
```

```
05:00:00:ff:78:00:00:04:4c:00 default-switch      1727 131074 esi.1727
2
RVTEP-IP      RVTEP-IFL  VENH  MASK-ID  FLAGS
10.255.255.1  vtep.32770 1737   1        2
10.255.255.0  vtep.32769 1724   0        2
```

Related Documentation

- [Understanding EVPN with VXLAN Data Plane Encapsulation on page 5](#)

CHAPTER 5

Configuring Integrated Bridging and Routing Using EVPNs

- [An EVPN with IRB Solution on EX9200 Switches Overview on page 99](#)
- [Configuring an EVPN with IRB Solution on EX9200 Switches on page 104](#)
- [Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 106](#)

An EVPN with IRB Solution on EX9200 Switches Overview

A data center service provider (DCSP) hosts the data center for its multiple customers on a common physical network. To each customer (also called a tenant), the service looks like a full-fledged data center that can expand to 4094 VLANs and all private subnets. For disaster recovery, high availability, and optimization of resource utilization, it is common for the DCSP to span the data center over multiple sites. To deploy data center services, a DCSP faces the following main challenges:

- Extending Layer 2 domains across more than one data center site. This requires optimal intra-subnet traffic forwarding.
- Supporting optimal inter-subnet traffic forwarding and optimal routing in the event of virtual machine (VM) motion.
- Supporting multiple tenants with independent VLAN and subnet space.

Ethernet VPN (EVPN) is targeted to handle all the preceding challenges, wherein:

- The basic EVPN functionality enables optimal intra-subnet traffic forwarding
- Implementing the integrated routing and bridging (IRB) solution in an EVPN deployment enables optimal inter-subnet traffic forwarding
- Configuring EVPN with virtual switch support enables multiple tenants with independent VLAN and subnet space

The following sections describe the integrated routing and bridging (IRB) solution for EVPNs:

- [Need for an EVPN IRB Solution on page 100](#)
- [Implementing the EVPN IRB Solution on page 100](#)

- [Benefits of Implementing the EVPN IRB Solution on page 102](#)
- [IPv6 Support for IRB Interfaces with EVPN Using Neighborhood Discovery Protocol \(NDP\) on page 104](#)

Need for an EVPN IRB Solution

EVPN is a technology used to provide Layer 2 extension and interconnection across an IP/MPLS core network to different physical sites belonging to a single Layer 2 domain. In a data center environment with EVPN, there is a need for both Layer 2 (intra-subnet traffic) and Layer 3 (inter-subnet traffic) forwarding and potentially interoperability with tenant Layer 3 VPNs.

With only a Layer 2 solution, there is no optimum forwarding of inter-subnet traffic, even when the traffic is local, for instance, when both the subnets are on the same server.

With only a Layer 3 solution, the following issues for intra-subnet traffic can arise:

- MAC address aliasing issue where duplicate MAC addresses are not detected.
- TTL issue for applications that use TTL 1 to confine traffic within a subnet.
- IPv6 link-local addressing and duplicate address detection that relies on Layer 2 connectivity.
- Layer 3 forwarding does not support the forwarding semantics of a subnet broadcast.
- Support of non-IP applications that require Layer 2 forwarding.

Because of the above mentioned shortcomings of a pure Layer 2 and Layer 3 solution, there is a need for a solution incorporating optimal forwarding of both Layer 2 and Layer 3 traffic in the data center environment when faced with operational considerations such as Layer 3 VPN interoperability and virtual machine (VM) mobility.

An EVPN-based integrated routing and bridging (IRB) solution provides optimum unicast and multicast forwarding for both intra-subnets and inter-subnets within and across data centers.

The EVPN IRB feature is useful for service providers operating in an IP/MPLS network that provides both Layer 2 VPN or VPLS services and Layer 3 VPN services who want to extend their service to provide cloud computation and storage services to their existing customers.

Implementing the EVPN IRB Solution

An EVPN IRB solution provides the following:

- Optimal forwarding for intra-subnet (Layer 2) traffic.
- Optimal forwarding for inter-subnet (Layer 3) traffic.
- Support for ingress replication for multicast traffic.
- Support for network-based as well as host-based overlay models.
- Support for consistent policy-based forwarding for both Layer 2 and Layer 3 traffic.

Junos OS supports several models of EVPN configuration to satisfy the individual needs of EVPN and data center cloud services customers. To provide flexibility and scalability, multiple VLANs can be defined within a particular EVPN instance. Likewise, one or more EVPN instances can be associated with a single Layer 3 VPN virtual routing and forwarding (VRF). In general, each data center tenant is assigned a unique Layer 3 VPN VRF, while a tenant could comprise one or more EVPN instances and one or more VLANs per EVPN instance. To support this model, each configured VLAN (including the default VLAN for an EVPN instance) requires an IRB interface to perform the Layer 2 and Layer 3 functions. Each VLAN or IRB interface maps to a unique IP subnet in the VRF.

There are two major functions that are supported for IRB in EVPN.

- Host MAC-IP synchronization

This includes:

- Advertising the IP address along with the MAC advertisement route in EVPN. This is done by using the IP field in the EVPN MAC advertisement route.
- The receiving PE router installs MAC into the EVPN instance (EVI) table and installs IP into the associated VRF.

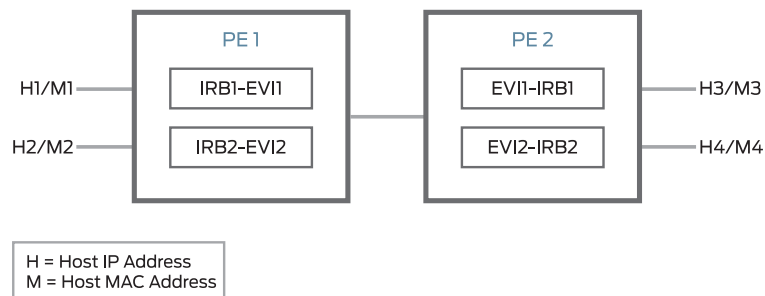
- Gateway MAC-IP synchronization

This includes:

- Advertising all local IRB MAC and IP addresses in an EVPN. This is achieved by including the default gateway extended community in the EVPN MAC advertisement route.
- The receiving PE creates a forwarding state to route packets destined for the gateway MAC, and a proxy ARP is done for the gateway IP with the MAC advertised in the route.

Figure 4 on page 101 illustrates the inter-subnet traffic forwarding between two provider edge (PE) devices—PE1 and PE2. The IRB1 and IRB2 interfaces on each PE device belong to a different subnet, but they share a common VRF.

Figure 4: Inter-Subnet Traffic Forwarding



The inter-subnet traffic forwarding is performed as follows:

1. PE2 advertises the H3-M3 and H4-M4 binding to PE1. Similarly, PE1 advertises the H1-M1 and H2-M2 binding to PE2.
2. PE1 and PE2 install the MAC address in the corresponding EVI MAC table, whereas the IP routes are installed in the shared VRF.
3. The advertising PE device is set as the next hop for the IP routes.
4. If H1 sends packets to H4, the packets are sent to IRB1 on PE1.
5. IP lookup for H4 happens in the shared VRF on PE1. Because the next hop for the H4 IP is PE2 (the advertising PE), an IP unicast packet is sent to PE2.
6. PE1 rewrites the MAC header based on the information in the VRF route, and PE2 performs a MAC lookup to forward the packet to H4.

Benefits of Implementing the EVPN IRB Solution

The main goal of the EVPN IRB solution is to provide optimal Layer 2 and Layer 3 forwarding. The solution is required to efficiently handle inter-subnet forwarding as well as virtual machine (VM) mobility. VM mobility refers to the ability of a VM to migrate from one server to another within the same or a different data center while retaining its existing MAC and IP address. Providing optimal forwarding for inter-subnet traffic and effective VM mobility involves solving two problems – the default gateway problem and the triangular routing problem.

- [Gateway MAC and IP Synchronization on page 102](#)
- [Layer 3 VPN Interworking on page 103](#)

Gateway MAC and IP Synchronization

In an EVPN IRB deployment, the IP default gateway for a VM is the IP address configured on the IRB interface of the provider edge (PE) router corresponding to the VLAN of which the VM is a member. The default gateway problem arises because a VM does not flush its ARP table when relocating from one server to another and continues sending packets with the destination MAC address set to that of the original gateway. If the old and new servers are not part of the same Layer 2 domain (the new Layer 2 domain could be within the current data center or a new data center), the gateway previously identified is no longer the optimal or local gateway. The new gateway needs to identify packets containing the MAC addresses of other gateways on remote PE routers and forward the traffic as if the packets were destined to the local gateway itself. At minimum, this functionality requires each PE router to advertise its gateway or IRB MAC and IP addresses to all other PE routers in the network. The gateway address exchange can be accomplished using the standard MAC route advertisement message (including the IP address parameter) and tagging that route with the default gateway extended community so that the remote

PE routers can distinguish the gateway MAC advertisement routes from normal MAC advertisement routes.

Layer 3 VPN Interworking

The inter-data center aspect of the EVPN IRB solution involves routing between VMs that are present in different data centers or routing between a host site completely outside of the data center environment and a VM within a data center. This solution relies on the ability of EVPN MAC route advertisements to carry both MAC address and IP address information. The local MAC learning functionality of the PE router is extended to also capture IP address information associated with MAC addresses learned locally. That IP-MAC address mapping information is then distributed to each PE router through normal EVPN procedures. When a PE router receives such MAC and IP information, it installs the MAC route in the EVPN instance as well as a host route for the associated IP address in the Layer 3 VPN VRF corresponding to that EVPN instance. When a VM moves from one data center to another, normal EVPN procedures result in the MAC and IP address being advertised from the new PE router which the VM resides behind. The host route installed in the VRF associated with an EVPN solicits Layer 3 traffic destined to that VM to the new PE router and avoids triangular routing between the source, the former PE router the VM resided behind, and the new PE router.

BGP scalability is a potential concern with the inter-data center triangular routing avoidance solution because of the potential for injection of many host routes into Layer 3 VPN. With the method previously described, in the worst case there is an IP host route for each MAC address learned through the local EVPN MAC learning procedures or through a MAC advertisement message received from a remote PE router. BGP route target filtering can be used to limit distribution of such routes.

The following functional elements are required to implement the inter-data center triangular routing avoidance using Layer 3 inter-subnet forwarding procedures:

1. The source host sends an IP packet using its own source MAC and IP address with the destination MAC of the IRB interface of the local PE router and the IP address of the destination host.
2. When the IRB interface receives the frame with its MAC as the destination, it performs a Layer 3 lookup in the VRF associated with the EVPN instance to determine where to route the packet.
3. In the VRF, the PE router finds the Layer 3 route derived from a MAC plus an IP EVPN route received from the remote PE router earlier. The destination MAC address is then changed to the destination MAC address corresponding to the destination IP.
4. The packet is then forwarded to the remote PE router serving the destination host using MPLS, using the label corresponding to the EVPN instance of which the destination host is a member.

5. The egress PE router receiving the packet performs a Layer 2 lookup for the destination host's MAC and sends the packet to the destination host on the attached subnet via the egress PE router's IRB interface.
6. Because the ingress PE router is performing Layer 3 routing, the IP TTL is decremented.

IPv6 Support for IRB Interfaces with EVPN Using Neighborhood Discovery Protocol (NDP)

Starting in Junos OS Release 17.3R1, IPv6 addresses are supported on IRB interfaces with EVPN using NDP. The following capabilities are introduced for IPv6 support with EVPN:

- IPv6 addresses on IRB interfaces in master routing instances
- Learning IPv6 neighbors from solicited neighbor advertisement (NA) messages
- Neighbor solicitation (NS) and neighbor advertisement (NA) packets on the IRB interfaces are disabled from network core
- Virtual gateway addresses are used as Layer 3 addresses
- Host MAC-IP synchronization for IPv6

You can configure the IPv6 addresses in the IRB interface at the **[edit interfaces irb]** hierarchy level.

Release History Table

Release	Description
17.3R1	Starting in Junos OS Release 17.3R1, IPv6 addresses are supported on IRB interfaces with EVPN

Related Documentation

- [Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 106](#)
- [EVPN Overview for Switches on page 3](#)

Configuring an EVPN with IRB Solution on EX9200 Switches

You can configure an Ethernet VPN (EVPN) with IRB solution to enable Layer 2 switching and Layer 3 routing operations within a single node, thus avoiding extra hops for inter-subnet traffic. The EVPN IRB solution eliminates the default gateway problem using the gateway MAC and IP synchronization, and avoids the triangular routing problem with Layer 3 interworking by creating IP host routes for virtual machines (VMs) in the tenant virtual routing and forwarding (VRF) routing instances.

Before you begin:

1. Configure the switch interfaces.
2. Configure the router ID and autonomous system number for the device.
3. Enable the chained composite next hop for EVPN.
4. Configure OSPF or any other IGP protocol.

5. Configure a BGP internal group.
6. Include the EVPN signaling network layer reachability information (NLRI) to the internal BGP group.
7. Configure LDP.
8. Configure MPLS.

To configure the PE device:

1. Configure the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance instance-type evpn
```

2. Set the VLAN identifier for the bridging domain in the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance vlan-id VLAN-ID
```

3. Configure the interface name for the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance interface CE-facing-interface
```

4. Configure the IRB interface as the routing interface for the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance l3-interface irb.0
```

5. Configure the route distinguisher for the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance route-distinguisher route-distinguisher-value
```

6. Configure the VPN routing and forwarding (VRF) target community for the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance vrf-target vrf-target-value
```

7. Assign the interface name that connects the PE device site to the VPN.

```
[edit routing-instances]
user@PE1# set evpn-instance protocols evpn interface CE-facing-interface
```

8. Configure the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance instance-type vrf
```

9. Configure the IRB interface as the routing interface for the VRF routing instance.

```
[edit routing-instances]
```

```
user@PE1# set vrf-instance interface irb.0
```

10. Configure the route distinguisher for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance route-distinguisher route-distinguisher-value
```

11. Configure the VRF label for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance vrf-table-label
```

12. Verify and commit the configuration.

For example:

```
[edit routing-instances]
user@PE1# set evpna instance-type evpn
user@PE1# set evpna vlan-id 10
user@PE1# set evpna interface ge-1/1/8.0
user@PE1# set evpna l3-interface irb.0
user@PE1# set evpna route-distinguisher 100.255.0.1:100
user@PE1# set evpna vrf-target target:100:100
user@PE1# set evpna protocols evpn interface ge-1/1/8.0
user@PE1# set vrf instance-type vrf
user@PE1# set vrf interface irb.0
user@PE1# set vrf route-distinguisher 100.255.0.1:300
user@PE1# set vrf vrf-target target:100:300
user@PE1# set vrf vrf-table-label

[edit]
user@PE1# commit
commit complete
```

Related Documentation • [Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 106](#)

Example: Configuring an EVPN with IRB Solution on EX9200 Switches

This example shows how to configure an integrated routing and bridging (IRB) solution in an Ethernet VPN (EVPN) deployment.

- [Requirements on page 106](#)
- [Overview on page 107](#)
- [Configuration on page 107](#)
- [Verification on page 113](#)

Requirements

This example uses the following hardware and software components:

- Two EX9200 switches configured as PE routers

- Junos OS Release 14.2 or later running on all the PE routers

Before you begin:

1. Configure the switch interfaces.
2. Configure OSPF or any other IGP protocol.
3. Configure BGP.
4. Configure LDP.
5. Configure MPLS.

Overview

In an EVPN solution, multiple VLANs can be defined within a particular EVPN instance, and one or more EVPN instances can be associated with a single Layer 3 VPN VRF. In general, each data center tenant is assigned a unique Layer 3 VPN virtual route forwarding (VRF), although the tenant can comprise one or more EVPN instances or VLANs per EVPN instance.

To support this flexibility and scalability factor, the EVPN solution provides support for the IRB interfaces on EX9200 switches to facilitate optimal Layer 2 and Layer 3 forwarding along with virtual machine mobility. The IRB interfaces are configured on each configured VLAN including the default VLAN for an EVPN instance.

IRB is the ability to do Layer 2 switching and Layer 3 routing within a single node, thus avoiding extra hops for inter-subnet traffic. The EVPN IRB solution eliminates the default gateway problem using the gateway MAC and IP synchronization, and avoids the triangular routing problem with Layer 3 interworking by creating IP host routes for virtual machines (VMs) in the tenant VRFs.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
CE1    set interfaces ge-1/1/7 vlan-tagging
       set interfaces ge-1/1/7 unit 0 vlan-id 10
       set interfaces ge-1/1/7 unit 0 family inet address 10.0.0.1/24
       set routing-options static route 198.51.100.0/24 next-hop 10.0.0.251

PE1    set interfaces ge-1/0/8 unit 0 family inet address 192.0.2.1/24
       set interfaces ge-1/0/8 unit 0 family mpls
       set interfaces ge-1/1/8 flexible-vlan-tagging
       set interfaces ge-1/1/8 encapsulation flexible-ethernet-services
       set interfaces ge-1/1/8 unit 0 encapsulation vlan-bridge
       set interfaces ge-1/1/8 unit 0 vlan-id 10
       set interfaces irb unit 0 family inet address 10.0.0.251/24
       set interfaces lo0 unit 0 family inet address 203.0.113.1/32
       set routing-options router-id 203.0.113.1
```

```
set routing-options autonomous-system 100
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 203.0.113.1
set protocols bgp group ibgp family evpn signaling
set protocols bgp group ibgp neighbor 203.0.113.2
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set routing-instances evpna instance-type evpn
set routing-instances evpna vlan-id 10
set routing-instances evpna interface ge-1/1/8.0
set routing-instances evpna l3-interface irb.0
set routing-instances evpna route-distinguisher 203.0.113.1:100
set routing-instances evpna vrf-target target:100:100
set routing-instances evpna protocols evpn interface ge-1/1/8.0
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
set routing-instances vrf route-distinguisher 203.0.113.1:300
set routing-instances vrf vrf-target target:100:300
set routing-instances vrf vrf-table-label
```

```
PE2 set interfaces ge-2/0/8 unit 0 family inet address 192.0.2.2/24
set interfaces ge-2/0/8 unit 0 family mpls
set interfaces ge-2/1/8 flexible-vlan-tagging
set interfaces ge-2/1/8 encapsulation flexible-ethernet-services
set interfaces ge-2/1/8 unit 0 encapsulation vlan-bridge
set interfaces ge-2/1/8 unit 0 vlan-id 20
set interfaces irb unit 0 family inet address 198.51.100.251/24
set interfaces lo0 unit 0 family inet address 203.0.113.2/32
set routing-options router-id 203.0.113.2
set routing-options autonomous-system 100
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 203.0.113.2
set protocols bgp group ibgp family evpn signaling
set protocols bgp group ibgp neighbor 203.0.113.1
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set routing-instances evpna instance-type evpn
set routing-instances evpna vlan-id 20
set routing-instances evpna interface ge-2/1/8.0
set routing-instances evpna l3-interface irb.0
set routing-instances evpna route-distinguisher 203.0.113.2:100
set routing-instances evpna vrf-target target:200:100
set routing-instances evpna protocols evpn interface ge-2/1/8.0
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
```

```

set routing-instances vrf route-distinguisher 203.0.113.2:300
set routing-instances vrf vrf-target target:200:300
set routing-instances vrf vrf-table-label

```

```

CE2  set interfaces ge-2/1/7 unit 0 vlan-id 20
      set interfaces ge-2/1/7 unit 0 family inet address 198.51.100.2/24
      set routing-options static route 10.0.0.0/24 next-hop 198.51.100.251

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure Router PE1:



NOTE: Repeat this procedure for Router PE2, after modifying the appropriate interface names, addresses, and other parameters.

1. Configure Router PE1 interfaces.

```

[edit interfaces]
user@PE1# set ge-1/0/8 unit 0 family inet address 192.0.2.1/24
user@PE1# set ge-1/0/8 unit 0 family mpls

user@PE1# set ge-1/1/8 flexible-vlan-tagging
user@PE1# set ge-1/1/8 encapsulation flexible-ethernet-services
user@PE1# set ge-1/1/8 unit 0 encapsulation vlan-bridge
user@PE1# set ge-1/1/8 unit 0 vlan-id 10

user@PE1# set irb unit 0 family inet address 10.0.0.251/24

user@PE1# set lo0 unit 0 family inet address 203.0.113.1/32

```

2. Set the router ID and autonomous system number for Router PE1.

```

[edit routing-options]
user@PE1# set router-id 203.0.113.1
user@PE1# set autonomous-system 100

```

3. Configure the chained composite next hop for EVPN.

```

[edit routing-options]
user@PE1# set forwarding-table chained-composite-next-hop ingress evpn

```

4. Enable LDP on all interfaces of Router PE1, excluding the management interface.

```

[edit protocols]
user@PE1# set ldp interface all
user@PE1# set ldp interface fxp0.0 disable

```

5. Enable MPLS on all the interfaces of Router PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set mpls interface all
user@PE1# set mpls mpls interface fxp0.0 disable
```

6. Configure the BGP group for Router PE1.

```
[edit protocols]
user@PE1# set bgp group ibgp type internal
```

7. Assign local and neighbor addresses to the ibgp BGP group for Router PE1 to peer with Router PE2.

```
[edit protocols]
user@PE1# set bgp group ibgp local-address 203.0.113.1
user@PE1# set bgp group ibgp neighbor 203.0.113.2
```

8. Include the EVPN signaling Network Layer Reachability Information (NLRI) to the ibgp BGP group.

```
[edit protocols]
user@PE1# set bgp group ibgp family evpn signaling
```

9. Configure OSPF on all the interfaces of Router PE1, excluding the management interface.

```
[edit protocols]
user@PE1# set ospf area 0.0.0.0 interface all
user@PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
```

10. Configure the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn instance-type evpn
```

11. Set the VLAN identifier for the bridging domain in the evpn routing instance.

```
[edit routing-instances]
user@PE1# set evpn vlan-id 10
```

12. Configure the interface name for the evpn routing instance.

```
[edit routing-instances]
user@PE1# set evpn interface ge-1/1/8.0
```

13. Configure the IRB interface as the routing interface for the evpn routing instance.

```
[edit routing-instances]
user@PE1# set evpn l3-interface irb.0
```

14. Configure the route distinguisher for the evpn routing instance.

```
[edit routing-instances]
user@PE1# set evpna route-distinguisher 203.0.113.1:100
```

15. Configure the VPN routing and forwarding (VRF) target community for the evpna routing instance.

```
[edit routing-instances]
user@PE1# set evpna vrf-target target:100:100
```

16. Assign the interface name that connects the PE1 site to the VPN.

```
[edit routing-instances]
user@PE1# set evpna protocols evpn interface ge-1/1/8.0
```

17. Configure the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf instance-type vrf
```

18. Configure the IRB interface as the routing interface for the vrf routing instance.

```
[edit routing-instances]
user@PE1# set vrf interface irb.0
```

19. Configure the route distinguisher for the vrf routing instance.

```
[edit routing-instances]
user@PE1# set vrf route-distinguisher 203.0.113.1:300
```

20. Configure the VRF label for the vrf routing instance.

```
[edit routing-instances]
user@PE1# set vrf vrf-table-label
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, **show protocols**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-1/0/8 {
  unit 0 {
    family inet {
      address 192.0.2.1/24;
    }
    family mpls;
  }
}
ge-1/1/8 {
  flexible-vlan-tagging;
```

```
encapsulation flexible-ethernet-services;
unit 0 {
    encapsulation vlan-bridge;
    vlan-id 10;
}
}
irb {
    unit 0 {
        family inet {
            address 10.0.0.251/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 203.0.113.1/32 {
            }
        }
    }
}
```

```
user@PE1# show routing-options
router-id 203.0.113.1;
autonomous-system 100;
forwarding-table {
    chained-composite-next-hop {
        ingress {
            evpn;
        }
    }
}
```

```
user@PE1# show protocols
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    group ibgp {
        type internal;
        local-address 203.0.113.1;
        family evpn {
            signaling;
        }
        neighbor 203.0.113.2;
    }
}
ospf {
```



```

    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}

user@PE1# show routing-instances
evpna {
    instance-type evpn;
    vlan-id 10;
    interface ge-1/1/8.0;
    l3-interface irb.0;
    route-distinguisher 203.0.113.1:100;
    vrf-target target:100:100;
    protocols {
        evpn {
            interface ge-1/1/8.0;
        }
    }
}
vrf {
    instance-type vrf;
    interface irb.0;
    route-distinguisher 203.0.113.1:300;
    vrf-target target:100:300;
    vrf-table-label;
}

```

Verification

Confirm that the configuration is working properly.

- [Verifying Local IRB MACs on page 113](#)
- [Verifying Remote IRB MACs on page 114](#)
- [Verifying Local IRB IPs on page 115](#)
- [Verifying Remote IRB IPs on page 116](#)
- [Verifying CE-CE Inter-Subnet Forwarding on page 117](#)

Verifying Local IRB MACs

Purpose Verify that the local IRB MACs are learned from L2ALD.

Action On Router PE1, determine the MAC address of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```

user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10

```

From operational mode, run the **show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10"** command.

```
user@PE1> show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10"
2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10/384 (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group PE type Internal) Type 1 val 0x2736568 (adv_entry)
  Advertised metrics:
    Flags: Nexthop Change
    Nexthop: Self
    Localpref: 100
    AS path: [100] I
    Communities: target:100:100 evpn-default-gateway
Path 2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10 Vector len 4. Val: 0
  *EVPN Preference: 170
    Next hop type: Indirect
    Address: 0x26f8354
    Next-hop reference count: 6
    Protocol next hop: 10.255.0.1
    Indirect next hop: 0x0 - INH Session ID: 0x0
    State: <Active Int Ext>
    Age: 23:29:08
    Validation State: unverified
    Task: evpn-evpn
    Announcement bits (1): 1-BGP_RT_Background
    AS path: I
    Communities: evpn-default-gateway
    Route Label: 299776
```

Meaning The MAC-only route for the local IRB interface appears in the EVPN instance route table on Router PE1 and is learned from EVPN and tagged with the default gateway extended community.

Verifying Remote IRB MACs

Purpose Verify that the remote IRB MACs are learned from BGP.

Action On Router PE1, determine the MAC address of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```
user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10
```

On Router PE2, verify that the remote IRB MACs are learned.

From operational mode, run the **show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10"** command.

```
user@PE2> show route table evpn.0 extensive | find "a8:d0:e5:54:0d:10"
```

```

2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10/384 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 2.91.223.24:100
    Next hop type: Indirect
    Address: 0x26f8d6c
    Next-hop reference count: 10
    Source: 10.255.0.1
    Protocol next hop: 10.255.0.1
    Indirect next hop: 0x2 no-forward INH Session ID: 0x0
    State: <Secondary Active Int Ext>
    Local AS: 100 Peer AS: 100
    Age: 23:22:17 Metric2: 1
    Validation State: unverified
    Task: BGP_100.10.255.0.1
    Announcement bits (1): 0-evpna-evpn
    AS path: I
    Communities: target:100:100 evpn-default-gateway
    Import Accepted
    Route Label: 299776
    Localpref: 100
    Router ID: 10.255.0.1
    Primary Routing Table bgp.evpn.0
    Indirect next hops: 1
      Protocol next hop: 10.255.0.1 Metric: 1
      Indirect next hop: 0x2 no-forward INH Session ID: 0x0
      Indirect path forwarding next hops: 1
        Next hop type: Router
        Next hop: 1.0.0.1 via ge-1/0/8.0
        Session Id: 0x1
      10.255.0.1/32 Originating RIB: inet.3
        Metric: 1 Node path count: 1
        Forwarding nexthops: 1
        Nexthop: 1.0.0.1 via ge-1/0/8.0

```

Meaning The MAC-only route for the remote IRB interface appears in the EVPN instance route table on Router PE2 and is learned from BGP and tagged with the default gateway extended community.

Verifying Local IRB IPs

Purpose Verify that the local IRB IPs are learned locally by RPD.

Action On Router PE1, determine the MAC and IP addresses of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```

user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10

```

From operational mode, run the **show interfaces irb.0 terse | match inet** command.

```

user@PE1> show interfaces irb.0 terse | match inet
irb.0                up    up    inet    10.0.0.251/24

```

From operational mode, run the **show route table evpn.evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"** command.

```
user@PE2> show route table evpn.evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"
2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10::10.0.0.251/384 (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group PE type Internal) Type 1 val 0x27365a0 (adv_entry)
  Advertised metrics:
    Flags: Nexthop Change
    Nexthop: Self
    Localpref: 100
    AS path: [100] I
    Communities: target:100:100 evpn-default-gateway
Path 2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10::10.0.0.251 Vector len 4. Val:
0
    *EVPN   Preference: 170 <<<<<
           Next hop type: Indirect
           Address: 0x26f8354
           Next-hop reference count: 6
           Protocol next hop: 10.255.0.1
           Indirect next hop: 0x0 - INH Session ID: 0x0
           State: <Active Int Ext>
           Age: 23:48:46
           Validation State: unverified
           Task: evpn-evpn
           Announcement bits (1): 1-BGP_RT_Background
           AS path: I
           Communities: evpn-default-gateway
           Route Label: 299776
```

Meaning The MAC plus IP route for the local IRB interface appears in the EVPN instance route table on Router PE1 and is learned from EVPN and tagged with the default gateway extended community.

Verifying Remote IRB IPs

Purpose Verify that the remote IRB IPs are learned from BGP.

Action On Router PE1, determine the MAC and IP addresses of the local IRB interface.

From operational mode, run the **show interfaces irb extensive | match "Current address"** command.

```
user@PE1> show interfaces irb extensive | match "Current address"
Current address: a8:d0:e5:54:0d:10, Hardware address: a8:d0:e5:54:0d:10
```

From operational mode, run the **show interfaces irb.0 terse | match inet** command.

```
user@PE1> show interfaces irb.0 terse | match inet
irb.0                up    up    inet    10.0.0.251/24
```

On Router PE2, verify that the remote IRB IPs are learnt.

From operational mode, run the **show route table evpn.evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"** command.

```
user@PE2> show route table evpn.evpn.0 extensive | find "a8:d0:e5:54:0d:10::10.0.0.251"
2:10.255.0.1:100::0::100::a8:d0:e5:54:0d:10::10.0.0.251/384 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 2.91.223.216:100
    Next hop type: Indirect
    Address: 0x26f8d6c
    Next-hop reference count: 10
    Source: 10.255.0.1
    Protocol next hop: 10.255.0.1
    Indirect next hop: 0x2 no-forward INH Session ID: 0x0
    State: <Secondary Active Int Ext>
    Local AS: 100 Peer AS: 100
    Age: 23:56:36 Metric2: 1
    Validation State: unverified
    Task: BGP_100.10.255.0.1
    Announcement bits (1): 0-evpn-evpn
    AS path: I
    Communities: target:100:100 evpn-default-gateway
    Import Accepted
    Route Label: 299776
    Localpref: 100
    Router ID: 10.255.0.1
    Primary Routing Table bgp.evpn.0
    Indirect next hops: 1
      Protocol next hop: 10.255.0.1 Metric: 1
      Indirect next hop: 0x2 no-forward INH Session ID: 0x0
      Indirect path forwarding next hops: 1
        Next hop type: Router
        Next hop: 1.0.0.1 via ge-1/0/8.0
        Session Id: 0x1
        10.255.0.1/32 Originating RIB: inet.3
        Metric: 1 Node path count: 1
        Forwarding nexthops: 1
        Nexthop: 1.0.0.1 via ge-1/0/8.0
```

Meaning The MAC plus IP route for the remote IRB interface appears in the EVPN instance route table on Router PE2 and is tagged with the default gateway extended community.

Verifying CE-CE Inter-Subnet Forwarding

Purpose Verify inter-subnet forwarding between Routers CE1 and CE2.

Action From operational mode, run the **show route table inet.0** command.

```
user@CE1> show route table inet.0
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:15:09
                   > to 10.0.0.251 via ge-1/1/7.0
10.0.0.0/24        *[Direct/0] 1d 23:24:30
                   > via ge-1/1/7.0
10.0.0.1/32        *[Local/0] 1d 23:24:38
                   Local via ge-1/1/7.0
```

From operational mode, run the **ping** command.

```
user@CE1> ping 198.51.100.2 interval 0.1 count 10
PING 198.51.100.2 (20.0.0.2): 56 data bytes
64 bytes from 198.51.100.2: icmp_seq=0 ttl=63 time=0.919 ms
64 bytes from 198.51.100.2: icmp_seq=1 ttl=63 time=0.727 ms
64 bytes from 198.51.100.2: icmp_seq=2 ttl=63 time=0.671 ms
64 bytes from 198.51.100.2: icmp_seq=3 ttl=63 time=0.671 ms
64 bytes from 198.51.100.2: icmp_seq=4 ttl=63 time=0.666 ms
64 bytes from 198.51.100.2: icmp_seq=5 ttl=63 time=0.704 ms
64 bytes from 198.51.100.2: icmp_seq=6 ttl=63 time=0.763 ms
64 bytes from 198.51.100.2: icmp_seq=7 ttl=63 time=0.750 ms
64 bytes from 198.51.100.2: icmp_seq=8 ttl=63 time=12.967 ms
64 bytes from 198.51.100.2: icmp_seq=9 ttl=63 time=0.752 ms

--- 198.51.100.2 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.666/1.959/12.967/3.670 ms
```

Meaning Ping from Router CE1 to Router CE2 is successful.

Related Documentation

- [An EVPN with IRB Solution on EX9200 Switches Overview on page 99](#)

CHAPTER 6

Configuring EVPN Multihoming

- [EVPN Multihoming Overview on page 119](#)
- [Configuring EVPN Active-Standby Multihoming on page 144](#)
- [Example: Configuring EVPN Active-Active Multihoming on page 147](#)
- [Example: Configuring an ESI on a Logical Interface With EVPN Multihoming on page 195](#)

EVPN Multihoming Overview

- [Introduction to EVPN Multihoming on page 119](#)
- [Understanding EVPN Multihoming Concepts on page 121](#)
- [EVPN Multihoming Mode of Operation on page 123](#)
- [EVPN Multihoming Implementation on page 124](#)
- [Designated Forwarder Election on page 136](#)
- [ESIs on Physical, Aggregated Ethernet, and Logical Interfaces on page 141](#)
- [Convergence in an EVPN Network on page 141](#)

Introduction to EVPN Multihoming

An Ethernet VPN (EVPN) comprises of customer edge (CE) devices that are connected to provider edge (PE) devices, which form the edge of the MPLS infrastructure. A CE device can be a host, a router, or a switch. The PE devices provide Layer 2 virtual bridge connectivity between the CE devices. There can be multiple EVPNs in the provider network. Learning between the PE routers occurs in the control plane using BGP, unlike traditional bridging, where learning occurs in the data plane.



NOTE: In releases earlier than Junos OS Release 15.1, EVPN functionality support on MX Series routers was limited to routers using MPC and MIC interfaces only. Starting with Junos OS Release 15.1, MX Series routers using DPCs can be leveraged to provide EVPN support on the CE device-facing interface.

DPC support for EVPN is provided with the following considerations:

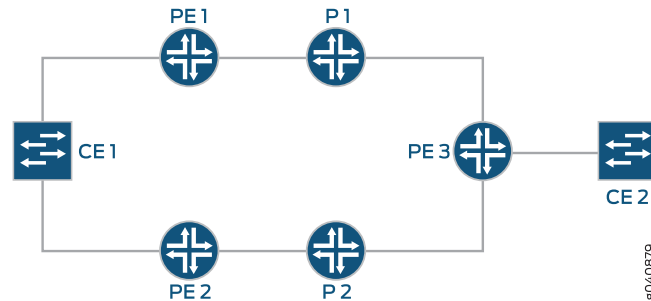
- DPCs provide support for EVPN in the active-standby mode of operation including support for the following:
 - EVPN instance (EVI)
 - Virtual switch
 - Integrated routing and bridging (IRB) interfaces
- DPCs intended for providing the EVPN active-standby support must be the CE device-facing line card. The PE device in the EVPN domain must be MPC interfaces or MIC interfaces.

The EVPN multihoming feature enables you to connect a customer site to two or more PE devices to provide redundant connectivity. A CE device can be multihomed to different PE devices or the same PE device. A redundant PE device can provide network service to the customer site as soon as a failure is detected. Thus, EVPN multihoming helps to maintain EVPN service and traffic forwarding to and from the multihomed site in the event of the following types of network failures:

- PE device to CE device link failure
- PE device failure
- MPLS-reachability failure between the local PE device and a remote PE device

Figure 5 on page 121 illustrates how a CE device can multihomed to two PE routers. Device CE 1 is multihomed to Routers PE 1 and PE 2. Device CE 2 has two potential paths to reach Device CE 1, and depending on the multihoming mode of redundancy, only one path or both the paths are active at any time. The multihoming mode of operation also determines which PE router or routers forward traffic to the CE device. The PE router forwarding traffic to the CE device (also called a designated forwarder) uses MPLS LSP or GRE tunnels to forward traffic. If a failure occurs over this path, a new designated forwarder is elected to forward the traffic to Device CE 1.

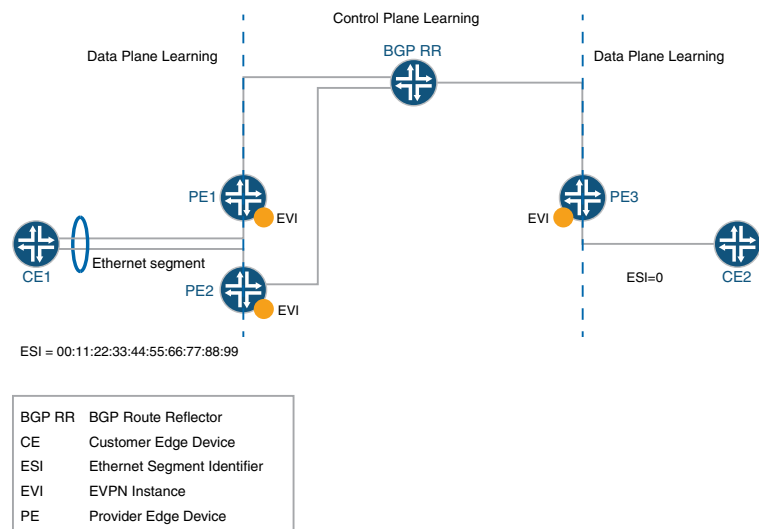
Figure 5: CE Device Multihomed to Two PE Routers



Understanding EVPN Multihoming Concepts

Figure 6 on page 121 shows a simple EVPN network topology to define EVPN multihoming concepts. .

Figure 6: Simple EVPN Topology



- **Ethernet segment**—When a CE device is multihomed to two or more PE routers, the set of Ethernet links constitutes an Ethernet segment. An Ethernet segment appears as a link aggregation group (LAG) to the CE device.

The links from Routers PE1 and PE2 to Device CE1 form an Ethernet segment.

In active-standby multihoming, the links that constitute an Ethernet segment form a bridge domain. In active-active multihoming, an Ethernet segment appears as a LAG to the CE device.

- **ESI**—An Ethernet segment must have a unique nonzero identifier, called the Ethernet segment identifier (ESI). The ESI is encoded as a 10-octet integer. When manually configuring an ESI value, the most significant octet, known as the type byte, must be

00. When a single-homed CE device is attached to an Ethernet segment, the entire ESI value is zero.

The Ethernet segment of the multihomed Device CE1 has an ESI value of 00:11:22:33:44:55:66:77:88:99 assigned. The single-homed Device CE2 has an ESI value of 0.

- **EVI**—An EVPN instance (EVI) is an EVPN routing and forwarding instance spanning all the PE routers participating in that VPN. An EVI is configured on the PE routers on a per-customer basis. Each EVI has a unique route distinguisher and one or more route targets.

An EVI is configured on Routers PE1, PE2, and PE3.



NOTE: The QFX10000 Series switches only support the default-switch routing instance.

- **Ethernet tag**—An Ethernet tag identifies a particular broadcast domain, such as a VLAN. An EVPN instance consists of one or more broadcast domains. Ethernet tags are assigned to the broadcast domains of a given EVPN instance by the provider of that EVPN. Each PE router in that EVPN instance performs a mapping between broadcast domain identifiers understood by each of its attached CE devices and the corresponding Ethernet tag.
- **Ethernet segment route**—The PE routers that are connected to a multihomed CE device use BGP Ethernet segment route messages to discover that each of the PE routers is connected to the same Ethernet segment. The PE routers advertise the Ethernet segment route, which consists of an ESI and ES-import extended community.

Routers PE1 and PE2 advertise an ES route with an ES-import extended community (along with other extended communities like the route target). The PE routers also construct a filter that is based on an ES-import extended community, which results in only these PE routers importing the ES route and identifying that they are connected to the same Ethernet segment.

- **Extended community**— An extended community is similar in most ways to a regular community. EVPNs use extended communities because the 4-octet regular community value does not provide enough expansion and flexibility. An extended community is an 8-octet value divided into two main sections.
- **BUM traffic**—This type of traffic is sent to multiple destinations, including broadcast traffic, unknown unicast traffic that is broadcast in the Ethernet segment, and multicast traffic.
- **DF**—When a CE device is multihomed to two or more PE routers, either one or all of the multihomed PE routers are used to reach the customer site depending on the multihoming mode of operation. The PE router that assumes the primary role for forwarding BUM traffic to the CE device is called the designated forwarder (DF).
- **BDF**—Each router in the set of other PE routers advertising the autodiscovery route per Ethernet segment for the same ESI, and serving as the backup path in case the DF

encounters a failure, is called a backup designated forwarder (BDF). A BDF is also called a non-DF router.

- **DF election**—On every Ethernet segment, the PE routers participate in a procedure called *designated forwarder* election to select the DF and the BDF PE routers.

EVPN Multihoming Mode of Operation

The different modes of operation for EVPN multihoming include:

- **Single**—When a PE router is connected to a single-homed customer site, this mode is in operation. The *single* mode is the default mode of operation, and does not require Ethernet segment values to be configured.
- **Active-standby**—When only a single PE router, among a group of PE routers attached to an Ethernet segment, is allowed to forward traffic to and from that Ethernet segment, the Ethernet segment is defined to be operating in the *active-standby* redundancy mode.

To configure the active-standby mode, include the ESI value and the **single-active** statement under the **[edit interfaces]** hierarchy level.

- **Active-active**—When all PE routers attached to an Ethernet segment are allowed to forward traffic to and from the Ethernet segment, the Ethernet segment is defined to be operating in the *active-active* redundancy mode.



NOTE: In Junos OS Release 14.2 and earlier, the EX9200 Series switch supports only the active-standby mode of operation for EVPN multihoming.



NOTE: Starting with Junos OS Release 14.1x53-D30, QFX5100 switches support the active-active mode of operation for EVPN multihoming. In this scenario, QFX5100 switches function as top-of-rack (ToR) switches in the data center for virtual networks. EVPN multihoming active-active functionality is used to provide access to the bare-metal servers connected to the top-of-rack switches.



NOTE: Starting with Junos OS Release 14.1R4, 14.2, 15.1F6, and 16.1R1, Junos OS supports the active-active mode for EVPN multihoming on MX Series routers.

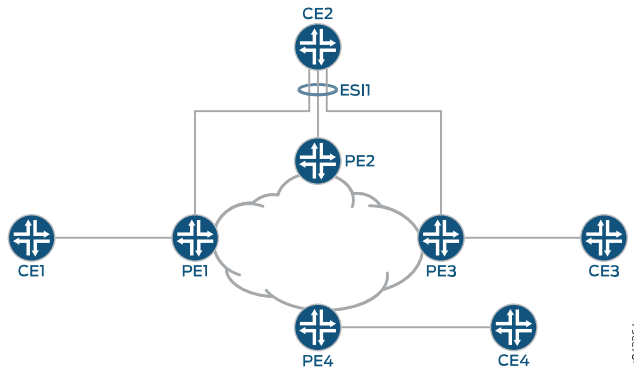
Starting with Junos OS Releases 16.1R4 and 16.2R2, all EX9200 switches support the active-active mode for EVPN multihoming.

To configure the active-active mode, include the ESI value and the **all-active** statement at the **[edit interfaces]** hierarchy level.

Figure 7 on page 124 shows a reference topology for EVPN active-active multihoming. The ESI1 Ethernet segment for Device CE2 is multihomed to Routers PE1, PE2, and PE3.

The Ethernet segment on the CE device can either be configured as a link aggregation group (LAG) or as an ECMP path. Devices CE1 and CE3 are the single-homed customer edge devices and have an ESI value of 0.

Figure 7: Active-Active EVPN Multihoming



EVPN Multihoming Implementation

The EVPN active-standby multihoming mode of operation provides redundancy for access link failures and PE node failure for the multihomed CE device, and is based on the EVPN *draft-ietf-l2vpn-evpn-03*.

The Junos OS implementation of the EVPN multihoming active-standby and active-active modes of operation includes the following:

- [New BGP NLRIs on page 124](#)
- [New Extended Communities on page 127](#)
- [New EVPN Route Types on page 129](#)
- [Update to the MAC Forwarding Table on page 130](#)
- [Traffic Flow on page 131](#)
- [Aliasing on page 133](#)
- [EVPN Active-Active Multihoming and Multichassis Link Aggregation on page 134](#)
- [EVPN Active-Active Multihoming and IRB on page 135](#)
- [Sample Configuration on page 135](#)

New BGP NLRIs

To support EVPN multihoming, the following new BGP network layer reachability information (NLRI) routes have been introduced:

- [Autodiscovery Route per Ethernet Segment on page 125](#)
- [Ethernet Segment Route on page 126](#)
- [Autodiscovery Route per EVPN Instance on page 127](#)

Autodiscovery Route per Ethernet Segment

- [Autodiscovery Route Features on page 125](#)
- [Autodiscovery Route Advertisement on page 125](#)
- [Autodiscovery Route Withdrawal on page 126](#)

Autodiscovery Route Features

The autodiscovery route NLRI features include:

- This is a Type 1 mandatory route, used for fast convergence and for advertising the split horizon label. It is also known as the mass withdraw route.
- Type 1 route distinguishers are used with the IP address (loopback) of the originating PE router as the route distinguisher value.
- This route carries the ESI in the NLRI (nonzero when it is a multihomed PE, zero otherwise).
- The split horizon label is per ESI only, and carries an explicit NULL (0).
- The bit in the active-standby flag field in the ESI label extended community is used for signaling the active-standby mode (bit set).
- The 3-byte label values in the NLRI and the Ethernet tag is zero.
- This route is advertised and imported by all multihomed and remote PE routers that share the same EVI on the advertising ESI.

Autodiscovery Route Advertisement

- Active-standby mode

In active-standby mode, the designated forwarder (DF) advertises the autodiscovery route per Ethernet segment with an ESI MPLS label extended community that has the standby bit set to 1. The autodiscovery route is advertised per ESI, and the ESI label is set to 0 when active-standby mode is in operation.

The autodiscovery route is imported by all the multihomed and remote PE routers that are part of the EVI. On receiving the autodiscovery route, the PE routers in the network topology learn that active-standby multihoming mode is in operation for the ESI advertised.

- Active-active mode

In active-active mode, each of the multihomed PE device advertises a mandatory autodiscovery route per Ethernet segment as in the active-standby state. However, in the active-active state, the autodiscovery route per Ethernet segment is modified such that the active-standby bit carried in the MPLS extended community is cleared to indicate that the active-active mode is in operation. The autodiscovery route per Ethernet segment in the active-active mode also includes the split horizon label.

In [Figure 7 on page 124](#), for the ESI1 Ethernet segment, Routers PE1, PE2, and PE3 advertise the autodiscovery route. Router PE4 receives this autodiscovery route.

Autodiscovery Route Withdrawal

The autodiscovery route per Ethernet segment withdrawal may result in mass withdrawal. The mass withdrawal feature is used when there is a link failure on the ESI, or when the ESI configuration changes.

When the link between a multihomed CE device and a multihomed PE device fails, the PE device withdraws the autodiscovery route per Ethernet segment. In such a case, the mass withdrawal feature is handled in the following ways by the other PE devices:

- Remote PE device

When a remote PE device receives the BGP update for mass withdrawal, the following is performed at the remote PE device:

1. The current next hop to reach the remote ESI or CE device is deleted.
2. A new next hop through the remaining multihomed PE devices is created to reach the remote ESI or CE device.
3. All the MAC routes behind the CE device are updated with the newly created next hop.

Starting with Junos OS Release 17.4R1, Junos OS supports Dynamic List Next Hops in an EVPN network. Now when the link between the CE device and a multihomed PE device fails, the next hop to the ESI or CE is updated, thus reducing the need for a mass withdrawal. For more information on enabling Dynamic List Next Hop, see *Configuring Dynamic List Next Hop*.

- Other multihomed PE device

As a result of the mass withdrawal, load balancing on the multihomed CE device happens because of the following:

- When the other multihomed PE devices receive the same set of MAC addresses on the link to the concerned ESI.

In this case, the local routes are preferred. If the remote routes learned from the DF PE device gets withdrawn, it does not affect routes pointing to the local ESI.

- When the other multihomed PE devices have not received the same set of MAC addresses on the link to the concerned ESI.

In this case, the PE devices install the MAC routes pointing to the concerned ESI, although the MACs are remotely learned from the DF PE device. When the DF PE device withdraws these routes, the withdrawn routes are flushed. Packets that are destined to the flushed MAC addresses are flooded on all the local segments.

Ethernet Segment Route

- [Ethernet Segment Route Features on page 127](#)
- [Ethernet Segment Route Advertisement on page 127](#)

Ethernet Segment Route Features

The Ethernet segment route NRLI features include:

- This is a Type 4 route. The purpose of this route is to enable the PE routers connected to the same Ethernet segment to automatically discover each other with minimal configuration on exchanging this route.
- This route is associated with an ES-import extended community with an ESI value condensed to 6 bytes, similar to a route target.
- This route is advertised and imported only by PE routers that are multihomed on the advertising Ethernet segment.

Ethernet Segment Route Advertisement

The Ethernet segment route is exchanged among all the PE routers within a data center with the ES-import extended community. The ES-import extended community is constructed based on the ESI PE routers that are multihomed, and the Ethernet segment route carries the ESI value related to the Ethernet segment on which the PE routers are multihomed.

The Ethernet segment routes are filtered based on the ES-import extended community, such that only the PE routers that are multihomed on the same Ethernet segment import this route. Each PE router that is connected to a particular Ethernet segment constructs an import filtering rule to import a route that carries the ES-import extended community.

Autodiscovery Route per EVPN Instance

In active-active mode, each of the multihomed PE devices advertise an autodiscovery route per EVPN instance (EVI) with a valid MPLS label. This route is advertised per ESI and is imported by the remote PE devices. The MPLS label included in the autodiscovery route per EVI is used later for aliasing.

New Extended Communities

An extended community is similar in most ways to a regular community. Some networking implementations, such as virtual private networks (VPNs), use extended communities because the 4-octet regular community value does not provide enough expansion and flexibility. An extended community is an 8-octet value divided into two main sections.

To support active-standby multihoming, the following extended communities have been introduced:

- [ESI-Import on page 127](#)
- [Split Horizon on page 128](#)

ESI-Import

This extended community is attached to the ES route, and is populated from the ESI-import value extracted from the configured ESI value under the interface. To solve the problem of a conflict with another regular route target, the type is set to 0x06, which has been allocated by IANA.

The ESI-import extended community route target populates the list of import route targets configured for the special instance from where the ES route using this community is advertised.

Therefore, incoming ESI routes with the same ESI-import value in the extended community are imported by the PE routers, if the PE router is configured with an Ethernet segment that has the same ESI value. Once the PE router receives a set of these ESI routes that have the same ESI-import extended community value, the DF and BDF election can be done locally.



NOTE: When the ESI-import extended community is not created implicitly, a policy must be configured to attach all the route targets to the autodiscovery route per Ethernet segment.

Split Horizon

With reference to [Figure 7 on page 124](#) for example, when a CE device that is multihomed to two or more PE devices on an Ethernet segment (ESI1) and operating in the active-active redundancy mode sends a BUM packet to one of the non-DF PE devices (say PE1), then Device PE1 forwards that packet to all or a subset of the other PE devices in that EVPN instance, including the DF PE device for that Ethernet segment. In this case the DF PE device that the CE device is multihomed to drops the packet without forwarding it back to the CE device. This filtering is referred to as split horizon.

- **Split horizon signaling**

The split horizon extended community is attached to the autodiscovery route per Ethernet segment. The value of the extended community is the split horizon or the Poisson label itself, which is 3 bytes, and is advertised as an opaque attribute.

- **Split horizon advertisement**

- In active-standby mode, the standby bit in the split horizon extended community is set to 1, and the ESI split horizon label is set to 0.
- In the active-active mode, the split horizon extended community is modified to clear the standby bit to 0 and includes a valid ESI label used for split horizon purposes.

- **Split horizon MPLS routes**

The DF PE device advertises an autodiscovery route per Ethernet segment with a split horizon label A, and an inclusive multicast route with label B for BUM traffic forwarding. On the DF, the BUM packet from the core can come with following labels:

- When the non-DF PE devices receive a BUM packet on their single-homed ESIs, the BUM packet is sent to the DF PE device with multicast label B.
- When the non-DF PE devices receive a BUM packet on ESI1, the BUM packet is sent to the DF PE device with two MPLS labels — the multicast label B as the outer label, and the split horizon label A as the inner label.

In the EVPN multihoming scenario, the multicast label B has the S-bit set to 1 when it is the only label in the label stack. In this case, the BUM packet needs to be flooded on

all the local ESIs on the DF PE device. But the label B has the S-bit set to 0 when split horizon label A is the innermost label in the label stack. In this case, the BUM packets need to be flooded on all local ESIs on the DF PE device, except the ESI that maps to the split horizon label A.

Assuming that packets originated from a multihomed CE device to a non-DF PE device on multihomed segment ESI1, when the non-DF PE device sends this packet to the DF PE device, the ESI label that the DF advertised to the non-DF PE device in its autodiscovery route per Ethernet segment is pushed first. The non-DF PE device also pushes the inclusive multicast label that the DF PE device advertised in its inclusive multicast route and further pushes the LSP label. The MPLS header thus contains two labels within a 32-bit field.

The base EVPN functionality uses a table-next hop to stitch the MPLS table with its corresponding EVPN EVI table. In the EVPN EVI table, the mac-lookup is performed to switch the packet.

The following routes are programmed in the mpls.0 table for EVPN multicast:

- The (multicast-label, S=1) route points to the EVPN-EVI table-next hop.
- The (multicast-label, S=0) route points to the MPLS table-next hop. This route loops the packet back to the MPLS table after popping the multicast-label.
- The (split horizon-label) route points to the EVPN-EVI table-next hop. This is the same table-next hop that is used by the multicast-label, S=1 route.

New EVPN Route Types

EVPN multihoming mode supports the following EVPN route types:

- Autodiscovery route per Ethernet segment
- Autodiscovery route per EVPN instance (EVI)
- Ethernet segment route

These route types conform to the following naming convention:

<route-type>:<RD>::<esi>::<route-specific>/304

For example:

Autodiscovery route per Ethernet
segment—1:10.255.0.2:0::112233445566778899::0/304

Autodiscovery route per EVI—1:100.100.100.1:1::22222222222222222222::0/304

Ethernet segment route—4:10.255.0.1:0::112233445566778899:10.255.0.1/304

where:

- **route-type**—Type of EVPN route.
 - 1—Autodiscovery route per Ethernet segment.
 - 1—Autodiscovery route per EVI.

- 4—Ethernet segment route.
- 5—Route with VXLAN/MPLS encapsulation

- **RD**—Route distinguisher value.

The route distinguisher value is set to the IP address of the PE router followed by 0.

- **esi**—Ethernet segment identifier. Displayed as 10 bytes of hexadecimal bytes, and leading 00 bytes are not displayed.
- **route-specific**—Differs per route type.
 - Autodiscovery route per Ethernet segment and autodiscovery route per EVI—This value is an MPLS label.



NOTE: The MPLS label is displayed in the extensive output, although it is not included in the prefix.

- Ethernet segment route—This value is the originating IP address.
- **304**—Maximum number of bits in an EVPN route. This is not very useful information and could be removed from the display. However, it might be useful in quickly identifying an EVPN route, either visually or with match operators.

Update to the MAC Forwarding Table

In active-standby EVPN multihoming, the MAC addresses are treated as routable addresses, and the MP-IBGP protocol is used to carry the customer MAC addresses. MAC learning at the PE routers does not occur in the data plane but in the control plane. This leads to more control applied in terms of the learning mechanism.

A PE router performs MAC learning in the data plane for packets coming from a customer network for a particular EVI. For CE MAC addresses that are behind other PE routers, the MAC addresses are advertised in BGP NLRI using a new MAC advertisement route type.

The MAC learning is of two types:

- Local MAC learning—PE routers must support the local MAC learning process through standard protocols.
- Remote MAC learning—Once the local learning process is completed, the PE routers can advertise the locally learned MAC address to remote PE router nodes through MP-IBGP. This process of receiving the remote MAC addresses of attached customers through MP-IBGP is known as the remote MAC learning process.

The MAC advertisement route type is used to advertise locally learned MAC addresses in BGP to remote PE routers. If an individual MAC address is advertised, the IP address field corresponds to that MAC address. If the PE router sees an ARP request for an IP address from a CE device, and if the PE router has the MAC address binding for that IP address, the PE router performs ARP proxy and responds to the ARP request.



NOTE: The ARP proxy is performed only for the gateway and not for the host.

The MPLS label field depends on the type of allocation. The PE router can advertise a single MPLS label for all MAC addresses per EVI, which requires the least number of MPLS labels and saves the PE router memory. However, when forwarding to the customer network, the PE router must perform a MAC lookup which can cause a delay and increase the number of CPU cycles.

Traffic Flow

In EVPN multihoming, traffic flow is performed in the forwarding-plane. Flood routes are created for flooding the packets, and are used in the following scenarios:

- When a packet is received on a local ESI
- When a packet is received from the core

The traffic flows in EVPN multihoming can be based on the two traffic types:

- Unicast traffic

Unicast traffic is a point-to-point communication with one sender and one receiver. In a multihomed EVPN, unicast traffic is forwarded as follows:

- In active-standby mode
 - CE to core—Traffic is learned and forwarded by the DF PE router.
 - Core to CE—The remote PE router learns the MAC addresses from the DF, and forwards all unicast traffic to the DF PE router.
- In active-active mode
 - CE to core—Traffic is load-balanced to all the connected multihomed PE devices.
 - Core to CE—Traffic from the remote PE devices is load-balanced to all the multihomed PE devices connected to the remote CE device.
- BUM traffic

Traffic that is sent to multiple destinations, including broadcast traffic, unknown unicast traffic that is broadcast in the Ethernet segment, and multicast traffic is known as BUM traffic. In a multihomed EVPN, BUM traffic is forwarded as follows:

- In active-standby mode
 - CE to core—The CE device floods any BUM traffic to all the links in the Ethernet segment. The DF PE router with the active path forwards the BUM packets to the core. The BDF PE router in the standby mode drops all the traffic from the CE device, because the EVPN multihomed status of the interface is in blocking state.
 - Core to CE—The remote PE routers flood all BUM traffic to both the DF and BDF PE routers. Only the DF forwards the BUM traffic to the CE device. The BDF PE

router drops all the traffic, because the EVPN multihomed status of the interface is in blocking state.

- In active-active mode

Based on the requirements, flooding and switching among local ESIs can be enabled or disabled in the active-active mode. This is referred to as the no-local-switching behavior.

The core of EVPN service provides a full-mesh connectivity among the multihomed PE devices. Because of this, EVPN uses split horizon in the core, so a packet received from the core is never switched or flooded back to the core. Instead, ingress replication is used to replicate the packets to the remote PE devices.

To flood packets to remote PE devices, the multicast and the split horizon next hops are used. The multicast next hop tunnels the packet with the inclusive multicast label, and the split horizon next hop tunnels the packet with a multicast-label and a split horizon label. One such next hop is required per multihomed ESI per remote PE device.

The following flood routes are used in the active-active mode:

- All-CE flood route

This flood route is used by the local ESIs for the following:

- Flooding the packet on the local ESIs (when local-switching is allowed).
- Flooding the packet to the remote PE devices. The remote PE devices flood the packet on their local ESIs.

Because BUM traffic is forwarded only by the Designated Forwarder (DF), and not by the non-DF multihomed PE devices, the non-DFs use the split horizon next hop to flood this packet to other PE devices. However, the multihomed local ESIs for which the PE device is a non-DF does not participate in the flooding.

The all-CE flood route is not used by the non-DF ESIs, and the next hop for these flood routes is created accordingly. In such cases, the non-DF ESI flood route is used.

- All-VE flood route

This flood route is used when the packet is received from the core. It is used for flooding the packet received from the core to the local ESIs. Because the packet received from the core can come with multicast-label only or with both multicast-label and split horizon label, appropriate forwarding rules must be followed to drop the packet on the multihomed ESI that maps to the split horizon label.

- Non-DF flood route

This flood route is used for the following:

- Flooding the packet on the local ESIs.
- Flooding the packet to the remote PE devices using ingress replication with SH-label for the DF for the ESI.

Aliasing

Starting in Junos OS Release 15.1, Junos OS supports aliasing in an EVPN. Aliasing is the ability of a remote PE device to load balance Layer 2 unicast traffic on all the other PE devices that have same Ethernet segment towards a CE device.

- [Aliasing in the Active-Active Mode on page 133](#)
- [Aliasing and Autodiscovery Routes on page 133](#)
- [Aliasing and Label Route on page 134](#)
- [Aliasing and Unicast Packet Forwarding on page 134](#)

Aliasing in the Active-Active Mode

In [Figure 7 on page 124](#), aliasing in the active-active mode works as follows:

1. ESI1 is configured on Routers PE1, PE2, and PE3. Routers PE1, PE2, and PE3 advertise the autodiscovery route per Ethernet segment for ESI1.
2. Device CE1 sends Layer 2 traffic with source MAC address (MAC1) to Router PE1.
3. Router PE1 learns the MAC1 address on (ESI1, vlan X) and advertises it to all PE routers using BGP.
4. Router PE4 receives the MAC1 route through BGP.
5. Because Router PE4 also received the autodiscovery route per EVI from Routers PE2 and PE3, it knows that MAC1 must be reachable through Routers PE2 and PE3. Router PE4 builds its forwarding state to load-balance the Layer 2 traffic for MAC1 among Routers PE1, PE2, and PE3.

Aliasing and Autodiscovery Routes

Autodiscovery routes from Routers PE2 and PE3 can come in any order. As a result, these routes are installed by the Layer 2 process as follows:

1. After receiving MAC1 from Router PE1, and if any autodiscovery routes have not been received by Router PE4, MAC1 is programmed by PE4 with a next hop pointing toward Router PE1. When PE4 receives the autodiscovery route from Router PE2 for the same ESI, the next hop is installed so the traffic for MAC1 is load-balanced to Routers PE1 and PE2. When PE4 receives the autodiscovery route from Router PE3 for the same ESI, the next hop is updated to load-balance the traffic for MAC1 among Routers PE1, PE2, and PE3.
2. If Router PE4 has already received the autodiscovery routes from more than one PE device (PE1, PE2, and PE3), PE4 installs the MAC routes with the multi-destination next hop.

Aliasing and Label Route

Any PE device that advertises the autodiscovery route per EVI with a valid MPLS label programs the advertised label in the mpls.0 routing table. For instance, if Router PE2 advertised the autodiscovery route per EVI with label A, the mpls.0 entry is as follows:

Label A route points to the EVPN-EVI table-next hop.

When the remote Router PE4 sends a unicast data packet toward Router PE2 with this label A, lookup is done in Router PE2's forwarding table, and as a result of this lookup, the packet is forwarded on ES11.

Aliasing and Unicast Packet Forwarding

When the unicast packets for MAC1 come from the remote Router PE4 to Router PE2, there could be two cases:

- Router PE2 also received the same set of MACs on its link to ES11—In this case, local routes are preferred and as a result of the MAC lookup, packets are forwarded to ES11.
- Router PE2 has not received the same set of MACs on its link to ES11—In this case, Router PE2 still installs MAC routes pointing to ES11, although MACs are remotely learned from Router PE1. As a result, the packets are forwarded to ES11.

EVPN Active-Active Multihoming and Multichassis Link Aggregation

When a CE device is configured with a LAG toward the PE devices, the following two options are available to run LACP on the PE devices:

- Configure the same LACP system ID on all the PE devices.
- Configure multichassis link aggregation on the PE devices.

When multichassis link aggregation is configured with EVPN, a reduced set of procedures for active-active multichassis link aggregation are required. These procedures provide link and node level redundancy. The multichassis link aggregation is completely transparent to the CE device, and is realized as pure LAG. Multichassis link aggregation operates at the port level as well. This essentially means that if multichassis link aggregation is configured as active-active, all VLANs on the multichassis link aggregation ports work in the active-active multihoming mode.

When multichassis link aggregation is configured along with EVPN, the following is considered:

- Both multichassis link aggregation and EVPN ESI must be enabled to work in the active-active mode only.
- The following functions are not required for multichassis link aggregation with EVPN:
 - Mac synchronization—This is performed in the BGP control plane of EVPN.
 - ICL linking—This is handled by the aliasing feature of EVPN.
 - ARP synchronization—This is handled by the BGP control plane with IRB functionality.

EVPN Active-Active Multihoming and IRB

When IRB is configured, the EVPN routes contain both MAC and IP information. The active-active multihoming requires ARP synchronization among the multihomed PE devices because the ARP responses can get hashed to a particular PE device.

Sample Configuration

The following is a sample configuration for EVPN active-standby multihoming on the following types of interfaces:

- Ethernet interface configuration

```
ge-0/1/2 {
  encapsulation ethernet-bridge;
  esi XX:XX:XX:XX:XX:XX:XX:XX:XX:XX;
  unit 0 {
    family bridge;
  }
}
```

- Single VLAN interface configuration

```
ge-0/1/3 {
  encapsulation extended-vlan-bridge;
  esi XX:XX:XX:XX:XX:XX:XX:XX:XX:XX;
  vlan-tagging
  unit 0 {
    family bridge;
    vlan-id 1;
  }
}
```



NOTE:

- An ESI value of 0 and all FFs are reserved and are not used for configuring a multihomed Ethernet segment.
- Two interfaces in the same EVI cannot be configured with the same ESI value.

The following is a sample routing instance configuration for EVPN active-standby multihoming:

- Routing instance configuration

```
routing-instances {
  evpn-0 {
    instance-type evpn;
    route-distinguisher value;
    vrf-target value;
    vlan-id vlan-ID;
    interface ge-0/1/2.0;
    interface ge-1/1/1.0;
    interface ge-2/2/2.0;
  }
}
```

```
protocols {  
  evpn {  
    designated-forwarder-election hold-time time;  
  }  
}
```



NOTE: With the active-standby mode configuration, the autodiscovery route per Ethernet segment is advertised with the active-standby bit set to 1 for each Ethernet segment.

Designated Forwarder Election

The following sections discuss DF election:

- [DF Election Roles on page 136](#)
- [DF Election as Per RFC 7432 on page 136](#)
- [Preference-Based DF Election on page 138](#)
- [DF Election for Virtual Switch on page 140](#)
- [Handling Failover on page 140](#)

DF Election Roles

The designated forwarder (DF) election process involves selecting the designated forwarder (DF) PE router and the backup designated forwarder (BDF) or a non-DF (non-designated forwarder PE router roles.

- **DF**—The MAC address from the customer site is reachable only through the PE router announcing the associated MAC advertisement route. This PE router is the primary PE router that is selected to forward BUM traffic to the multihomed CE device, and is called the designated forwarder (DF) PE router.
- **BDF**—Each PE router in the set of other PE routers advertising the autodiscovery route per Ethernet segment for the same ESI, and serving as the backup path in case the DF encounters a failure, is called a backup designated forwarder (BDF) or a non-DF (non-designated forwarder) PE router.

As a result of the DF election process, if a local PE router is elected as the BDF, the multihomed interface connecting to the customer site is put into a blocking state for the active-standby mode. The interface remains in the blocking state until the PE router is elected as the DF for the Ethernet segment that the interface belongs to.

DF Election as Per RFC 7432

- [DF Election Procedure on page 137](#)
- [DF Election Trigger on page 137](#)

DF Election Procedure

The default procedure for DF election at the granularity of the ESI and EVI is referred to as service carving. With *service carving*, it is possible to elect multiple DFs per Ethernet segment (one per EVI) in order to perform load-balancing of multidestination traffic destined for a given Ethernet segment. The load-balancing procedures carve up the EVI space among the PE nodes evenly, in such a way that every PE is the DF for a disjoint set of EVIs.

The procedure for service carving is as follows:

1. When a PE router discovers the ESI of the attached Ethernet segment, it advertises an autodiscovery route per Ethernet segment with the associated ES-import extended community attribute.
2. The PE router then starts a timer (default value of 3 seconds) to allow the reception of the autodiscovery routes from other PE nodes connected to the same Ethernet segment. This timer value must be the same across all the PE routers connected to the same Ethernet segment.

The default wait timer can be overwritten using the **designated-forwarder-election hold-time** configuration statement.

3. When the timer expires, each PE router builds an ordered list of the IP addresses of all the PE nodes connected to the Ethernet segment (including itself), in increasing numeric order. Every PE router is then given an ordinal indicating its position in the ordered list, starting with 0 as the ordinal for the PE with the numerically lowest IP address. The ordinals are used to determine which PE node is the DF for a given EVI on the Ethernet segment.
4. The PE router that is elected as the DF for a given EVI unblocks traffic for the Ethernet tags associated with that EVI. The DF PE unblocks multidestination traffic in the egress direction toward the Ethernet segment. All the non-DF PE routers continue to drop multidestination traffic (for the associated EVIs) in the egress direction toward the Ethernet segment.

In [Figure 7 on page 124](#), the election of the DF for active-active multihoming is performed among Routers PE1, PE2, and PE3. As a result of this DF election, each one of these routers can become the DF for a particular VLAN from a range of VLANs configured on ESI1. The DF is responsible for forwarding BUM traffic on that ESI and VLAN for which it is elected as the DF. The non-DF PE routers block the BUM traffic on that particular Ethernet segment.

DF Election Trigger

In general, a DF election process is triggered in the following conditions:

- When an interface is newly configured with a nonzero ESI, or when the PE router transitions from an isolated-from-the-core (no BGP session) state to a connected-to-the-core (has established BGP session) state, a wait timer is imposed.

By default, the interface is put into a blocking state until the PE router is elected as the DF.

- After completing a DF election process, a PE router receives a new Ethernet segment route or detects the withdrawal of an existing Ethernet segment route, without an imposed wait timer.
- When an interface of a non-DF PE router recovers from a link failure, the PE router has no knowledge of the wait time imposed by other PE routers. As a result, no wait timer is imposed for the recovered PE router to avoid traffic loss.

Preference-Based DF Election

The DF election based on RFC 7432 does not meet some of the operational requirements needed by some service providers. As a solution to this, starting with Junos OS Release 17.3, the DF election in a multihoming EVPN network can be controlled by using an administrative preference value for an ESI.

In the default DF election procedure (as specified in RFC 7432), the DF is elected randomly from one of the multihoming devices with modulo operation. With the preference-based DF election, the DF is elected manually using interface configuration options, such as the preference value, the Don't Preempt (DP) bit, and router ID or loopback address.

- [Preference-Based DF Election Procedure on page 138](#)
- [DF Election Algorithm Mismatch on page 139](#)
- [DF Election Algorithm Migration on page 139](#)
- [Changing Preference for Maintenance on page 139](#)

Preference-Based DF Election Procedure

The preference-based DF election is supported on EVPN and PBB-EVPN, and allows for manually electing a DF. This is useful when there is a need to choose the DF based on interface attributes, such as the bandwidth associated with an interface.

The preference-based DF election is executed as follows:

1. The DF election type and preference value are configured under an ESI. By default, the preference-based DF election type is based on the modulo (MOD) operation.
2. The configured preference value and DP bit are advertised to the multihoming PE devices using the DF election extended community in the type 4 routes.
3. After receiving the type 4 route, the PE device builds the list of candidate DF devices, in the order of the preference value, DP bit, and IP address.
4. When the DF timer expires, the PE device selects the DF based on the highest preference value.

By default, the DF is elected based on highest preference per EVI. However, the preference-based DF election allows for electing the DF based on the lowest preference

value when the **designated-forwarder-preference-least** statement is included at the **[edit routing-instances routing-instance-name protocols evpn]** hierarchy level.



NOTE: The **designated-forwarder-preference-least** configuration should be the same on both the multihoming EVIs; otherwise there can be two DFs causing traffic loss or loop.

5. When the same preference value is configured, then the PE device selects the DF based on the DP bit. When the DP bit is also the same, the DF is elected based on the lowest IP address.

DF Election Algorithm Mismatch

When there is a mismatch between a locally configured DF election algorithm and a remote PE device's DF election algorithm, then all the PE devices should fall back to the default DF election as specified in RFC 7432.

DF Election Algorithm Migration

During the migration of the old DF election to the new DF election, it is expected to change the configuration during the maintenance window by bringing down the ESI, and changing the DF election algorithm.

To do the migration, do the following:

1. After a software upgrade, on the non-DF device bring down all the interfaces that have the same ESI.
2. Configure the new DF election algorithm on the DF PE.
3. Configure the DF election algorithm on other multihoming PE devices.
4. Bring up all the interfaces on the non-DF PE devices.

Changing Preference for Maintenance

After migrating the DF election algorithm, and all the multihoming PE device are running the preference-based DF election algorithm, maintenance tasks required on the existing DF can be executed by simply changing the configured preference value. This changes the DF for a given ESI.

To change the DF for a given ESI:

1. Change the preference value to a higher value on the current non-DF device.
2. Change the preference value to a lower value on the current DF device.



NOTE: Changing the preference value for an ESI can lead to some traffic loss during the short duration required to integrate the delay in the updated BGP route propagation with the new preference value.

DF Election for Virtual Switch

The virtual switch allows multiple bridge domains in a single EVPN instance (EVI). The virtual switch also supports trunk and access ports. Junos OS allows flexible Ethernet services on the port; therefore different VLANs on a single port can be part of different EVIs.

The DF election for virtual switch depends on the following:

- Port mode—Sub-interface, trunk interface, and access port
- EVI mode—Virtual switch with EVPN and EVPN-EVI

In the virtual switch, multiple Ethernet tags can be associated with a single EVI, wherein the numerically lowest Ethernet tag value in the EVI is used for the DF election.

Handling Failover

A failover can occur when:

- The DF PE router loses its DF role.
- There is a link or port failure on the DF PE router.

On losing the DF role, the customer-facing interface on the DF PE router is put in the blocking state.

In the case of link or port failure, a DF election process is triggered, resulting in the BDF PE router to be selected as the DF. At that time, unicast traffic and BUM flow of traffic are affected as follows:

- [Unicast Traffic on page 140](#)
- [BUM Traffic on page 141](#)

Unicast Traffic

- CE to Core—The CE device continues to flood traffic on all the links. The previous BDF PE router changes the EVPN multihomed status of the interface from the blocking state to the forwarding state, and traffic is learned and forwarded through this PE router.
- Core to CE—The failed DF PE router withdraws the autodiscovery route per Ethernet segment and the locally-learned MAC routes, causing the remote PE routers to redirect traffic to the BDF.



NOTE: The transition of the BDF PE router to the DF role can take some time, causing the EVPN multihomed status of the interface to continue to be in the blocking state, resulting in traffic loss.

BUM Traffic

- CE to Core—All the traffic is routed toward the BDF.
- Core to CE—The remote PE routers flood the BUM traffic in the core.

ESIs on Physical, Aggregated Ethernet, and Logical Interfaces

In releases before Junos OS Release 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI only on a physical or aggregated Ethernet interface, for example, **set interfaces ae0 esi 00:11:22:33:44:55:66:77:88:99**. If you specify an ESI on a physical or aggregated Ethernet interface, keep in mind that an ESI is a factor in the designated forwarder (DF) election process. For example, assume that you configure EVPN multihoming active-standby on aggregated Ethernet interface ae0, and given the ESI configured on ae0 and other determining factors, the DF election results in ae0 being in the down state. Further, all logical interfaces configured on aggregated Ethernet interface ae0, for example, **set interfaces ae0 unit 1** and **set interfaces ae0 unit 2** are also in the down state, which renders logical interfaces ae0.1 and ae0.2 unable to provide services to their respective customer sites (VLANs).

To better utilize logical interfaces in EVPN multihoming active-standby or active-active mode, starting with Junos OS Releases 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI on a logical interface. As a result, even if a logical interface is a non-DF, other logical interfaces on the same physical or aggregated Ethernet interface are still able to provide services to their respective customer sites (VLANs).

For more information, see [“Example: Configuring an ESI on a Logical Interface With EVPN Multihoming” on page 195](#).

Convergence in an EVPN Network

When there are changes in the network topology in a large-scale EVPN system, the convergence time might be significant. You can prioritize NLRI updates that are critical to route selection in routing policies to improve convergence. [Table 3 on page 141](#) lists the NLRI route types and the priority that must be configured in the routing policy.

Table 3: Priority for NLRI Route Type

NLRI Route Type	Description	Priority
NLRI Route Type 1	Ethernet auto-discovery route—Type 1 supports fast convergence and aliasing and is used to signal MAC mass withdrawal.	High
NLRI Route Type 2	MAC/IP advertisement route—Type 2 is used to advertise MAC addresses and IP addresses in EVPN networks.	Low

Table 3: Priority for NLRI Route Type (*continued*)

NLRI Route Type	Description	Priority
NLRI Route Type 3	Inclusive multicast Ethernet tag—Type 3 is used to set up a path for BUM traffic.	Low
NLRI Route Type 4	Ethernet segment route—Type 4 is used in the selection of a designated forwarder.	High

To prioritize the NLRI route type, set the **bgp-output-queue-priority** priority for **nlri-route-type** at the **[edit policy-options policy-statement]** hierarchy level on all provider edge routers and route reflectors in the EVPN network. In this example, a high priority was configured for NLRI route type 1 and NLRI route type 4.

```

user@PE1#show policy-options
policy-statement evpn-rt-priority-policy {
  term 1 {
    from {
      family evpn;
      nlri-route-type 1;
    }
    then {
      bgp-output-queue-priority priority 16;
    }
  }
  term 2 {
    from {
      family evpn;
      nlri-route-type 2;
    }
    then {
      bgp-output-queue-priority priority 1;
    }
  }
  term 3 {
    from {
      family evpn;
      nlri-route-type 3;
    }
    then {
      bgp-output-queue-priority priority 2;
    }
  }
  term 4 {
    from {
      family evpn;
      nlri-route-type 4;
    }
    then {
      bgp-output-queue-priority priority 16;
    }
  }
}

```



NOTE: There are 17 prioritized output queues: an expedited queue that has the highest priority, and 16 numbered queues for which 1 is the lowest priority and 16 is the highest.

For more information about how to configure routing policies, see *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*.

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, Junos OS supports Dynamic List Next Hops in an EVPN network.
16.1R4	Starting with Junos OS Releases 16.1R4 and 16.2R2, all EX9200 switches support the active-active mode for EVPN multihoming.
15.1F6	To better utilize logical interfaces in EVPN multihoming active-standby or active-active mode, starting with Junos OS Releases 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI on a logical interface. As a result, even if a logical interface is a non-DF, other logical interfaces on the same physical or aggregated Ethernet interface are still able to provide services to their respective customer sites (VLANs).
15.1	Starting in Junos OS Release 15.1, Junos OS supports aliasing in an EVPN.
14.1x53-D30	Starting with Junos OS Release 14.1x53-D30, QFX5100 switches support the active-active mode of operation for EVPN multihoming.
14.1R4	Starting with Junos OS Release 14.1R4, 14.2, 15.1F6, and 16.1R1, Junos OS supports the active-active mode for EVPN multihoming on MX Series routers.

Related Documentation

- [Configuring Dynamic List Next Hop](#)
- [Example: Configuring EVPN Active-Standby Multihoming](#)
- [Example: Configuring EVPN Active-Active Multihoming on page 147](#)

Configuring EVPN Active-Standby Multihoming

You can configure an Ethernet VPN (EVPN) with multihoming support to provide multihoming functionality with active-standby redundancy mode of operation in the EVPN and virtual switch routing instance. This mode enables autodiscovery of Ethernet segments, Ethernet segment route construction, and Ethernet segment identifier (ESI) label assignment.

When configuring active-standby EVPN multihoming, be aware of the following limitations:

- An interface or ESI can be attached to more than one EVPN instance (EVI), with a maximum limit of 200 EVIs per ESI.
- For an EVPN routing instance, only one logical interface per physical interface or ESI can be attached to an EVI.
- For a virtual switch routing instance, only one logical interface per physical interface or ESI can be configured under a bridge domain.
- All the PE routers in the network topology should be running Junos OS Release 14.1 or later releases, which are based on the EVPN draft-ietf-l2vpn-evpn-03. Junos OS releases prior to 14.1 support the older version of the EVPN draft, causing interoperability issues when Junos OS Release 14.1 and a previous release are running.

Before you begin:

1. Configure the router interfaces.
2. Configure the router ID and autonomous system number for the device.
3. Configure OSPF or any other IGP protocol.
4. Configure a BGP internal group.
5. Include the EVPN signaling network layer reachability information (NLRI) to the internal BGP group.
6. Configure LDP.
7. Configure MPLS.
8. Configure RSVP MPLS LSP or GRE tunnels.

To configure the PE device:

1. Enable EVPN active-standby multihoming on the multihomed interfaces.

```
[edit interfaces]
user@PE1# set interface-name vlan-tagging
user@PE1# set interface-name encapsulation flexible-ethernet-services
user@PE1# set interface-name esi esi-value
user@PE1# set interface-name esi single-active
user@PE1# set interface-name unit 0 encapsulation vlan-bridge
user@PE1# set interface-name unit 0 vlan-id VLAN-ID
```

For example:


```
[edit interfaces]
user@PE1# set ge-0/0/4 vlan-tagging
user@PE1# set ge-0/0/4 encapsulation flexible-ethernet-services
user@PE1# set ge-0/0/4 esi 00:22:44:66:88:00:22:44:66:88
user@PE1# set ge-0/0/4 esi single-active
user@PE1# set ge-0/0/4 unit 0 encapsulation vlan-bridge
user@PE1# set ge-0/0/4 unit 0 vlan-id 300
```

2. In configuration mode, go to the following hierarchy level:

```
[edit]
user@PE1# edit routing-instances
```

3. Configure the virtual switch routing instance.

```
[edit routing-instances]
user@PE1# set virtual-switch-instance instance-type virtual-switch
```

4. Configure the extended VLAN list for the virtual switch routing instance.

```
[edit routing-instances]
user@PE1# set virtual-switch-instance protocols evpn extended-vlan-list VLAN-ID
```

5. Set the type for the bridging domain in the virtual switch routing instance.

```
[edit routing-instances]
user@PE1# set virtual-switch-instance bridge-domains bridge-domain-name
domain-type bridge
```

6. Set the VLAN identifier for the bridging domain in the virtual switch routing instance.

```
[edit routing-instances]
user@PE1# set virtual-switch-instance bridge-domains bridge-domain-name vlan-id
VLAN-ID
```

7. Configure the interfaces for the virtual switch routing instance.

```
[edit routing-instances]
user@PE1# set virtual-switch-instance bridge-domains bridge-domain-name interface
interface-name
user@PE1# set virtual-switch-instance bridge-domains bridge-domain-name
routing-interface interface-name
```

8. Configure the route distinguisher for the virtual switch routing instance.

```
[edit routing-instances]
user@PE1# set virtual-switch-instance route-distinguisher route-distinguisher-value
```

9. Configure the VPN routing and forwarding (VRF) target community for the virtual switch routing instance.

```
[edit routing-instances]
user@PE1# set virtual-switch-instance vrf-target vrf-target
```

10. Configure the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance instance-type evpn
```

11. Set the VLAN identifier for the bridging domain in the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance vlan-id VLAN-ID
```

12. Configure the interface names for the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance interface interface-name
user@PE1# set evpn-instance routing-interface interface-name
```

13. Configure the route distinguisher for the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance route-distinguisher route-distinguisher-value
```

14. Configure the VPN routing and forwarding (VRF) target community for the EVPN routing instance.

```
[edit routing-instances]
user@PE1# set evpn-instance vrf-target vrf-target
```

15. Configure the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance instance-type vrf
```

16. Configure the interface names for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance interface interface-name
```

17. Configure the route distinguisher for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance route-distinguisher route-distinguisher-value
```

18. Configure the VPN routing and forwarding (VRF) target community for the VRF routing instance.

```
[edit routing-instances]
user@PE1# set vrf-instance vrf-target vrf-target
user@PE1# set vrf-instance vrf-table-label
```

19. Verify and commit the configuration.

For example:

```
[edit routing-instances]
```

```

user@PE1# set ALPHA instance-type virtual-switch
user@PE1# set ALPHA route-distinguisher 10.255.0.1:100
user@PE1# set ALPHA vrf-target target:100:100
user@PE1# set ALPHA protocols evpn extended-vlan-list 100
user@PE1# set ALPHA bridge-domains ONE domain-type bridge
user@PE1# set ALPHA bridge-domains ONE vlan-id 100
user@PE1# set ALPHA bridge-domains ONE interface ae0.0
user@PE1# set ALPHA bridge-domains ONE interface ge-0/0/2.0
user@PE1# set ALPHA bridge-domains ONE routing-interface irb.0
user@PE1# set BETA instance-type evpn
user@PE1# set BETA vlan-id 300
user@PE1# set BETA interface ge-0/0/4.0
user@PE1# set BETA interface ae1.0
user@PE1# set BETA routing-interface irb.1
user@PE1# set BETA route-distinguisher 10.255.0.1:300
user@PE1# set BETA vrf-target target:300:300
user@PE1# set DELTA instance-type vrf
user@PE1# set DELTA interface irb.0
user@PE1# set DELTA interface irb.1
user@PE1# set DELTA route-distinguisher 10.255.0.1:200
user@PE1# set DELTA vrf-target target:200:200
user@PE1# set DELTA vrf-table-label

[edit]
user@PE1# commit
commit complete

```

Related Documentation

- [Example: Configuring EVPN Active-Standby Multihoming](#)

Example: Configuring EVPN Active-Active Multihoming

This example shows how to configure Ethernet VPN (EVPN) for multihomed customer edge devices in the active-active redundancy mode, so the Layer 2 unicast traffic can be load-balanced across all the multihomed links on and toward the CE device.

- [Requirements on page 147](#)
- [Overview and Topology on page 148](#)
- [Configuration on page 149](#)
- [Verification on page 178](#)

Requirements

This example uses the following hardware and software components:

- Five MX Series 3D Universal Edge Routers with MPC interfaces only, where:
 - Three devices are configured as provider edge (PE) routers connected to a common multihomed customer site.
 - One device is configured as a remote PE router connected to a single-homed customer site.

- Six customer edge (CE) devices, with one multihomed CE device and the rest of the CE devices being single-homed to each of the PE routers.
- Junos OS Release 16.1 or later running on all the PE routers.



NOTE: The EVPN multihoming active-active mode of operation is supported in Junos OS Releases 16.1 and later releases.

Starting with Junos OS Release 16.1R4, EVPN multihoming active-active mode is supported on all EX9200 switches. For information about configuration specific to EX9200 switches, see [EX9200 Configuration on page 37](#).

Before you begin:

1. Configure the router interfaces.
2. Configure IS-IS or any other IGP protocol.
3. Configure BGP.
4. Configure LDP.
5. Configure MPLS.
6. Configure RSVP MPLS LSP or GRE tunnels.

Overview and Topology

Starting with Junos OS Release 15.1, the EVPN solution on MX Series routers with MPC interfaces is extended to provide multihoming functionality in the active-active redundancy mode of operation. This feature enables load balancing of Layer 2 unicast traffic across all the multihomed links on and toward a customer edge device.

The EVPN active-active multihoming feature provides link-level and node-level redundancy along with effective utilization of resources.

To enable EVPN active-active multihoming, include the **all-active** statement at the **[edit interfaces esi]** hierarchy level.

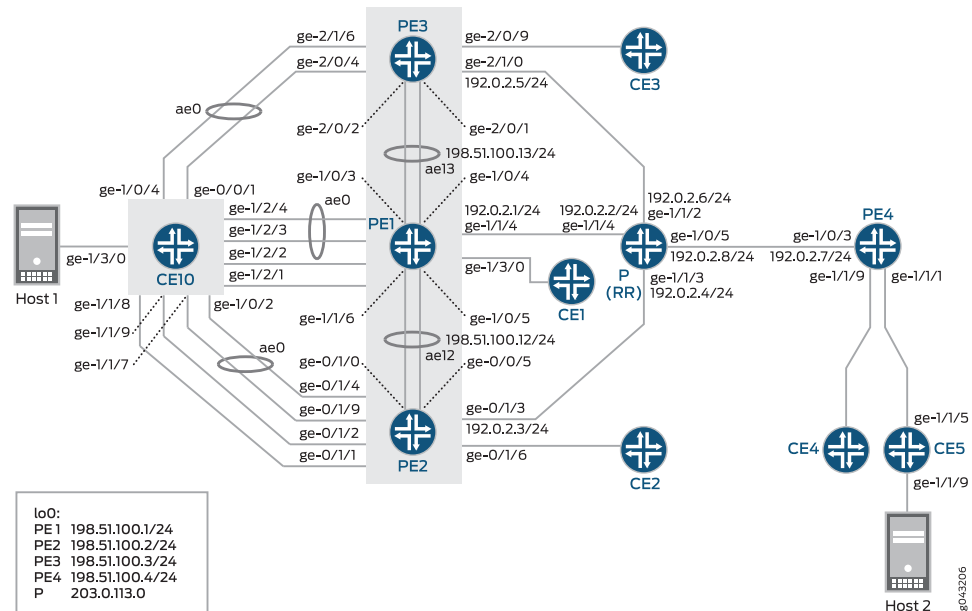
In [Figure 8 on page 149](#), the core consists of four provider edge (PE) routers and a provider router (P) that is configured as a route reflector (RR). Router CE10 is multihomed to Routers PE1, PE2, and PE3. Each PE router is also connected to a single-homed customer site.

There are three routing instances running in the topology – VS-1, VS-2, and mhevpn, along with the default routing instance. The routing instances share nine VLANs with three ESIs each. The VS-1 and VS-2 routing instances are configured as a virtual switch type of routing instance, and the mhevpn routing instance is an EVPN routing instance.

Three aggregated bundles – ae0, ae1, and ae2 – are used to connect the multihomed CE device, CE10, to Routers PE1, PE2, and PE3. These aggregated bundles are configured for

three ESIs each. The aggregated bundles, ae12 and ae13, are used to interconnect Routers PE1 and PE2, and PE1 and PE3, respectively.

Figure 8: EVPN Active-Active Multihoming Topology



Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
CE10 set chassis aggregated-devices ethernet device-count 20
set chassis network-services enhanced-ip
set interfaces ge-1/0/0 gigether-options 802.3ad ae0
set interfaces ge-1/0/2 gigether-options 802.3ad ae0
set interfaces ge-1/0/4 gigether-options 802.3ad ae0
set interfaces ge-1/1/7 gigether-options 802.3ad ae0
set interfaces ge-1/1/8 gigether-options 802.3ad ae2
set interfaces ge-1/1/9 gigether-options 802.3ad ae1
set interfaces ge-1/2/1 gigether-options 802.3ad ae2
set interfaces ge-1/2/2 gigether-options 802.3ad ae1
set interfaces ge-1/2/3 gigether-options 802.3ad ae0
set interfaces ge-1/2/4 gigether-options 802.3ad ae0
set interfaces ge-1/3/0 flexible-vlan-tagging
set interfaces ge-1/3/0 encapsulation flexible-ethernet-services
set interfaces ge-1/3/0 unit 0 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 0 vlan-id 10
set interfaces ge-1/3/0 unit 1 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 1 vlan-id 20
set interfaces ge-1/3/0 unit 2 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 2 vlan-id 30
```

```
set interfaces ge-1/3/0 unit 110 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 110 vlan-id 110
set interfaces ge-1/3/0 unit 120 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 120 vlan-id 120
set interfaces ge-1/3/0 unit 130 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 130 vlan-id 130
set interfaces ge-1/3/0 unit 210 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 210 vlan-id 210
set interfaces ge-1/3/0 unit 220 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 220 vlan-id 220
set interfaces ge-1/3/0 unit 230 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 230 vlan-id 230
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 unit 0 encapsulation vlan-bridge
set interfaces ae0 unit 0 vlan-id 10
set interfaces ae0 unit 110 encapsulation vlan-bridge
set interfaces ae0 unit 110 vlan-id 110
set interfaces ae0 unit 210 encapsulation vlan-bridge
set interfaces ae0 unit 210 vlan-id 210
set interfaces ae1 flexible-vlan-tagging
set interfaces ae1 encapsulation flexible-ethernet-services
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 unit 1 encapsulation vlan-bridge
set interfaces ae1 unit 1 vlan-id 20
set interfaces ae1 unit 120 encapsulation vlan-bridge
set interfaces ae1 unit 120 vlan-id 120
set interfaces ae1 unit 220 encapsulation vlan-bridge
set interfaces ae1 unit 220 vlan-id 220
set interfaces ae2 flexible-vlan-tagging
set interfaces ae2 encapsulation flexible-ethernet-services
set interfaces ae2 aggregated-ether-options minimum-links 1
set interfaces ae2 unit 2 encapsulation vlan-bridge
set interfaces ae2 unit 2 vlan-id 30
set interfaces ae2 unit 130 encapsulation vlan-bridge
set interfaces ae2 unit 130 vlan-id 130
set interfaces ae2 unit 230 encapsulation vlan-bridge
set interfaces ae2 unit 230 vlan-id 230
set routing-options forwarding-table export load-balancing-policy
set protocols l2-learning global-mac-table-aging-time 18000
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set bridge-domains bd10 domain-type bridge
set bridge-domains bd10 vlan-id 10
set bridge-domains bd10 interface ae0.0
set bridge-domains bd10 interface ge-1/3/0.0
set bridge-domains bd110 domain-type bridge
set bridge-domains bd110 vlan-id 110
set bridge-domains bd110 interface ae0.110
set bridge-domains bd110 interface ge-1/3/0.110
set bridge-domains bd120 domain-type bridge
set bridge-domains bd120 vlan-id 120
set bridge-domains bd120 interface ge-1/3/0.120
set bridge-domains bd120 interface ae1.120
set bridge-domains bd130 domain-type bridge
set bridge-domains bd130 vlan-id 130
```

```

set bridge-domains bd130 interface ge-1/3/0.130
set bridge-domains bd130 interface ae2.130
set bridge-domains bd20 domain-type bridge
set bridge-domains bd20 vlan-id 20
set bridge-domains bd20 interface ge-1/3/0.1
set bridge-domains bd20 interface ae1.1
set bridge-domains bd210 domain-type bridge
set bridge-domains bd210 vlan-id 210
set bridge-domains bd210 interface ae0.210
set bridge-domains bd210 interface ge-1/3/0.210
set bridge-domains bd220 domain-type bridge
set bridge-domains bd220 vlan-id 220
set bridge-domains bd220 interface ge-1/3/0.220
set bridge-domains bd220 interface ae1.220
set bridge-domains bd230 domain-type bridge
set bridge-domains bd230 vlan-id 230
set bridge-domains bd230 interface ge-1/3/0.230
set bridge-domains bd230 interface ae2.230
set bridge-domains bd30 domain-type bridge
set bridge-domains bd30 vlan-id 30
set bridge-domains bd30 interface ge-1/3/0.2
set bridge-domains bd30 interface ae2.2

```

```

CE5  set chassis aggregated-devices ethernet device-count 20
      set chassis network-services enhanced-ip
      set interfaces ge-1/1/5 gigether-options 802.3ad ae0
      set interfaces ge-1/1/9 flexible-vlan-tagging
      set interfaces ge-1/1/9 encapsulation flexible-ethernet-services
      set interfaces ge-1/1/9 unit 0 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 0 vlan-id 10
      set interfaces ge-1/1/9 unit 1 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 1 vlan-id 20
      set interfaces ge-1/1/9 unit 2 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 2 vlan-id 30
      set interfaces ge-1/1/9 unit 110 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 110 vlan-id 110
      set interfaces ge-1/1/9 unit 120 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 120 vlan-id 120
      set interfaces ge-1/1/9 unit 130 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 130 vlan-id 130
      set interfaces ge-1/1/9 unit 210 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 210 vlan-id 210
      set interfaces ge-1/1/9 unit 220 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 220 vlan-id 220
      set interfaces ge-1/1/9 unit 230 encapsulation vlan-bridge
      set interfaces ge-1/1/9 unit 230 vlan-id 230
      set interfaces ae0 flexible-vlan-tagging
      set interfaces ae0 encapsulation flexible-ethernet-services
      set interfaces ae0 unit 0 encapsulation vlan-bridge
      set interfaces ae0 unit 0 vlan-id 10
      set interfaces ae0 unit 1 encapsulation vlan-bridge
      set interfaces ae0 unit 1 vlan-id 20
      set interfaces ae0 unit 2 encapsulation vlan-bridge
      set interfaces ae0 unit 2 vlan-id 30
      set interfaces ae0 unit 110 encapsulation vlan-bridge

```

```
set interfaces ae0 unit 110 vlan-id 110
set interfaces ae0 unit 120 encapsulation vlan-bridge
set interfaces ae0 unit 120 vlan-id 120
set interfaces ae0 unit 130 encapsulation vlan-bridge
set interfaces ae0 unit 130 vlan-id 130
set interfaces ae0 unit 210 encapsulation vlan-bridge
set interfaces ae0 unit 210 vlan-id 210
set interfaces ae0 unit 220 encapsulation vlan-bridge
set interfaces ae0 unit 220 vlan-id 220
set interfaces ae0 unit 230 encapsulation vlan-bridge
set interfaces ae0 unit 230 vlan-id 230
set interfaces lo0 unit 6 family inet address 203.0.113.0/24
set interfaces lo0 unit 6 family iso address
    47.0005.80ff.f800.0000.0108.0000.6006.0070.0600
set routing-options forwarding-table export load-balancing-policy
set protocols l2-learning global-mac-table-aging-time 18000
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set bridge-domains bd10 domain-type bridge
set bridge-domains bd10 vlan-id 10
set bridge-domains bd10 interface ae0.0
set bridge-domains bd10 interface ge-1/1/9.0
set bridge-domains bd110 domain-type bridge
set bridge-domains bd110 vlan-id 110
set bridge-domains bd110 interface ae0.110
set bridge-domains bd110 interface ge-1/1/9.110
set bridge-domains bd120 domain-type bridge
set bridge-domains bd120 vlan-id 120
set bridge-domains bd120 interface ge-1/1/9.120
set bridge-domains bd120 interface ae0.120
set bridge-domains bd130 domain-type bridge
set bridge-domains bd130 vlan-id 130
set bridge-domains bd130 interface ge-1/1/9.130
set bridge-domains bd130 interface ae0.130
set bridge-domains bd20 domain-type bridge
set bridge-domains bd20 vlan-id 20
set bridge-domains bd20 interface ae0.1
set bridge-domains bd20 interface ge-1/1/9.1
set bridge-domains bd210 domain-type bridge
set bridge-domains bd210 vlan-id 210
set bridge-domains bd210 interface ae0.210
set bridge-domains bd210 interface ge-1/1/9.210
set bridge-domains bd220 domain-type bridge
set bridge-domains bd220 vlan-id 220
set bridge-domains bd220 interface ge-1/1/9.220
set bridge-domains bd220 interface ae0.220
set bridge-domains bd230 domain-type bridge
set bridge-domains bd230 vlan-id 230
set bridge-domains bd230 interface ge-1/1/9.230
set bridge-domains bd230 interface ae0.230
set bridge-domains bd30 domain-type bridge
set bridge-domains bd30 vlan-id 30
set bridge-domains bd30 interface ge-1/1/9.2
set bridge-domains bd30 interface ae0.2
```

PE1 set chassis aggregated-devices ethernet device-count 20


```
set chassis network-services enhanced-ip
set interfaces ge-1/0/3 gigether-options 802.3ad ae13
set interfaces ge-1/0/4 gigether-options 802.3ad ae13
set interfaces ge-1/0/5 gigether-options 802.3ad ae12
set interfaces ge-1/1/4 unit 0 family inet address 192.0.2.1/24
set interfaces ge-1/1/4 unit 0 family iso
set interfaces ge-1/1/4 unit 0 family mpls
set interfaces ge-1/1/6 gigether-options 802.3ad ae12
set interfaces ge-1/2/1 flexible-vlan-tagging
set interfaces ge-1/2/1 encapsulation flexible-ethernet-services
set interfaces ge-1/2/1 esi 00:33:33:33:33:33:33:33
set interfaces ge-1/2/1 esi all-active
set interfaces ge-1/2/1 unit 0 encapsulation vlan-bridge
set interfaces ge-1/2/1 unit 0 vlan-id 30
set interfaces ge-1/2/1 unit 130 family bridge interface-mode trunk
set interfaces ge-1/2/1 unit 130 family bridge vlan-id-list 130
set interfaces ge-1/2/1 unit 230 family bridge interface-mode trunk
set interfaces ge-1/2/1 unit 230 family bridge vlan-id-list 230
set interfaces ge-1/2/2 flexible-vlan-tagging
set interfaces ge-1/2/2 encapsulation flexible-ethernet-services
set interfaces ge-1/2/2 esi 00:22:22:22:22:22:22:22
set interfaces ge-1/2/2 esi all-active
set interfaces ge-1/2/2 unit 0 encapsulation vlan-bridge
set interfaces ge-1/2/2 unit 0 vlan-id 20
set interfaces ge-1/2/2 unit 120 family bridge interface-mode trunk
set interfaces ge-1/2/2 unit 120 family bridge vlan-id-list 120
set interfaces ge-1/2/2 unit 220 family bridge interface-mode trunk
set interfaces ge-1/2/2 unit 220 family bridge vlan-id-list 220
set interfaces ge-1/2/3 gigether-options 802.3ad ae0
set interfaces ge-1/2/4 gigether-options 802.3ad ae0
set interfaces ge-1/3/0 flexible-vlan-tagging
set interfaces ge-1/3/0 encapsulation flexible-ethernet-services
set interfaces ge-1/3/0 unit 0 encapsulation vlan-bridge
set interfaces ge-1/3/0 unit 0 vlan-id 10
set interfaces ge-1/3/0 unit 100 family bridge interface-mode trunk
set interfaces ge-1/3/0 unit 100 family bridge vlan-id-list 110
set interfaces ge-1/3/0 unit 100 family bridge vlan-id-list 120
set interfaces ge-1/3/0 unit 100 family bridge vlan-id-list 130
set interfaces ge-1/3/0 unit 200 family bridge interface-mode trunk
set interfaces ge-1/3/0 unit 200 family bridge vlan-id-list 210
set interfaces ge-1/3/0 unit 200 family bridge vlan-id-list 220
set interfaces ge-1/3/0 unit 200 family bridge vlan-id-list 230
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 esi 00:11:11:11:11:11:11:11
set interfaces ae0 esi all-active
set interfaces ae0 unit 0 encapsulation vlan-bridge
set interfaces ae0 unit 0 vlan-id 10
set interfaces ae0 unit 110 family bridge interface-mode trunk
set interfaces ae0 unit 110 family bridge vlan-id-list 110
set interfaces ae0 unit 210 family bridge interface-mode trunk
set interfaces ae0 unit 210 family bridge vlan-id-list 210
set interfaces ae12 flexible-vlan-tagging
set interfaces ae12 encapsulation flexible-ethernet-services
set interfaces ae12 aggregated-ether-options minimum-links 1
set interfaces ae12 unit 0 vlan-id 1200
```

```
set interfaces ae12 unit 0 family inet address 198.51.100.12/24
set interfaces ae12 unit 0 family iso
set interfaces ae12 unit 0 family mpls
set interfaces ae13 flexible-vlan-tagging
set interfaces ae13 encapsulation flexible-ethernet-services
set interfaces ae13 aggregated-ether-options minimum-links 1
set interfaces ae13 unit 0 vlan-tags outer 1300
set interfaces ae13 unit 0 vlan-tags inner 13
set interfaces ae13 unit 0 family inet address 198.51.100.13/24
set interfaces ae13 unit 0 family iso
set interfaces ae13 unit 0 family mpls
set interfaces irb unit 0 family inet address 192.0.2.9/24
set interfaces irb unit 0 mac 00:99:99:99:01:99
set interfaces irb unit 1 family inet address 192.0.2.10/24
set interfaces irb unit 1 mac 00:99:99:99:02:99
set interfaces irb unit 2 family inet address 192.0.2.11/24
set interfaces irb unit 2 mac 00:99:99:99:03:99
set interfaces irb unit 10 family inet address 192.0.2.12/24
set interfaces irb unit 10 mac 00:99:99:99:01:90
set interfaces lo0 unit 0 family inet address 198.51.100.1/24 primary
set interfaces lo0 unit 0 family iso
set routing-options router-id 198.51.100.1
set routing-options autonomous-system 65221
set routing-options forwarding-table export load-balancing-policy
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls label-switched-path pe1tope2 from 198.51.100.1
set protocols mpls label-switched-path pe1tope2 to 198.51.100.2
set protocols mpls label-switched-path pe1tope2 primary direct_to_pe2
set protocols mpls label-switched-path pe1tope3 from 198.51.100.1
set protocols mpls label-switched-path pe1tope3 to 198.51.100.3
set protocols mpls label-switched-path pe1tope3 primary direct_to_pe3
set protocols mpls label-switched-path pe1tope4 from 198.51.100.1
set protocols mpls label-switched-path pe1tope4 to 198.51.100.4
set protocols mpls label-switched-path pe1tope4 primary direct_to_pe4
set protocols mpls path pe4_to_pe3 198.51.100.4 strict
set protocols mpls path pe4_to_pe3 198.51.100.3 strict
set protocols mpls path direct_to_pe2 198.51.100.5 strict
set protocols mpls path direct_to_pe3 198.51.100.6 strict
set protocols mpls path direct_to_pe4 198.51.100.9 strict
set protocols mpls path pe2_to_pe3 198.51.100.2 strict
set protocols mpls path pe2_to_pe3 198.51.100.3 strict
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group RR type internal
set protocols bgp group RR local-address 198.51.100.1
set protocols bgp group RR family evpn signaling
set protocols bgp group RR neighbor 203.0.113.0
set protocols isis level 1 disable
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 level 2 metric 0
set protocols ldp deaggregate
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols evpn
```

```

set protocols l2-learning global-mac-table-aging-time 18000
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set routing-instances VS-1 instance-type virtual-switch
set routing-instances VS-1 interface ge-1/2/1.130
set routing-instances VS-1 interface ge-1/2/2.120
set routing-instances VS-1 interface ge-1/3/0.100
set routing-instances VS-1 interface ae0.110
set routing-instances VS-1 route-distinguisher 198.51.100.1:101
set routing-instances VS-1 vrf-target target:100:101
set routing-instances VS-1 protocols evpn extended-vlan-list 110
set routing-instances VS-1 protocols evpn extended-vlan-list 120
set routing-instances VS-1 protocols evpn extended-vlan-list 130
set routing-instances VS-1 protocols evpn default-gateway do-not-advertise
set routing-instances VS-1 bridge-domains bd-110 vlan-id 110
set routing-instances VS-1 bridge-domains bd-110 routing-interface irb.0
set routing-instances VS-1 bridge-domains bd-120 vlan-id 120
set routing-instances VS-1 bridge-domains bd-120 routing-interface irb.1
set routing-instances VS-1 bridge-domains bd-130 vlan-id 130
set routing-instances VS-1 bridge-domains bd-130 routing-interface irb.2
set routing-instances VS-2 instance-type virtual-switch
set routing-instances VS-2 interface ge-1/2/1.230
set routing-instances VS-2 interface ge-1/2/2.220
set routing-instances VS-2 interface ge-1/3/0.200
set routing-instances VS-2 interface ae0.210
set routing-instances VS-2 route-distinguisher 198.51.100.1:201
set routing-instances VS-2 vrf-target target:100:201
set routing-instances VS-2 protocols evpn extended-vlan-list 210
set routing-instances VS-2 protocols evpn extended-vlan-list 220
set routing-instances VS-2 protocols evpn extended-vlan-list 230
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 210
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 220
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 230
set routing-instances mhevnp instance-type evpn
set routing-instances mhevnp vlan-id 10
set routing-instances mhevnp interface ge-1/2/1.0
set routing-instances mhevnp interface ge-1/2/2.0
set routing-instances mhevnp interface ge-1/3/0.0
set routing-instances mhevnp interface ae0.0
set routing-instances mhevnp routing-interface irb.10
set routing-instances mhevnp route-distinguisher 198.51.100.1:1
set routing-instances mhevnp vrf-target target:100:1
set routing-instances mhevnp protocols evpn default-gateway do-not-advertise
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
set routing-instances vrf interface irb.1
set routing-instances vrf interface irb.2
set routing-instances vrf interface irb.10
set routing-instances vrf route-distinguisher 198.51.100.1:11
set routing-instances vrf vrf-target target:100:11
set routing-instances vrf vrf-table-label

```

```

PE2 set chassis aggregated-devices ethernet device-count 20
set chassis network-services enhanced-ip
set interfaces ge-0/0/5 gigether-options 802.3ad ae12
set interfaces ge-0/1/0 gigether-options 802.3ad ae12

```

```
set interfaces ge-0/1/1 flexible-vlan-tagging
set interfaces ge-0/1/1 encapsulation flexible-ethernet-services
set interfaces ge-0/1/1 esi 00:33:33:33:33:33:33:33:33
set interfaces ge-0/1/1 esi all-active
set interfaces ge-0/1/1 unit 0 encapsulation vlan-bridge
set interfaces ge-0/1/1 unit 0 vlan-id 30
set interfaces ge-0/1/1 unit 130 family bridge interface-mode trunk
set interfaces ge-0/1/1 unit 130 family bridge vlan-id-list 130
set interfaces ge-0/1/1 unit 230 family bridge interface-mode trunk
set interfaces ge-0/1/1 unit 230 family bridge vlan-id-list 230
set interfaces ge-0/1/2 flexible-vlan-tagging
set interfaces ge-0/1/2 encapsulation flexible-ethernet-services
set interfaces ge-0/1/2 esi 00:22:22:22:22:22:22:22:22
set interfaces ge-0/1/2 esi all-active
set interfaces ge-0/1/2 unit 0 encapsulation vlan-bridge
set interfaces ge-0/1/2 unit 0 vlan-id 20
set interfaces ge-0/1/2 unit 120 family bridge interface-mode trunk
set interfaces ge-0/1/2 unit 120 family bridge vlan-id-list 120
set interfaces ge-0/1/2 unit 220 family bridge interface-mode trunk
set interfaces ge-0/1/2 unit 220 family bridge vlan-id-list 220
set interfaces ge-0/1/3 unit 0 family inet address 192.0.2.3/24
set interfaces ge-0/1/3 unit 0 family iso
set interfaces ge-0/1/3 unit 0 family mpls
set interfaces ge-0/1/4 gigether-options 802.3ad ae0
set interfaces ge-0/1/6 flexible-vlan-tagging
set interfaces ge-0/1/6 encapsulation flexible-ethernet-services
set interfaces ge-0/1/6 unit 0 encapsulation vlan-bridge
set interfaces ge-0/1/6 unit 0 vlan-id 10
set interfaces ge-0/1/6 unit 100 family bridge interface-mode trunk
set interfaces ge-0/1/6 unit 100 family bridge vlan-id-list 110
set interfaces ge-0/1/6 unit 100 family bridge vlan-id-list 120
set interfaces ge-0/1/6 unit 100 family bridge vlan-id-list 130
set interfaces ge-0/1/6 unit 200 family bridge interface-mode trunk
set interfaces ge-0/1/6 unit 200 family bridge vlan-id-list 210
set interfaces ge-0/1/6 unit 200 family bridge vlan-id-list 220
set interfaces ge-0/1/6 unit 200 family bridge vlan-id-list 230
set interfaces ge-0/1/9 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 esi 00:11:11:11:11:11:11:11:11
set interfaces ae0 esi all-active
set interfaces ae0 unit 0 encapsulation vlan-bridge
set interfaces ae0 unit 0 vlan-id 10
set interfaces ae0 unit 110 family bridge interface-mode trunk
set interfaces ae0 unit 110 family bridge vlan-id-list 110
set interfaces ae0 unit 210 family bridge interface-mode trunk
set interfaces ae0 unit 210 family bridge vlan-id-list 210
set interfaces ae12 flexible-vlan-tagging
set interfaces ae12 encapsulation flexible-ethernet-services
set interfaces ae12 aggregated-ether-options minimum-links 1
set interfaces ae12 unit 0 vlan-id 1200
set interfaces ae12 unit 0 family inet address 198.51.100.5/24
set interfaces ae12 unit 0 family iso
set interfaces ae12 unit 0 family mpls
set interfaces irb unit 0 family inet address 192.0.2.9/24
set interfaces irb unit 0 mac 00:99:99:99:01:99
```

```
set interfaces irb unit 1 family inet address 192.0.2.10/24
set interfaces irb unit 1 mac 00:99:99:99:02:99
set interfaces irb unit 2 family inet address 192.0.2.11/24
set interfaces irb unit 2 mac 00:99:99:99:03:99
set interfaces irb unit 10 family inet address 192.0.2.12/24
set interfaces irb unit 10 mac 00:99:99:99:01:90
set interfaces lo0 unit 0 family inet address 198.51.100.2/32 primary
set interfaces lo0 unit 0 family iso
set routing-options router-id 198.51.100.2
set routing-options autonomous-system 65221
set routing-options forwarding-table export load-balancing-policy
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls label-switched-path pe2tope1 from 198.51.100.2
set protocols mpls label-switched-path pe2tope1 to 198.51.100.1
set protocols mpls label-switched-path pe2tope1 primary direct_to_pe1
set protocols mpls label-switched-path pe2tope3 from 198.51.100.2
set protocols mpls label-switched-path pe2tope3 to 198.51.100.3
set protocols mpls label-switched-path pe2tope3 primary direct_to_pe3
set protocols mpls label-switched-path pe2tope4 from 198.51.100.2
set protocols mpls label-switched-path pe2tope4 to 198.51.100.4
set protocols mpls label-switched-path pe2tope4 primary direct_to_pe4
set protocols mpls path direct_to_pe1 198.51.100.12 strict
set protocols mpls path direct_to_pe3 198.51.100.7 strict
set protocols mpls path direct_to_pe4 198.51.100.8 strict
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group RR type internal
set protocols bgp group RR local-address 198.51.100.2
set protocols bgp group RR family evpn signaling
set protocols bgp group RR neighbor 203.0.113.0
set protocols isis level 1 disable
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 level 2 metric 0
set protocols ldp deaggregate
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols l2-learning global-mac-table-aging-time 18000
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set routing-instances VS-1 instance-type virtual-switch
set routing-instances VS-1 interface ge-0/1/1.130
set routing-instances VS-1 interface ge-0/1/2.120
set routing-instances VS-1 interface ge-0/1/6.100
set routing-instances VS-1 interface ae0.110
set routing-instances VS-1 route-distinguisher 198.51.100.2:101
set routing-instances VS-1 vrf-target target:100:101
set routing-instances VS-1 protocols evpn extended-vlan-list 110
set routing-instances VS-1 protocols evpn extended-vlan-list 120
set routing-instances VS-1 protocols evpn extended-vlan-list 130
set routing-instances VS-1 protocols evpn default-gateway do-not-advertise
set routing-instances VS-1 bridge-domains bd-110 vlan-id 110
set routing-instances VS-1 bridge-domains bd-110 routing-interface irb.0
set routing-instances VS-1 bridge-domains bd-120 vlan-id 120
set routing-instances VS-1 bridge-domains bd-120 routing-interface irb.1
set routing-instances VS-1 bridge-domains bd-130 vlan-id 130
```

```
set routing-instances VS-1 bridge-domains bd-130 routing-interface irb.2
set routing-instances VS-2 instance-type virtual-switch
set routing-instances VS-2 interface ge-0/1/1.230
set routing-instances VS-2 interface ge-0/1/2.220
set routing-instances VS-2 interface ge-0/1/6.200
set routing-instances VS-2 interface ae0.210
set routing-instances VS-2 route-distinguisher 198.51.100.2:201
set routing-instances VS-2 vrf-target target:100:201
set routing-instances VS-2 protocols evpn extended-vlan-list 210
set routing-instances VS-2 protocols evpn extended-vlan-list 220
set routing-instances VS-2 protocols evpn extended-vlan-list 230
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 210
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 220
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 230
set routing-instances mhevnp instance-type evpn
set routing-instances mhevnp vlan-id 10
set routing-instances mhevnp interface ge-0/1/1.0
set routing-instances mhevnp interface ge-0/1/2.0
set routing-instances mhevnp interface ge-0/1/6.0
set routing-instances mhevnp interface ae0.0
set routing-instances mhevnp routing-interface irb.10
set routing-instances mhevnp route-distinguisher 198.51.100.2:1
set routing-instances mhevnp vrf-target target:100:1
set routing-instances mhevnp protocols evpn default-gateway do-not-advertise
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
set routing-instances vrf interface irb.1
set routing-instances vrf interface irb.2
set routing-instances vrf interface irb.10
set routing-instances vrf route-distinguisher 198.51.100.2:11
set routing-instances vrf vrf-target target:100:11
set routing-instances vrf vrf-table-label
```

```
PE3 set chassis aggregated-devices ethernet device-count 20
set chassis network-services enhanced-ip
set interfaces ge-2/0/1 gigether-options 802.3ad ae13
set interfaces ge-2/0/2 gigether-options 802.3ad ae13
set interfaces ge-2/0/4 gigether-options 802.3ad ae0
set interfaces ge-2/0/9 flexible-vlan-tagging
set interfaces ge-2/0/9 encapsulation flexible-ethernet-services
set interfaces ge-2/0/9 unit 0 encapsulation vlan-bridge
set interfaces ge-2/0/9 unit 0 vlan-id 10
set interfaces ge-2/0/9 unit 100 family bridge interface-mode trunk
set interfaces ge-2/0/9 unit 100 family bridge vlan-id-list 110
set interfaces ge-2/0/9 unit 100 family bridge vlan-id-list 120
set interfaces ge-2/0/9 unit 100 family bridge vlan-id-list 130
set interfaces ge-2/0/9 unit 200 family bridge interface-mode trunk
set interfaces ge-2/0/9 unit 200 family bridge vlan-id-list 210
set interfaces ge-2/0/9 unit 200 family bridge vlan-id-list 220
set interfaces ge-2/0/9 unit 200 family bridge vlan-id-list 230
set interfaces ge-2/1/0 unit 0 family inet address 192.0.2.5/24
set interfaces ge-2/1/0 unit 0 family iso
set interfaces ge-2/1/0 unit 0 family mpls
set interfaces ge-2/1/6 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
```

```
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 esi 00:11:11:11:11:11:11:11
set interfaces ae0 esi all-active
set interfaces ae0 unit 0 encapsulation vlan-bridge
set interfaces ae0 unit 0 vlan-id 10
set interfaces ae0 unit 110 family bridge interface-mode trunk
set interfaces ae0 unit 110 family bridge vlan-id-list 110
set interfaces ae0 unit 210 family bridge interface-mode trunk
set interfaces ae0 unit 210 family bridge vlan-id-list 210
set interfaces ae13 flexible-vlan-tagging
set interfaces ae13 encapsulation flexible-ethernet-services
set interfaces ae13 aggregated-ether-options minimum-links 1
set interfaces ae13 unit 0 vlan-tags outer 1300
set interfaces ae13 unit 0 vlan-tags inner 13
set interfaces ae13 unit 0 family inet address 198.51.100.6/24
set interfaces ae13 unit 0 family iso
set interfaces ae13 unit 0 family mpls
set interfaces irb unit 0 family inet address 192.0.2.9/24
set interfaces irb unit 0 mac 00:99:99:99:01:99
set interfaces irb unit 1 family inet address 192.0.2.10/24
set interfaces irb unit 1 mac 00:99:99:99:02:99
set interfaces irb unit 2 family inet address 192.0.2.11/24
set interfaces irb unit 2 mac 00:99:99:99:03:99
set interfaces irb unit 10 family inet address 192.0.2.12/24
set interfaces irb unit 10 mac 00:99:99:99:01:90
set interfaces lo0 unit 0 family inet address 198.51.100.3/32 primary
set interfaces lo0 unit 0 family iso
set routing-options router-id 198.51.100.3
set routing-options autonomous-system 65221
set routing-options forwarding-table export load-balancing-policy
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls label-switched-path pe3tope1 from 198.51.100.3
set protocols mpls label-switched-path pe3tope1 to 198.51.100.1
set protocols mpls label-switched-path pe3tope1 primary direct_to_pe1
set protocols mpls label-switched-path pe3tope2 from 198.51.100.3
set protocols mpls label-switched-path pe3tope2 to 198.51.100.2
set protocols mpls label-switched-path pe3tope2 primary direct_to_pe2
set protocols mpls label-switched-path pe3tope4 from 198.51.100.3
set protocols mpls label-switched-path pe3tope4 to 198.51.100.4
set protocols mpls label-switched-path pe3tope4 primary direct_to_pe4
set protocols mpls path direct_to_pe1 198.51.100.13 strict
set protocols mpls path direct_to_pe2 198.51.100.10 strict
set protocols mpls path direct_to_pe4 198.51.100.11 strict
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group RR type internal
set protocols bgp group RR local-address 198.51.100.3
set protocols bgp group RR family evpn signaling
set protocols bgp group RR neighbor 203.0.113.0
set protocols isis level 1 disable
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 level 2 metric 0
set protocols ldp deaggregate
set protocols ldp interface all
```

```
set protocols ldp interface fxp0.0 disable
set protocols l2-learning global-mac-table-aging-time 18000
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set routing-instances VS-1 instance-type virtual-switch
set routing-instances VS-1 interface ge-2/0/0.100
set routing-instances VS-1 interface ge-2/0/9.100
set routing-instances VS-1 interface ae0.110
set routing-instances VS-1 route-distinguisher 198.51.100.3:101
set routing-instances VS-1 vrf-target target:100:101
set routing-instances VS-1 protocols evpn extended-vlan-list 110
set routing-instances VS-1 protocols evpn extended-vlan-list 120
set routing-instances VS-1 protocols evpn extended-vlan-list 130
set routing-instances VS-1 protocols evpn default-gateway do-not-advertise
set routing-instances VS-1 bridge-domains bd-110 vlan-id 110
set routing-instances VS-1 bridge-domains bd-110 routing-interface irb.0
set routing-instances VS-1 bridge-domains bd-120 vlan-id 120
set routing-instances VS-1 bridge-domains bd-120 routing-interface irb.1
set routing-instances VS-1 bridge-domains bd-130 vlan-id 130
set routing-instances VS-1 bridge-domains bd-130 routing-interface irb.2
set routing-instances VS-2 instance-type virtual-switch
set routing-instances VS-2 interface ge-2/0/0.200
set routing-instances VS-2 interface ge-2/0/9.200
set routing-instances VS-2 interface ae0.210
set routing-instances VS-2 route-distinguisher 198.51.100.3:201
set routing-instances VS-2 vrf-target target:100:201
set routing-instances VS-2 protocols evpn extended-vlan-list 210
set routing-instances VS-2 protocols evpn extended-vlan-list 220
set routing-instances VS-2 protocols evpn extended-vlan-list 230
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 210
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 220
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 230
set routing-instances mhevnp instance-type evpn
set routing-instances mhevnp vlan-id 10
set routing-instances mhevnp interface ge-2/0/0.0
set routing-instances mhevnp interface ge-2/0/9.0
set routing-instances mhevnp interface ae0.0
set routing-instances mhevnp routing-interface irb.10
set routing-instances mhevnp route-distinguisher 198.51.100.3:1
set routing-instances mhevnp vrf-target target:100:1
set routing-instances mhevnp protocols evpn default-gateway do-not-advertise
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
set routing-instances vrf interface irb.1
set routing-instances vrf interface irb.2
set routing-instances vrf interface irb.10
set routing-instances vrf route-distinguisher 198.51.100.3:11
set routing-instances vrf vrf-target target:100:11
set routing-instances vrf vrf-table-label
```

```
PE4 set chassis aggregated-devices ethernet device-count 20
set chassis network-services enhanced-ip
set interfaces ge-1/0/3 unit 0 family inet address 192.0.2.7/24
set interfaces ge-1/0/3 unit 0 family iso
set interfaces ge-1/0/3 unit 0 family mpls
set interfaces ge-1/1/1 flexible-vlan-tagging
```



```
set interfaces ge-1/1/1 encapsulation flexible-ethernet-services
set interfaces ge-1/1/1 esi 00:44:44:44:44:44:44:44:44
set interfaces ge-1/1/1 esi all-active
set interfaces ge-1/1/1 unit 0 encapsulation vlan-bridge
set interfaces ge-1/1/1 unit 0 vlan-id 10
set interfaces ge-1/1/1 unit 100 family bridge interface-mode trunk
set interfaces ge-1/1/1 unit 100 family bridge vlan-id-list 110
set interfaces ge-1/1/1 unit 100 family bridge vlan-id-list 120
set interfaces ge-1/1/1 unit 100 family bridge vlan-id-list 130
set interfaces ge-1/1/1 unit 200 family bridge interface-mode trunk
set interfaces ge-1/1/1 unit 200 family bridge vlan-id-list 210
set interfaces ge-1/1/1 unit 200 family bridge vlan-id-list 220
set interfaces ge-1/1/1 unit 200 family bridge vlan-id-list 230
set interfaces ge-1/1/9 flexible-vlan-tagging
set interfaces ge-1/1/9 encapsulation flexible-ethernet-services
set interfaces ge-1/1/9 unit 0 encapsulation vlan-bridge
set interfaces ge-1/1/9 unit 0 vlan-id 10
set interfaces ge-1/1/9 unit 100 family bridge interface-mode trunk
set interfaces ge-1/1/9 unit 100 family bridge vlan-id-list 110
set interfaces ge-1/1/9 unit 100 family bridge vlan-id-list 120
set interfaces ge-1/1/9 unit 100 family bridge vlan-id-list 130
set interfaces ge-1/1/9 unit 200 family bridge interface-mode trunk
set interfaces ge-1/1/9 unit 200 family bridge vlan-id-list 210
set interfaces ge-1/1/9 unit 200 family bridge vlan-id-list 220
set interfaces ge-1/1/9 unit 200 family bridge vlan-id-list 230
set interfaces irb unit 0 family inet address 192.0.2.9/24
set interfaces irb unit 0 mac 00:99:99:99:01:99
set interfaces irb unit 1 family inet address 192.0.2.10/24
set interfaces irb unit 1 mac 00:99:99:99:02:99
set interfaces irb unit 2 family inet address 192.0.2.11/24
set interfaces irb unit 2 mac 00:99:99:99:03:99
set interfaces irb unit 10 family inet address 192.0.2.12/24
set interfaces irb unit 10 mac 00:99:99:99:01:90
set interfaces lo0 unit 0 family inet address 198.51.100.4/32 primary
set routing-options router-id 198.51.100.4
set routing-options autonomous-system 65221
set routing-options forwarding-table export load-balancing-policy
set protocols rsvp interface all
set protocols rsvp interface fxp0.0 disable
set protocols mpls label-switched-path pe4tope1 from 198.51.100.4
set protocols mpls label-switched-path pe4tope1 to 198.51.100.1
set protocols mpls label-switched-path pe4tope1 primary direct_to_pe1
set protocols mpls label-switched-path pe4tope2 from 198.51.100.4
set protocols mpls label-switched-path pe4tope2 to 198.51.100.2
set protocols mpls label-switched-path pe4tope2 primary direct_to_pe2
set protocols mpls label-switched-path pe4tope3 from 198.51.100.4
set protocols mpls label-switched-path pe4tope3 to 198.51.100.3
set protocols mpls label-switched-path pe4tope3 primary direct_to_pe3
set protocols mpls path pe2_to_pe3 198.51.100.2 strict
set protocols mpls path pe2_to_pe3 198.51.100.3 strict
set protocols mpls path direct_to_pe1 198.51.100.14 strict
set protocols mpls path direct_to_pe2 198.51.100.15 strict
set protocols mpls path direct_to_pe3 198.51.100.16 strict
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group RR type internal
```

```
set protocols bgp group RR local-address 198.51.100.4
set protocols bgp group RR family evpn signaling
set protocols bgp group RR neighbor 203.0.113.0
set protocols isis level 1 disable
set protocols isis interface all level 2 metric 10
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0 level 2 metric 0
set protocols ldp deaggregate
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols l2-learning global-mac-table-aging-time 18000
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set routing-instances VS-1 instance-type virtual-switch
set routing-instances VS-1 interface ge-1/1/1.100
set routing-instances VS-1 interface ge-1/1/9.100
set routing-instances VS-1 route-distinguisher 198.51.100.4:101
set routing-instances VS-1 vrf-target target:100:101
set routing-instances VS-1 protocols evpn extended-vlan-list 110
set routing-instances VS-1 protocols evpn extended-vlan-list 120
set routing-instances VS-1 protocols evpn extended-vlan-list 130
set routing-instances VS-1 protocols evpn default-gateway do-not-advertise
set routing-instances VS-1 bridge-domains bd-110 vlan-id 110
set routing-instances VS-1 bridge-domains bd-110 routing-interface irb.0
set routing-instances VS-1 bridge-domains bd-120 vlan-id 120
set routing-instances VS-1 bridge-domains bd-120 routing-interface irb.1
set routing-instances VS-1 bridge-domains bd-130 vlan-id 130
set routing-instances VS-1 bridge-domains bd-130 routing-interface irb.2
set routing-instances VS-2 instance-type virtual-switch
set routing-instances VS-2 interface ge-1/1/1.200
set routing-instances VS-2 interface ge-1/1/9.200
set routing-instances VS-2 route-distinguisher 198.51.100.4:201
set routing-instances VS-2 vrf-target target:100:201
set routing-instances VS-2 protocols evpn extended-vlan-list 210
set routing-instances VS-2 protocols evpn extended-vlan-list 220
set routing-instances VS-2 protocols evpn extended-vlan-list 230
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 210
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 220
set routing-instances VS-2 bridge-domains bd-a vlan-id-list 230
set routing-instances mhevpn instance-type evpn
set routing-instances mhevpn vlan-id 10
set routing-instances mhevpn interface ge-1/1/1.0
set routing-instances mhevpn interface ge-1/1/9.0
set routing-instances mhevpn routing-interface irb.10
set routing-instances mhevpn route-distinguisher 198.51.100.4:1
set routing-instances mhevpn vrf-target target:100:1
set routing-instances mhevpn protocols evpn default-gateway do-not-advertise
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.0
set routing-instances vrf interface irb.1
set routing-instances vrf interface irb.2
set routing-instances vrf interface irb.10
set routing-instances vrf route-distinguisher 198.51.100.4:11
set routing-instances vrf vrf-target target:100:11
set routing-instances vrf vrf-table-label
```

```

P (RR)  set interfaces ge-1/0/5 unit 0 family inet address 192.0.2.8/24
        set interfaces ge-1/1/2 unit 0 family inet address 192.0.2.6/24
        set interfaces ge-1/1/3 unit 0 family inet address 192.0.2.4/24
        set interfaces ge-1/1/4 unit 0 family inet address 192.0.2.2/24
        set interfaces lo0 unit 0 family inet address 203.0.113.0
        set protocols bgp group RR type internal
        set protocols bgp group RR local-address 203.0.113.0
        set protocols bgp group RR family evpn signaling
        set protocols bgp group RR cluster 1.2.3.4
        set protocols bgp group RR neighbor 198.51.100.1
        set protocols bgp group RR neighbor 198.51.100.2
        set protocols bgp group RR neighbor 198.51.100.3
        set protocols bgp group RR neighbor 198.51.100.4
        set protocols isis interface all level 1 disable
        set protocols ldp interface all
        set routing-options router-id 203.0.113.0
        set routing-options autonomous-system 65221

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure Router PE1:



NOTE: Repeat this procedure for all other multihomed PE routers after modifying the appropriate interface names, addresses, and other parameters.

1. Configure the MX Series router to operate in the enhanced-ip mode because the EVPN active-active functionality is supported on routers with MPCs and MIC interfaces only.

A system reboot is required on committing this configuration.

```

[edit chassis]
user@PE1# set network-services enhanced-ip

```

2. Specify the number of aggregated Ethernet interfaces to be created.

```

[edit chassis]
user@PE1# set aggregated-devices ethernet device-count 20

```

3. Configure Router PE1 interfaces within the ae0 aggregated bundle toward the multihomed customer site, Router CE10.

- a. Assign interfaces ge-1/2/3 and ge-1/2/4 within the ae0 aggregated bundle.

```

[edit interfaces]
user@PE1# set ge-1/2/3 gigether-options 802.3ad ae0
user@PE1# set ge-1/2/4 gigether-options 802.3ad ae0

```

- b. Configure the ae0 aggregated bundle parameters for VLAN tagging and encapsulation.

```
[edit interfaces]
user@PE1# set ae0 flexible-vlan-tagging
user@PE1# set ae0 encapsulation flexible-ethernet-services
```

- c. Assign an ESI value for the first Ethernet segment and enable EVPN active-active multihoming for the ae0 aggregated bundle.

```
[edit interfaces]
user@PE1# set ae0 esi 00:11:11:11:11:11:11:11
user@PE1# set ae0 esi all-active
```

- d. Configure a trunk interface on the bridge network for the ae0 aggregated bundle.

```
[edit interfaces]
user@PE1# set ae0 unit 0 encapsulation vlan-bridge
user@PE1# set ae0 unit 0 vlan-id 10
user@PE1# set ae0 unit 110 family bridge interface-mode trunk
user@PE1# set ae0 unit 110 family bridge vlan-id-list 110
user@PE1# set ae0 unit 210 family bridge interface-mode trunk
user@PE1# set ae0 unit 210 family bridge vlan-id-list 210
```

4. Configure the other Router PE1 interfaces toward the multihomed customer site, Router CE10.

- a. Configure the VLAN tagging and encapsulation parameters for the ge-1/2/2 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/2/2 flexible-vlan-tagging
user@PE1# set ge-1/2/2 encapsulation flexible-ethernet-services
```

- b. Assign an ESI value for the second Ethernet segment and enable EVPN active-active multihoming for the ge-1/2/2 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/2/2 esi 00:22:22:22:22:22:22:22
user@PE1# set ge-1/2/2 esi all-active
```

- c. Configure a trunk interface on the bridge network for the ge-1/2/2 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/2/2 unit 0 encapsulation vlan-bridge
user@PE1# set ge-1/2/2 unit 0 vlan-id 20
user@PE1# set ge-1/2/2 unit 120 family bridge interface-mode trunk
user@PE1# set ge-1/2/2 unit 120 family bridge vlan-id-list 120
user@PE1# set ge-1/2/2 unit 220 family bridge interface-mode trunk
user@PE1# set ge-1/2/2 unit 220 family bridge vlan-id-list 220
```

- d. Configure the VLAN tagging and encapsulation parameters for the ge-1/2/1 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/2/1 flexible-vlan-tagging
user@PE1# set ge-1/2/1 encapsulation flexible-ethernet-services
```

- e. Assign an ESI value for the third Ethernet segment and enable EVPN active-active multihoming for the ge-1/2/1 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/2/1 esi 00:33:33:33:33:33:33:33
user@PE1# set ge-1/2/1 esi all-active
```

- f. Configure a trunk interface on the bridge network for the ge-1/2/1 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/2/1 unit 0 encapsulation vlan-bridge
user@PE1# set ge-1/2/1 unit 0 vlan-id 30
user@PE1# set ge-1/2/1 unit 130 family bridge interface-mode trunk
user@PE1# set ge-1/2/1 unit 130 family bridge vlan-id-list 130
user@PE1# set ge-1/2/1 unit 230 family bridge interface-mode trunk
user@PE1# set ge-1/2/1 unit 230 family bridge vlan-id-list 230
```

5. Configure Router PE1 interfaces toward Router PE2.

- a. Assign the interfaces ge-1/0/5 and ge-1/1/6 within the ae12 aggregated bundle.

```
[edit interfaces]
user@PE1# set ge-1/0/5 gigether-options 802.3ad ae12
user@PE1# set ge-1/1/6 gigether-options 802.3ad ae12
```

- b. Specify the minimum number of links for the ae12 aggregated bundle to be labeled “up”.

```
[edit interfaces]
user@PE1# set ae12 aggregated-ether-options minimum-links 1
```

- c. Assign an IP address for the ae12 aggregated bundle and enable MPLS and IS-IS protocol families on the bundle.

```
[edit interfaces]
user@PE1# set ae12 unit 0 family inet address 198.51.100.12/24
user@PE1# set ae12 unit 0 family iso
user@PE1# set ae12 unit 0 family mpls
```

- d. Configure the VLAN tagging and encapsulation parameters for the ae12 aggregated bundle and assign VLAN ID 1200 for the bundle.

```
[edit interfaces]
user@PE1# set ae12 flexible-vlan-tagging
user@PE1# set ae12 encapsulation flexible-ethernet-services
user@PE1# set ae12 unit 0 vlan-id 1200
```

6. Configure Router PE1 interfaces toward Router PE3.

- a. Assign the interfaces ge-1/0/3 and ge-1/0/4 within the ae13 aggregated bundle.

```
[edit interfaces]
user@PE1# set ge-1/0/3 gigether-options 802.3ad ae13
user@PE1# set ge-1/0/4 gigether-options 802.3ad ae13
```

- b. Specify the minimum number of links for the ae13 aggregated bundle to be labeled “up”.

```
[edit interfaces]
user@PE1# set ae13 aggregated-ether-options minimum-links 1
```

- c. Assign an IP address for the ae13 aggregated bundle and enable MPLS and IS-IS protocol families on the bundle.

```
[edit interfaces]
user@PE1# set ae13 unit 0 family inet address 198.51.100.13/24
user@PE1# set ae13 unit 0 family iso
user@PE1# set ae13 unit 0 family mpls
```

- d. Configure the VLAN tagging and encapsulation parameters for the ae12 aggregated bundle and assign the inner and outer VLAN tags for the bundle.

```
[edit interfaces]
user@PE1# set ae13 flexible-vlan-tagging
user@PE1# set ae13 encapsulation flexible-ethernet-services
user@PE1# set ae13 unit 0 vlan-tags outer 1300
user@PE1# set ae13 unit 0 vlan-tags inner 13
```

7. Configure the Router PE1 interface toward the single-homed customer site, Router CE1.

- a. Configure the VLAN tagging and encapsulation parameters for the ge-1/3/0 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/3/0 flexible-vlan-tagging
user@PE1# set ge-1/3/0 encapsulation flexible-ethernet-services
```

- b. Configure a trunk interface on the bridge network for the ge-1/3/0 PE1 interface.

```
[edit interfaces]
user@PE1# set ge-1/3/0 unit 0 encapsulation vlan-bridge
user@PE1# set ge-1/3/0 unit 0 vlan-id 10
user@PE1# set ge-1/3/0 unit 100 family bridge interface-mode trunk
user@PE1# set ge-1/3/0 unit 100 family bridge vlan-id-list 110
user@PE1# set ge-1/3/0 unit 100 family bridge vlan-id-list 120
user@PE1# set ge-1/3/0 unit 100 family bridge vlan-id-list 130
user@PE1# set ge-1/3/0 unit 200 family bridge interface-mode trunk
user@PE1# set ge-1/3/0 unit 200 family bridge vlan-id-list 210
user@PE1# set ge-1/3/0 unit 200 family bridge vlan-id-list 220
user@PE1# set ge-1/3/0 unit 200 family bridge vlan-id-list 230
```

8. Configure the Router PE1 interface toward Router P (RR) and enable the MPLS and IS-IS protocol families for the interface.

```
[edit interfaces]
user@PE1# set ge-1/1/4 unit 0 family inet address 192.0.2.1/24
user@PE1# set ge-1/1/4 unit 0 family iso
user@PE1# set ge-1/1/4 unit 0 family mpls
```

9. Configure an IRB interface for Router PE1.

```
[edit interfaces]
user@PE1# set irb unit 0 family inet address 192.0.2.9/24
user@PE1# set irb unit 0 mac 00:99:99:99:01:99
user@PE1# set irb unit 1 family inet address 192.0.2.10/24
user@PE1# set irb unit 1 mac 00:99:99:99:02:99
user@PE1# set irb unit 2 family inet address 192.0.2.11/24
user@PE1# set irb unit 2 mac 00:99:99:99:03:99
user@PE1# set irb unit 10 family inet address 192.0.2.12/24
user@PE1# set irb unit 10 mac 00:99:99:99:01:90
```

10. Configure the loopback interface for Router PE1.

```
[edit interfaces]
user@PE1# set lo0 unit 0 family inet address 198.51.100.1/24 primary
user@PE1# set lo0 unit 0 family iso
```

11. Assign a router ID and the autonomous system number for Router PE1.

```
[edit routing-options]  
user@PE1# set router-id 198.51.100.1  
user@PE1# set autonomous-system 65221
```
12. Assign a load-balancing policy to the forwarding table of Router PE1.

```
[edit routing-options]  
user@PE1# set forwarding-table export load-balancing-policy
```
13. Configure IS-IS on Router PE1.

```
[edit protocols]  
user@PE1# set isis level 1 disable  
user@PE1# set isis interface all level 2 metric 10  
user@PE1# set isis interface fxp0.0 disable  
user@PE1# set isis interface lo0.0 level 2 metric 0
```
14. Configure an internal BGP group for Router PE1 to peer with route reflector, Router P.

```
[edit protocols]  
user@PE1# set bgp group RR type internal  
user@PE1# set bgp group RR local-address 198.51.100.1  
user@PE1# set bgp group RR neighbor 203.0.113.0
```
15. Enable EVPN signaling for the RR BGP group on Router PE1.

```
[edit protocols]  
user@PE1# set bgp group RR family evpn signaling
```
16. Configure RSVP, LDP, MPLS, EVPN, and L2 learning on Router PE1.

```
[edit protocols]  
user@PE1# set rsvp interface all  
user@PE1# set rsvp interface fxp0.0 disable  
  
user@PE1# set ldp deaggregate  
user@PE1# set ldp interface all  
user@PE1# set ldp interface fxp0.0 disable  
  
user@PE1# set mpls interface all  
user@PE1# set mpls interface fxp0.0 disable  
  
user@PE1# set evpn  
  
user@PE1# set l2-learning global-mac-table-aging-time 18000
```
17. Configure label-switched paths between the PE routers.

```
[edit protocols]  
user@PE1# set mpls label-switched-path pe1tope2 from 198.51.100.1
```

```
user@PE1# set mpls label-switched-path pe1tope2 to 198.51.100.2
user@PE1# set mpls label-switched-path pe1tope2 primary direct_to_pe2
```

```
user@PE1# set mpls label-switched-path pe1tope3 from 198.51.100.1
user@PE1# set mpls label-switched-path pe1tope3 to 198.51.100.3
user@PE1# set mpls label-switched-path pe1tope3 primary direct_to_pe3
```

```
user@PE1# set mpls label-switched-path pe1tope4 from 198.51.100.1
user@PE1# set mpls label-switched-path pe1tope4 to 198.51.100.4
user@PE1# set mpls label-switched-path pe1tope4 primary direct_to_pe4
```

18. Configure MPLS paths from Router PE1 to other PE routers.

```
user@PE1# set mpls path pe4_to_pe3 198.51.100.4 strict
user@PE1# set mpls path pe4_to_pe3 198.51.100.3 strict
```

```
user@PE1# set mpls path direct_to_pe2 198.51.100.5 strict
user@PE1# set mpls path direct_to_pe3 198.51.100.6 strict
```

```
user@PE1# set mpls path direct_to_pe4 198.51.100.9 strict
```

```
user@PE1# set mpls path pe2_to_pe3 198.51.100.2 strict
user@PE1# set mpls path pe2_to_pe3 198.51.100.3 strict
```

19. Configure the load-balancing policy to enable load balancing per packet.

```
[edit policy-options]
user@PE1# set policy-statement load-balancing-policy then load-balance per-packet
```

20. Configure the first virtual switch routing instance.

- a. Configure the routing-instance type and assign Router PE1 interfaces to the routing instance.

```
[edit routing-instances]
user@PE1# set VS-1 instance-type virtual-switch
user@PE1# set VS-1 interface ge-1/2/1.130
user@PE1# set VS-1 interface ge-1/2/2.120
user@PE1# set VS-1 interface ge-1/3/0.100
user@PE1# set VS-1 interface ae0.110
```

- b. Configure the route distinguisher and the VPN routing and forwarding (VRF) target for the VS-1 routing instance.

```
[edit routing-instances]
user@PE1# set VS-1 route-distinguisher 198.51.100.1:101
user@PE1# set VS-1 vrf-target target:100:101
```

- c. Configure EVPN and assign VLANs to the VS-1 routing instance.

```
[edit routing-instances]
user@PE1# set VS-1 protocols evpn extended-vlan-list 110
user@PE1# set VS-1 protocols evpn extended-vlan-list 120
user@PE1# set VS-1 protocols evpn extended-vlan-list 130
user@PE1# set VS-1 protocols evpn default-gateway do-not-advertise
```


- d. Configure the bridge domains and their associated VLANs and IRB interfaces for the VS-1 routing instance.

```
[edit routing-instances]
user@PE1# set VS-1 bridge-domains bd-110 vlan-id 110
user@PE1# set VS-1 bridge-domains bd-110 routing-interface irb.0
user@PE1# set VS-1 bridge-domains bd-120 vlan-id 120
user@PE1# set VS-1 bridge-domains bd-120 routing-interface irb.1
user@PE1# set VS-1 bridge-domains bd-130 vlan-id 130
user@PE1# set VS-1 bridge-domains bd-130 routing-interface irb.2
```

21. Configure the second virtual switch routing instance.

- a. Configure the routing-instance type and assign Router PE1 interfaces to the routing instance.

```
[edit routing-instances]
user@PE1# set VS-2 instance-type virtual-switch
user@PE1# set VS-2 interface ge-1/2/1.230
user@PE1# set VS-2 interface ge-1/2/2.220
user@PE1# set VS-2 interface ge-1/3/0.200
user@PE1# set VS-2 interface ae0.210
```

- b. Configure the route distinguisher and the VPN routing and forwarding (VRF) target for the VS-2 routing instance.

```
[edit routing-instances]
user@PE1# set VS-2 route-distinguisher 198.51.100.1:201
user@PE1# set VS-2 vrf-target target:100:201
```

- c. Configure EVPN and assign VLANs to the VS-2 routing instance.

```
[edit routing-instances]
user@PE1# set VS-2 protocols evpn extended-vlan-list 210
user@PE1# set VS-2 protocols evpn extended-vlan-list 220
user@PE1# set VS-2 protocols evpn extended-vlan-list 230
```

- d. Configure the bridge domains and their associated VLANs for the VS-2 routing instance.

```
[edit routing-instances]
user@PE1# set routing-instances VS-2 bridge-domains bd-a vlan-id-list 210
user@PE1# set routing-instances VS-2 bridge-domains bd-a vlan-id-list 220
user@PE1# set routing-instances VS-2 bridge-domains bd-a vlan-id-list 230
```

22. Configure the multihomed EVPN routing instance.

- a. Configure the routing-instance type and assign VLANs and Router PE1 interfaces to the routing instance.

```
[edit routing-instances]
user@PE1# set mhevpn instance-type evpn
user@PE1# set mhevpn vlan-id 10
user@PE1# set mhevpn interface ge-1/2/1.0
user@PE1# set mhevpn interface ge-1/2/2.0
user@PE1# set mhevpn interface ge-1/3/0.0
user@PE1# set mhevpn interface ae0.0
user@PE1# set mhevpn routing-interface irb.10
```

- b. Configure the route distinguisher and the VPN routing and forwarding (VRF) target for the mhevpn routing instance.

```
[edit routing-instances]
user@PE1# set mhevpn route-distinguisher 198.51.100.1:1
user@PE1# set mhevpn vrf-target target:100:1
```

- c. Configure EVPN to the mhevpn routing instance.

```
[edit routing-instances]
user@PE1# set routing-instances mhevpn protocols evpn default-gateway
do-not-advertise
```

23. Configure the default routing instance.

- a. Configure the routing-instance type and assign IRB interfaces to the routing instance.

```
[edit routing-instances]
user@PE1# set vrf instance-type vrf
user@PE1# set vrf interface irb.0
user@PE1# set vrf interface irb.1
user@PE1# set vrf interface irb.2
user@PE1# set vrf interface irb.10
```

- b. Configure the route distinguisher and the VPN routing and forwarding (VRF) target for the vrf routing instance.

```
[edit routing-instances]
user@PE1# set vrf route-distinguisher 198.51.100.1:11
user@PE1# set vrf vrf-target target:100:11
user@PE1# set vrf vrf-table-label
```

Configuration on EX9200 Switches

Step-by-Step Procedure

Several configuration statements used to configure active-active mode differ on EX9200 switches from those used on MX Series routers. This procedure shows which configuration statements are specific to EX9200 switches. All other configuration in this example applies both to EX9200 switches and MX Series routers.

1. To configure a trunk interface, include the **family ethernet-switching** statements instead of the **family bridge** statements in all occurrences.

```
[edit interfaces]
user@PE#1# set interfaces ge-1/2/1 unit 130 family ethernet-switching
interface-mode trunk
```

2. To configure the Layer 2 Ethernet switching domain, include the **vlan members (vlan-id | name)** statement instead of the **vlan-id-list vlan-id** statement in all occurrences.

```
[edit interfaces]
user@PE#1# set interfaces ge-1/2/1 unit 130 family ethernet-switching vlan members
130
```

3. To configure the VLAN domain and associated VLANs for each routing instance, include the **vlan name** statement, instead of the **bridge-domains** statement in all occurrences.

```
[edit]
user@PE#1# set routing-instances VS-1 vlans bd-110 vlan-id 110
```

4. To configure the IRB interface in each routing instance, include the **l3-interface irb-interface-name** statement instead of the **routing-interface** statement in all occurrences.

```
[edit]
user@PE#1# set routing-instances VS-1 vlans bd-110 l3-interface irb.0
```

Results

From configuration mode, confirm your configuration by entering the **show chassis**, **show interfaces**, **show routing-options**, **show protocols**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1 show chassis
aggregated-devices {
  ethernet {
    device-count 20;
  }
}
network-services enhanced-ip;

user@PE1 show interfaces
ge-1/0/3 {
  gigether-options {
    802.3ad ae13;
  }
}
ge-1/0/4 {
  gigether-options {
    802.3ad ae13;
  }
}
ge-1/0/5 {
  gigether-options {
    802.3ad ae12;
  }
}
ge-1/1/4 {
  unit 0 {
    family inet {
      address 192.0.2.1/24;
    }
    family iso;
    family mpls;
  }
}
```

```
ge-1/1/6 {
  gigaether-options {
    802.3ad ae12;
  }
}
ge-1/2/1 {
  flexible-vlan-tagging;
  encapsulation flexible-ethernet-services;
  esi {
    00:33:33:33:33:33:33:33:33:33;
    all-active;
  }
  unit 0 {
    encapsulation vlan-bridge;
    vlan-id 30;
  }
  unit 130 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 130;
    }
  }
  unit 230 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 230;
    }
  }
}
ge-1/2/2 {
  flexible-vlan-tagging;
  encapsulation flexible-ethernet-services;
  esi {
    00:22:22:22:22:22:22:22:22:22;
    all-active;
  }
  unit 0 {
    encapsulation vlan-bridge;
    vlan-id 20;
  }
  unit 120 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 120;
    }
  }
  unit 220 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 220;
    }
  }
}
ge-1/2/3 {
  gigaether-options {
    802.3ad ae0;
```

```

    }
  }
  ge-1/2/4 {
    gigether-options {
      802.3ad ae0;
    }
  }
  ge-1/3/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
      encapsulation vlan-bridge;
      vlan-id 10;
    }
    unit 100 {
      family bridge {
        interface-mode trunk;
        vlan-id-list [ 110 120 130 ];
      }
    }
    unit 200 {
      family bridge {
        interface-mode trunk;
        vlan-id-list [ 210 220 230 ];
      }
    }
  }
}
ae0 {
  flexible-vlan-tagging;
  encapsulation flexible-ethernet-services;
  esi {
    00:11:11:11:11:11:11:11;
    all-active;
  }
  unit 0 {
    encapsulation vlan-bridge;
    vlan-id 10;
  }
  unit 110 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 110;
    }
  }
  unit 210 {
    family bridge {
      interface-mode trunk;
      vlan-id-list 210;
    }
  }
}
ae12 {
  flexible-vlan-tagging;
  encapsulation flexible-ethernet-services;
  aggregated-ether-options {
    minimum-links 1;
  }
}

```

```
    }
    unit 0 {
        vlan-id 1200;
        family inet {
            address 198.51.100.12/24;
        }
        family iso;
        family mpls;
    }
}
ae13 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    aggregated-ether-options {
        minimum-links 1;
    }
    unit 0 {
        vlan-tags outer 1300 inner 13;
        family inet {
            address 198.51.100.13/24;
        }
        family iso;
        family mpls;
    }
}
irb {
    unit 0 {
        family inet {
            address 192.0.2.9/24;
        }
        mac 00:99:99:99:01:99;
    }
    unit 1 {
        family inet {
            address 192.0.2.10/24;
        }
        mac 00:99:99:99:02:99;
    }
    unit 2 {
        family inet {
            address 192.0.2.11/24;
        }
        mac 00:99:99:99:03:99;
    }
    unit 10 {
        family inet {
            address 192.0.2.12/24;
        }
        mac 00:99:99:99:01:90;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 198.51.100.1/24 {
                primary;
            }
        }
    }
}
```

```

    }
  }
  family iso;
}
}

user@PE1# show routing-options
router-id 198.51.100.1;
autonomous-system 65221;
forwarding-table {
  export load-balancing-policy;
}

user@PE1# show protocols
rsdp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
mpls {
  label-switched-path pe1tope2 {
    from 198.51.100.1;
    to 198.51.100.2;
    primary direct_to_pe2;
  }
  label-switched-path pe1tope3 {
    from 198.51.100.1;
    to 198.51.100.3;
    primary direct_to_pe3;
  }
  label-switched-path pe1tope4 {
    from 198.51.100.1;
    to 198.51.100.4;
    primary direct_to_pe4;
  }
  path pe4_to_pe3 {
    198.51.100.4 strict;
    198.51.100.3 strict;
  }
  path direct_to_pe2 {
    198.51.100.5 strict;
  }
  path direct_to_pe3 {
    198.51.100.6 strict;
  }
  path direct_to_pe4 {
    198.51.100.9 strict;
  }
  path pe2_to_pe3 {
    198.51.100.2 strict;
    198.51.100.3 strict;
  }
  interface all;
  interface fxp0.0 {
    disable;
  }
}

```

```
}
bgp {
  group RR {
    type internal;
    local-address 198.51.100.1;
    family evpn {
      signaling;
    }
    neighbor 203.0.113.0;
  }
}
isis {
  level 1 disable;
  interface all {
    level 2 metric 10;
  }
  interface fxp0.0 {
    disable;
  }
  interface lo0.0 {
    level 2 metric 0;
  }
}
ldp {
  deaggregate;
  interface all;
  interface fxp0.0 {
    disable;
  }
}
evpn {
}
l2-learning {
  global-mac-table-aging-time 18000;
}

user@PE1# show policy-options
policy-statement load-balancing-policy {
  then {
    load-balance per-packet;
  }
}

user@PE1# show routing-instances
VS-1 {
  instance-type virtual-switch;
  interface ge-1/2/1.130;
  interface ge-1/2/2.120;
  interface ge-1/3/0.100;
  interface ae0.110;
  route-distinguisher 198.51.100.1:101;
  vrf-target target:100:101;
  protocols {
    evpn {
      extended-vlan-list [ 110 120 130 ];
      default-gateway do-not-advertise;
    }
  }
}
```



```

}
bridge-domains {
  bd-110 {
    vlan-id 110;
    routing-interface irb.0;
  }
  bd-120 {
    vlan-id 120;
    routing-interface irb.1;
  }
  bd-130 {
    vlan-id 130;
    routing-interface irb.2;
  }
}
}
VS-2 {
  instance-type virtual-switch;
  interface ge-1/2/1.230;
  interface ge-1/2/2.220;
  interface ge-1/3/0.200;
  interface ae0.210;
  route-distinguisher 198.51.100.1:201;
  vrf-target target:100:201;
  protocols {
    evpn {
      extended-vlan-list [ 210 220 230 ];
    }
  }
  bridge-domains {
    bd-a {
      vlan-id-list [ 210 220 230 ];
    }
  }
}
mhevpn {
  instance-type evpn;
  vlan-id 10;
  interface ge-1/2/1.0;
  interface ge-1/2/2.0;
  interface ge-1/3/0.0;
  interface ae0.0;
  routing-interface irb.10;
  route-distinguisher 198.51.100.1:1;
  vrf-target target:100:1;
  protocols {
    evpn {
      default-gateway do-not-advertise;
    }
  }
}
vrf {
  instance-type vrf;
  interface irb.0;
  interface irb.1;
  interface irb.2;

```

```
interface irb.10;  
  route-distinguisher 198.51.100.1:11;  
  vrf-target target:100:11;  
  vrf-table-label;  
}
```

Verification

Confirm that the configuration is working properly.

- [Verifying VPN Services in the Core on page 178](#)
- [Verifying the EVPN Instance Status on page 180](#)
- [Verifying the Autodiscovery Routes per Ethernet Segment on page 185](#)
- [Verifying the Ethernet Segment Route on page 187](#)
- [Verifying the DF Status on page 189](#)
- [Verifying the BDF Status on page 190](#)
- [Verifying the Remote IRB and Host IP on page 191](#)

Verifying VPN Services in the Core

Purpose Ensure that the protocols in the VPN core are functioning properly.

Action From operational mode, enter the **show isis adjacency** command.

```
user@PE1> show isis adjacency
Interface          System      L State      Hold (secs) SNPA
ge-1/2/4.0         CE10        2 Up          24
5c:5e:ab:e:6f:4
```

From operational mode, enter the **show bgp summary** command.

```
user@PE1> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
bgp.evpn.0
              45         45         0         0         0         0
Peer          AS      InPkt    OutPkt    OutQ    Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
203.0.113.0   65221      90      26      0      0      3:18
Establ
  bgp.evpn.0: 45/45/45/0
  VS-1.evpn.0: 19/19/19/0
  VS-2.evpn.0: 19/19/19/0
  mhevpn.evpn.0: 13/13/13/0
  __default_evpn__.evpn.0: 4/4/4/0
```

```
user@P> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
bgp.evpn.0
              68         68         0         0         0         0
Peer          AS      InPkt    OutPkt    OutQ    Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
198.51.100.1  65221      25      90      0      0      3:04
Establ
  bgp.evpn.0: 22/22/22/0
198.51.100.2  65221      32      80      0      0      6:12
Establ
  bgp.evpn.0: 22/22/22/0
198.51.100.3  65221      31      62      0      0      6:58
Establ
  bgp.evpn.0: 12/12/12/0
198.51.100.4  65221      28      88      0      0      6:04
Establ
  bgp.evpn.0: 12/12/12/0
```

From operational mode, enter the **show mpls lsp** command.

```
user@PE1> show mpls lsp
Ingress LSP: 3 sessions
To          From          State Rt P    ActivePath  LSPName
198.51.100.2 198.51.100.1 Up    0 *    direct_to_pe2 pe1tope2
198.51.100.3 198.51.100.1 Up    0 *    direct_to_pe3 pe1tope3
198.51.100.4 198.51.100.1 Up    0 *    direct_to_pe4 pe1tope4
Total 3 displayed, Up 3, Down 0

Egress LSP: 3 sessions
To          From          State Rt Style Labelin Labelout LSPName
198.51.100.1 198.51.100.3 Up    0 1 FF      3      - pe3tope1
198.51.100.1 198.51.100.4 Up    0 1 FF      3      - pe4tope1
198.51.100.1 198.51.100.2 Up    0 1 FF      3      - pe2tope1
```

Total 3 displayed, Up 3, Down 0

Transit LSP: 0 sessions

Total 0 displayed, Up 0, Down 0

From operational mode, enter the **show interface ae* terse** command.

```
user@PE1> show interface ae* terse
Interface           Admin Link Proto  Local Remote
ae0                  up    up
ae0.0                up    up    bridge
ae0.110              up    up    bridge
ae0.210              up    up    bridge
ae0.32767            up    up    multiservice
ae12                  up    up
ae12.0               up    up    inet    198.51.100.12/24
                        iso
                        mpls
                        multiservice
ae12.32767           up    up    multiservice
ae13                  up    up
ae13.0               up    up    inet    198.51.100.13/24
                        iso
                        mpls
                        multiservice
ae13.32767           up    up    multiservice
```

Meaning The protocols IS-IS, BGP and MPLS are up and running. The aggregated bundles configured on Router PE1 are up.

Verifying the EVPN Instance Status

Purpose Verify the EVPN routing instances and their status.

Action From operational mode, run the **show evpn instance extensive** command.

```
user@PE1> show evpn instance extensive
Instance: VS-1
Route Distinguisher: 198.51.100.1:101
Per-instance MAC route label: 301664
MAC database status
Total MAC addresses: 0
Default gateway MAC addresses: 3
Number of local interfaces: 4 (3 up)
Interface name  ESI                               Mode           Status
ae0.110         00:11:11:11:11:11:11:11:11:11:11:11 all-active     Up
ge-1/2/1.130    00:33:33:33:33:33:33:33:33:33:33:33 all-active     Up
ge-1/2/2.120    00:22:22:22:22:22:22:22:22:22:22:22 all-active     Up
ge-1/3/0.100    00:00:00:00:00:00:00:00:00:00:00:00 single-homed    Up
Number of IRB interfaces: 3 (3 up)
Interface name  VLAN ID  Status  L3 context
irb.0           110      Up      vrf
irb.1           120      Up      vrf
irb.2           130      Up      vrf
Number of bridge domains: 3
```

VLAN ID	Intfs / up	Mode	MAC sync	IM route label
110	2 1	Extended	Enabled	301984
120	2 1	Extended	Enabled	302000
130	2 1	Extended	Enabled	302016

Number of neighbors: 3

198.51.100.2

Received routes

MAC address advertisement: 0

MAC+IP address advertisement: 0

Inclusive multicast: 3

Ethernet auto-discovery: 6

198.51.100.3

Received routes

MAC address advertisement: 0

MAC+IP address advertisement: 0

Inclusive multicast: 1

Ethernet auto-discovery: 2

198.51.100.4

Received routes

MAC address advertisement: 0

MAC+IP address advertisement: 0

Inclusive multicast: 3

Ethernet auto-discovery: 2

Number of ethernet segments: 4

ESI: 00:11:11:11:11:11:11:11:11

Status: Resolved by IFL ae0.110

Local interface: ae0.110, Status: Up/Forwarding

Number of remote PEs connected: 2

Remote PE	MAC label	Aliasing label	Mode
198.51.100.3	0	305584	all-active
198.51.100.2	0	306000	all-active

Designated forwarder: 198.51.100.3

Backup forwarder: 198.51.100.1

Backup forwarder: 198.51.100.2

Advertised MAC label: 301792

Advertised aliasing label: 301792

Advertised split horizon label: 301808

ESI: 00:22:22:22:22:22:22:22:22

Status: Resolved by IFL ge-1/2/2.120

Local interface: ge-1/2/2.120, Status: Up/Forwarding

Number of remote PEs connected: 1

Remote PE	MAC label	Aliasing label	Mode
198.51.100.2	0	306032	all-active

Designated forwarder: 198.51.100.1

Backup forwarder: 198.51.100.2

Advertised MAC label: 301824

Advertised aliasing label: 301824

Advertised split horizon label: 301840

ESI: 00:33:33:33:33:33:33:33:33

Status: Resolved by IFL ge-1/2/1.130

Local interface: ge-1/2/1.130, Status: Up/Forwarding

Number of remote PEs connected: 1

Remote PE	MAC label	Aliasing label	Mode
198.51.100.2	0	306064	all-active

Designated forwarder: 198.51.100.1

Backup forwarder: 198.51.100.2

Advertised MAC label: 301856

Advertised aliasing label: 301856

Advertised split horizon label: 301872

ESI: 00:44:44:44:44:44:44:44:44

Status: Resolved by NH 1048613

```

Number of remote PEs connected: 1
  Remote PE      MAC label  Aliasing label  Mode
  198.51.100.4   0          305152         all-active

Instance: VS-2
Route Distinguisher: 198.51.100.1:201
Per-instance MAC route label: 301696
MAC database status
  Total MAC addresses: 0      0
  Default gateway MAC addresses: 0      0
Number of local interfaces: 4 (3 up)
  Interface name  ESI                                     Mode      Status
  ae0.210         00:11:11:11:11:11:11:11:11:11:11:11  all-active Up
  ge-1/2/1.230    00:33:33:33:33:33:33:33:33:33:33:33  all-active Up
  ge-1/2/2.220    00:22:22:22:22:22:22:22:22:22:22:22  all-active Up
  ge-1/3/0.200    00:00:00:00:00:00:00:00:00:00:00:00  single-homed Down
Number of IRB interfaces: 0 (0 up)
Number of bridge domains: 3
  VLAN ID  Intfs / up  Mode      MAC sync  IM route label
  210      2 1      Extended  Enabled   302032
  220      2 1      Extended  Enabled   302048
  230      2 1      Extended  Enabled   302064
Number of neighbors: 3
  198.51.100.2
    Received routes
      MAC address advertisement: 0
      MAC+IP address advertisement: 0
      Inclusive multicast: 3
      Ethernet auto-discovery: 6
  198.51.100.3
    Received routes
      MAC address advertisement: 0
      MAC+IP address advertisement: 0
      Inclusive multicast: 1
      Ethernet auto-discovery: 2
  198.51.100.4
    Received routes
      MAC address advertisement: 0
      MAC+IP address advertisement: 0
      Inclusive multicast: 3
      Ethernet auto-discovery: 2
Number of ethernet segments: 4
ESI: 00:11:11:11:11:11:11:11:11:11:11
Status: Resolved by IFL ae0.210
Local interface: ae0.210, Status: Up/Forwarding
Number of remote PEs connected: 2
  Remote PE      MAC label  Aliasing label  Mode
  198.51.100.3   0          305648         all-active
  198.51.100.2   0          306096         all-active
Designated forwarder: 198.51.100.1
Backup forwarder: 198.51.100.2
Backup forwarder: 198.51.100.3
Advertised MAC label: 301888
Advertised aliasing label: 301888
Advertised split horizon label: 301808
ESI: 00:22:22:22:22:22:22:22:22:22:22
Status: Resolved by IFL ge-1/2/2.220
Local interface: ge-1/2/2.220, Status: Up/Forwarding
Number of remote PEs connected: 1
  Remote PE      MAC label  Aliasing label  Mode
  198.51.100.2   0          306112         all-active

```

```

    Designated forwarder: 198.51.100.1
    Backup forwarder: 198.51.100.2
    Advertised MAC label: 301904
    Advertised aliasing label: 301904
    Advertised split horizon label: 301840
    ESI: 00:33:33:33:33:33:33:33:33
    Status: Resolved by IFL ge-1/2/1.230
    Local interface: ge-1/2/1.230, Status: Up/Forwarding
    Number of remote PEs connected: 1
      Remote PE      MAC label  Aliasing label  Mode
      198.51.100.2    0          306128          all-active
    Designated forwarder: 198.51.100.1
    Backup forwarder: 198.51.100.2
    Advertised MAC label: 301920
    Advertised aliasing label: 301920
    Advertised split horizon label: 301872
    ESI: 00:44:44:44:44:44:44:44:44
    Status: Resolved by NH 1048616
    Number of remote PEs connected: 1
      Remote PE      MAC label  Aliasing label  Mode
      198.51.100.4    0          305184          all-active

Instance: __default_evpn__
Route Distinguisher: 198.51.100.1:0
VLAN ID: None
Per-instance MAC route label: 301760
MAC database status
  Total MAC addresses:          Local Remote
  Default gateway MAC addresses: 0      0
Number of local interfaces: 0 (0 up)
Number of IRB interfaces: 0 (0 up)
Number of bridge domains: 0
Number of neighbors: 2
  198.51.100.2
    Received routes
      Ethernet auto-discovery:          0
      Ethernet Segment:                  3
  198.51.100.3
    Received routes
      Ethernet auto-discovery:          0
      Ethernet Segment:                  1
Number of ethernet segments: 0

Instance: mhevpn
Route Distinguisher: 198.51.100.1:1
VLAN ID: 10
Per-instance MAC route label: 301728
MAC database status
  Total MAC addresses:          Local Remote
  Default gateway MAC addresses: 1      0
Number of local interfaces: 4 (3 up)
  Interface name  ESI                                     Mode      Status
  ae0.0           00:11:11:11:11:11:11:11:11             all-active Up
  ge-1/2/1.0      00:33:33:33:33:33:33:33:33             all-active Up
  ge-1/2/2.0      00:22:22:22:22:22:22:22:22             all-active Up
  ge-1/3/0.0      00:00:00:00:00:00:00:00:00             single-homed Down
Number of IRB interfaces: 1 (1 up)
  Interface name  VLAN ID  Status  L3 context
  irb.10          10       Up      vrf
Number of bridge domains: 1
  VLAN ID  Intfs / up  Mode      MAC sync  IM route label

```

```

10          4 3      Extended      Enabled  302080
Number of neighbors: 3
198.51.100.2
  Received routes
    MAC address advertisement:      0
    MAC+IP address advertisement:   0
    Inclusive multicast:            1
    Ethernet auto-discovery:        6
198.51.100.3
  Received routes
    MAC address advertisement:      0
    MAC+IP address advertisement:   0
    Inclusive multicast:            1
    Ethernet auto-discovery:        2
198.51.100.4
  Received routes
    MAC address advertisement:      0
    MAC+IP address advertisement:   0
    Inclusive multicast:            1
    Ethernet auto-discovery:        2
Number of ethernet segments: 4
ESI: 00:11:11:11:11:11:11:11:11
Status: Resolved by IFL ae0.0
Local interface: ae0.0, Status: Up/Forwarding
Number of remote PEs connected: 2
  Remote PE      MAC label  Aliasing label  Mode
  198.51.100.3   0          305680         all-active
  198.51.100.2   0          306144         all-active
Designated forwarder: 198.51.100.2
Backup forwarder: 198.51.100.1
Backup forwarder: 198.51.100.3
Advertised MAC label: 301936
Advertised aliasing label: 301936
Advertised split horizon label: 301808
ESI: 00:22:22:22:22:22:22:22:22
Status: Resolved by IFL ge-1/2/2.0
Local interface: ge-1/2/2.0, Status: Up/Forwarding
Number of remote PEs connected: 1
  Remote PE      MAC label  Aliasing label  Mode
  198.51.100.2   0          306160         all-active
Designated forwarder: 198.51.100.1
Backup forwarder: 198.51.100.2
Advertised MAC label: 301952
Advertised aliasing label: 301952
Advertised split horizon label: 301840
ESI: 00:33:33:33:33:33:33:33:33
Status: Resolved by IFL ge-1/2/1.0
Local interface: ge-1/2/1.0, Status: Up/Forwarding
Number of remote PEs connected: 1
  Remote PE      MAC label  Aliasing label  Mode
  198.51.100.2   0          306176         all-active
Designated forwarder: 198.51.100.1
Backup forwarder: 198.51.100.2
Advertised MAC label: 301968
Advertised aliasing label: 301968
Advertised split horizon label: 301872
ESI: 00:44:44:44:44:44:44:44:44
Status: Resolved by NH 1048612
Number of remote PEs connected: 1
  Remote PE      MAC label  Aliasing label  Mode
  198.51.100.4   0          305200         all-active

```


Meaning The output provides the following information:

- List of EVPN and virtual switch routing instances
- Mode of operation of each interface
- Neighbors of each routing instance
- Number of different routes received from each neighbor
- ESI attached to each routing instance
- Number of Ethernet segments on each routing instance
- DF election roles for each ESI in an EVI
- VLAN ID and MAC labels for each routing instance
- IRB interface details
- Number of default gateway MAC addresses received for the virtual switch routing instance (VS-1 and VS-2)

Verifying the Autodiscovery Routes per Ethernet Segment

Purpose Verify that the autodiscovery routes per Ethernet segment are received.

Action From operational mode, run the **show route table mhevpn.evpn.0** command.

Router PE1

```
user@PE1> show route table mhevpn.evpn.0
mhevpn.evpn.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:198.51.100.1:1::111111111111111111::0/304
    *[EVPN/170] 00:11:37
    Indirect
1:198.51.100.1:1::222222222222222222::0/304
    *[EVPN/170] 00:11:37
    Indirect
1:198.51.100.1:1::333333333333333333::0/304
    *[EVPN/170] 00:11:37
    Indirect
1:198.51.100.2:0::111111111111111111::FFFF:FFFF/304
    *[BGP/170] 00:11:33, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:0::222222222222222222::FFFF:FFFF/304
    *[BGP/170] 00:11:33, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:0::333333333333333333::FFFF:FFFF/304
    *[BGP/170] 00:11:33, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:1::111111111111111111::0/304
```

```

* [BGP/170] 00:11:33, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:1::22222222222222222222::0/304
* [BGP/170] 00:11:33, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:1::33333333333333333333::0/304
* [BGP/170] 00:11:33, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.3:0::11111111111111111111::FFFF:FFFF/304
* [BGP/170] 00:11:37, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.6 via ae13.0, label-switched-path pe1tope3
1:198.51.100.3:1::11111111111111111111::0/304
* [BGP/170] 00:11:37, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.6 via ae13.0, label-switched-path pe1tope3
3:198.51.100.1:1:10::198.51.100.1/304
* [EVPN/170] 00:13:38
  Indirect
3:198.51.100.2:1:10::198.51.100.2/304
* [BGP/170] 00:11:33, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
3:198.51.100.3:1:10::198.51.100.3/304
* [BGP/170] 00:11:37, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.6 via ae13.0, label-switched-path pe1tope3

```

Router PE2

```

user@PE2> show route table mhevpn.evpn.0
mhevpn.evpn.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:198.51.100.1:0::11111111111111111111::FFFF:FFFF/304
* [BGP/170] 01:10:26, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:0::22222222222222222222::FFFF:FFFF/304
* [BGP/170] 01:10:26, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:0::33333333333333333333::FFFF:FFFF/304
* [BGP/170] 01:10:26, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:1:11111111111111111111::0/304
* [BGP/170] 01:10:26, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:1:22222222222222222222::0/304
* [BGP/170] 01:10:26, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:1:33333333333333333333::0/304
* [BGP/170] 01:10:26, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
  > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1

```

```

1:198.51.100.2:1::11111111111111111111::0/304
    *[EVPN/170] 01:10:26
    Indirect
1:198.51.100.2:1::22222222222222222222::0/304
    *[EVPN/170] 01:10:26
    Indirect
1:198.51.100.2:1::33333333333333333333::0/304
    *[EVPN/170] 01:10:26
    Indirect
1:198.51.100.3:0::11111111111111111111::FFFF:FFFF/304
    *[BGP/170] 01:10:26, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.7 via ge-0/0/4.0, label-switched-path pe2tope3
1:198.51.100.3:1::11111111111111111111::0/304
    *[BGP/170] 01:10:26, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.7 via ge-0/0/4.0, label-switched-path pe2tope3
1:198.51.100.4:0::44444444444444444444::FFFF:FFFF/304
    *[BGP/170] 01:10:17, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.8 via ge-0/1/8.0, label-switched-path pe2tope4
1:198.51.100.4:1::44444444444444444444::0/304
    *[BGP/170] 01:10:17, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.8 via ge-0/1/8.0, label-switched-path pe2tope4
3:198.51.100.1:1::10::198.51.100.1/304
    *[BGP/170] 01:10:26, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
3:198.51.100.2:1::10::198.51.100.2/304
    *[EVPN/170] 01:12:14
    Indirect
3:198.51.100.3:1::10::198.51.100.3/304
    *[BGP/170] 01:10:26, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.7 via ge-0/0/4.0, label-switched-path pe2tope3
3:198.51.100.4:1::10::198.51.100.4/304
    *[BGP/170] 01:10:17, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.8 via ge-0/1/8.0, label-switched-path pe2tope4

```

Meaning The remote type 1 autodiscovery route is received for the ESI attached to Router PE2, which is the other PE router connected to the multihomed CE device.

Verifying the Ethernet Segment Route

Purpose Verify that the local and advertised autodiscovery routes per Ethernet segment and the Ethernet segment routes are received.

Action From operational mode, run the **show route table __default_evpn__evpn.0** command.

Router PE1

```

user@PE1> show route table __default_evpn__evpn.0
__default_evpn__evpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0
hidden)

```

```

+ = Active Route, - = Last Active, * = Both

1:198.51.100.1:0::111111111111111111::FFFF:FFFF/304
    *[EVPN/170] 00:25:18
    Indirect
1:198.51.100.1:0::222222222222222222::FFFF:FFFF/304
    *[EVPN/170] 00:25:18
    Indirect
1:198.51.100.1:0::333333333333333333::FFFF:FFFF/304
    *[EVPN/170] 00:25:18
    Indirect
4:198.51.100.1:0::111111111111111111:198.51.100.1/304
    *[EVPN/170] 00:25:18
    Indirect
4:198.51.100.1:0::222222222222222222:198.51.100.1/304
    *[EVPN/170] 00:25:18
    Indirect
4:198.51.100.1:0::333333333333333333:198.51.100.1/304
    *[EVPN/170] 00:25:18
    Indirect
4:198.51.100.2:0::111111111111111111:198.51.100.2/304
    *[BGP/170] 00:25:14, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
4:198.51.100.2:0::222222222222222222:198.51.100.2/304
    *[BGP/170] 00:25:14, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
4:198.51.100.2:0::333333333333333333:198.51.100.2/304
    *[BGP/170] 00:25:14, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
4:198.51.100.3:0::111111111111111111:198.51.100.3/304
    *[BGP/170] 00:25:18, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.6 via ae13.0, label-switched-path pe1tope3

```

Router PE2

```

user@PE2> show route table __default_evpn__.evpn.0
__default_evpn__.evpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

1:198.51.100.2:0::111111111111111111::FFFF:FFFF/304
    *[EVPN/170] 01:17:59
    Indirect
1:198.51.100.2:0::222222222222222222::FFFF:FFFF/304
    *[EVPN/170] 01:17:59
    Indirect
1:198.51.100.2:0::333333333333333333::FFFF:FFFF/304
    *[EVPN/170] 01:17:59
    Indirect
4:198.51.100.1:0::111111111111111111:198.51.100.1/304
    *[BGP/170] 01:17:59, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
4:198.51.100.1:0::222222222222222222:198.51.100.1/304
    *[BGP/170] 01:17:59, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1

```

```

4:198.51.100.1:0::33333333333333333333333333333333:198.51.100.1/304
    * [BGP/170] 01:17:59, localpref 100, from 203.0.113.0
      AS path: I, validation-state: unverified
      > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
4:198.51.100.2:0::11111111111111111111111111111111:198.51.100.2/304
    * [EVPN/170] 01:17:59
      Indirect
4:198.51.100.2:0::22222222222222222222222222222222:198.51.100.2/304
    * [EVPN/170] 01:17:59
      Indirect
4:198.51.100.2:0::33333333333333333333333333333333:198.51.100.2/304
    * [EVPN/170] 01:17:59
      Indirect
4:198.51.100.3:0::11111111111111111111111111111111:198.51.100.3/304
    * [BGP/170] 01:17:59, localpref 100, from 203.0.113.0
      AS path: I, validation-state: unverified
      > to 198.51.100.7 via ge-0/0/4.0, label-switched-path pe2tope3

```

Meaning The output displays the local and remote type 1 (autodiscovery) and type 4 (Ethernet segment) routes.

Verifying the DF Status

Purpose Confirm which PE router is the designated forwarder (DF) for each routing instance.

Action From operational mode, run the **show evpn instance designated-forwarder** command.

```

user@PE1> show evpn instance designated forwarder
Instance: VS-1
  Number of ethernet segments: 4
    ESI: 00:11:11:11:11:11:11:11:11:11
      Designated forwarder: 198.51.100.3
    ESI: 00:22:22:22:22:22:22:22:22:22
      Designated forwarder: 198.51.100.1
    ESI: 00:33:33:33:33:33:33:33:33:33
      Designated forwarder: 198.51.100.1
    ESI: 00:44:44:44:44:44:44:44:44:44
      Designated forwarder: No local attachment to ethernet segment

Instance: VS-2
  Number of ethernet segments: 4
    ESI: 00:11:11:11:11:11:11:11:11:11
      Designated forwarder: 198.51.100.1
    ESI: 00:22:22:22:22:22:22:22:22:22
      Designated forwarder: 198.51.100.1
    ESI: 00:33:33:33:33:33:33:33:33:33
      Designated forwarder: 198.51.100.1
    ESI: 00:44:44:44:44:44:44:44:44:44
      Designated forwarder: No local attachment to ethernet segment

Instance: mhevpn
  Number of ethernet segments: 4
    ESI: 00:11:11:11:11:11:11:11:11:11
      Designated forwarder: 198.51.100.2
    ESI: 00:22:22:22:22:22:22:22:22:22
      Designated forwarder: 198.51.100.1

```

```
ESI: 00:33:33:33:33:33:33:33:33:33
  Designated forwarder: 198.51.100.1
ESI: 00:44:44:44:44:44:44:44:44:44
  Designated forwarder: No local attachment to ethernet segment
```

Meaning The designated forwarder is displayed for each routing instance and ESI.

Verifying the BDF Status

Purpose Confirm which PE router is the backup designated forwarder (BDF) for each routing instance.

Action From operational mode, run the **show evpn instance backup-forwarder** command.

```
user@PE1> show evpn instance backup-forwarder
Instance: VS-1
  Number of ethernet segments: 4
    ESI: 00:11:11:11:11:11:11:11:11:11
      Backup forwarder: 198.51.100.1
      Backup forwarder: 198.51.100.2
    ESI: 00:22:22:22:22:22:22:22:22:22
      Backup forwarder: 198.51.100.2
    ESI: 00:33:33:33:33:33:33:33:33:33
      Backup forwarder: 198.51.100.2
    ESI: 00:44:44:44:44:44:44:44:44:44
      Backup forwarder: No local attachment to ethernet segment

Instance: VS-2
  Number of ethernet segments: 4
    ESI: 00:11:11:11:11:11:11:11:11:11
      Backup forwarder: 198.51.100.2
      Backup forwarder: 198.51.100.3
    ESI: 00:22:22:22:22:22:22:22:22:22
      Backup forwarder: 198.51.100.2
    ESI: 00:33:33:33:33:33:33:33:33:33
      Backup forwarder: 198.51.100.2
    ESI: 00:44:44:44:44:44:44:44:44:44
      Backup forwarder: No local attachment to ethernet segment

Instance: mhevpn
  Number of ethernet segments: 4
    ESI: 00:11:11:11:11:11:11:11:11:11
      Backup forwarder: 198.51.100.1
      Backup forwarder: 198.51.100.3
    ESI: 00:22:22:22:22:22:22:22:22:22
      Backup forwarder: 198.51.100.2
    ESI: 00:33:33:33:33:33:33:33:33:33
      Backup forwarder: 198.51.100.2
    ESI: 00:44:44:44:44:44:44:44:44:44
      Backup forwarder: No local attachment to ethernet segment
```

Meaning The backup designated forwarder is displayed for each routing instance and ESI.

Verifying the Remote IRB and Host IP

Purpose Verify that the remote IRB IP and the host IP are received.

Router PE1 Action

From operational mode, run the **show route table mhevpn** command.

```
user@PE1> show route table mhevpn
mhevpn.evpn.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:198.51.100.1:1::1111111111111111::0/304
    *[EVPN/170] 01:34:26
    Indirect
1:198.51.100.1:1::2222222222222222::0/304
    *[EVPN/170] 01:34:26
    Indirect
1:198.51.100.1:1::3333333333333333::0/304
    *[EVPN/170] 01:34:26
    Indirect
1:198.51.100.2:0::1111111111111111::FFFF:FFFF/304
    *[BGP/170] 01:34:22, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:0::2222222222222222::FFFF:FFFF/304
    *[BGP/170] 01:34:22, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:0::3333333333333333::FFFF:FFFF/304
    *[BGP/170] 01:34:22, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:1::1111111111111111::0/304
    *[BGP/170] 01:34:22, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:1::2222222222222222::0/304
    *[BGP/170] 01:34:22, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.2:1::3333333333333333::0/304
    *[BGP/170] 01:34:22, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
1:198.51.100.3:0::1111111111111111::FFFF:FFFF/304
    *[BGP/170] 01:34:26, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.6 via ae13.0, label-switched-path pe1tope3
1:198.51.100.3:1::1111111111111111::0/304
    *[BGP/170] 01:34:26, localpref 100, from 203.0.113.0
    AS path: I, validation-state: unverified
    > to 198.51.100.6 via ae13.0, label-switched-path pe1tope3
3:198.51.100.1:1:10::198.51.100.1/304
    *[EVPN/170] 01:36:27
    Indirect
3:198.51.100.2:1:10::198.51.100.2/304
    *[BGP/170] 01:34:22, localpref 100, from 203.0.113.0
```

```

        AS path: I, validation-state: unverified
        > to 198.51.100.5 via ae12.0, label-switched-path pe1tope2
3:198.51.100.3:1::10::198.51.100.3/304
    *[BGP/170] 01:34:26, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.6 via ae13.0, label-switched-path pe1tope3

```

Router PE2

From operational mode, run the **show route table mhevpn** command.

```

user@PE2> show route table mhevpn
mhevpn.evpn.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:198.51.100.1:0::11111111111111111111::FFFF:FFFF/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:0::22222222222222222222::FFFF:FFFF/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:0::33333333333333333333::FFFF:FFFF/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:1::11111111111111111111::0/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:1::22222222222222222222::0/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.1:1::33333333333333333333::0/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
1:198.51.100.2:1::11111111111111111111::0/304
    *[EVPN/170] 01:35:11
        Indirect
1:198.51.100.2:1::22222222222222222222::0/304
    *[EVPN/170] 01:35:11
        Indirect
1:198.51.100.2:1::33333333333333333333::0/304
    *[EVPN/170] 01:35:11
        Indirect
1:198.51.100.3:0::11111111111111111111::FFFF:FFFF/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.7 via ge-0/0/4.0, label-switched-path pe2tope3
1:198.51.100.3:1::11111111111111111111::0/304
    *[BGP/170] 01:35:11, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.7 via ge-0/0/4.0, label-switched-path pe2tope3
1:198.51.100.4:0::44444444444444444444::FFFF:FFFF/304
    *[BGP/170] 01:35:02, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.8 via ge-0/1/8.0, label-switched-path pe2tope4
1:198.51.100.4:1::44444444444444444444::0/304

```



```

* [BGP/170] 01:35:02, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.8 via ge-0/1/8.0, label-switched-path pe2tope4
3:198.51.100.1:1::10::198.51.100.1/304
* [BGP/170] 01:35:11, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.12 via ae12.0, label-switched-path pe2tope1
3:198.51.100.2:1::10::198.51.100.2/304
* [EVPN/170] 01:36:59
  Indirect
3:198.51.100.3:1::10::198.51.100.3/304
* [BGP/170] 01:35:11, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.7 via ge-0/0/4.0, label-switched-path pe2tope3
3:198.51.100.4:1::10::198.51.100.4/304
* [BGP/170] 01:35:02, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.8 via ge-0/1/8.0, label-switched-path pe2tope4

```

Router PE3

From operational mode, run the **show route table mhevpn** command.

```

user@PE3> show route table mhevpn
mhevpn.evpn.0: 19 destinations, 19 routes (19 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:198.51.100.1:0::11111111111111111111::FFFF:FFFF/304
* [BGP/170] 01:36:10, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.13 via ae13.0, label-switched-path pe3tope1
1:198.51.100.1:0::22222222222222222222::FFFF:FFFF/304
* [BGP/170] 01:36:10, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.13 via ae13.0, label-switched-path pe3tope1
1:198.51.100.1:0::33333333333333333333::FFFF:FFFF/304
* [BGP/170] 01:36:10, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.13 via ae13.0, label-switched-path pe3tope1
1:198.51.100.1:1::11111111111111111111::0/304
* [BGP/170] 01:36:10, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.13 via ae13.0, label-switched-path pe3tope1
1:198.51.100.1:1::22222222222222222222::0/304
* [BGP/170] 01:36:10, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.13 via ae13.0, label-switched-path pe3tope1
1:198.51.100.1:1::33333333333333333333::0/304
* [BGP/170] 01:36:10, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.13 via ae13.0, label-switched-path pe3tope1
1:198.51.100.2:0::11111111111111111111::FFFF:FFFF/304
* [BGP/170] 01:36:06, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.10 via ge-2/0/3.0, label-switched-path pe3tope2
1:198.51.100.2:0::22222222222222222222::FFFF:FFFF/304
* [BGP/170] 01:36:06, localpref 100, from 203.0.113.0
  AS path: I, validation-state: unverified
> to 198.51.100.10 via ge-2/0/3.0, label-switched-path pe3tope2
1:198.51.100.2:0::33333333333333333333::FFFF:FFFF/304
* [BGP/170] 01:36:06, localpref 100, from 203.0.113.0

```

```

        AS path: I, validation-state: unverified
        > to 198.51.100.10 via ge-2/0/3.0, label-switched-path pe3tope2
1:198.51.100.2:1::11111111111111111111::0/304
        *[BGP/170] 01:36:06, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.10 via ge-2/0/3.0, label-switched-path pe3tope2
1:198.51.100.2:1::22222222222222222222::0/304
        *[BGP/170] 01:36:06, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.10 via ge-2/0/3.0, label-switched-path pe3tope2
1:198.51.100.2:1::33333333333333333333::0/304
        *[BGP/170] 01:36:06, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.10 via ge-2/0/3.0, label-switched-path pe3tope2
1:198.51.100.3:1::11111111111111111111::0/304
        *[EVPN/170] 01:36:25
        Indirect
1:198.51.100.4:0::44444444444444444444::FFFF:FFFF/304
        *[BGP/170] 01:35:58, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.11 via ge-2/1/3.0, label-switched-path pe3tope4
1:198.51.100.4:1::44444444444444444444::0/304
        *[BGP/170] 01:35:58, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.11 via ge-2/1/3.0, label-switched-path pe3tope4
3:198.51.100.1:1::10::198.51.100.1/304
        *[BGP/170] 01:36:10, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.13 via ae13.0, label-switched-path pe3tope1
3:198.51.100.2:1::10::198.51.100.2/304
        *[BGP/170] 01:36:06, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.10 via ge-2/0/3.0, label-switched-path pe3tope2
3:198.51.100.3:1::10::198.51.100.3/304
        *[EVPN/170] 01:37:33
        Indirect
3:198.51.100.4:1::10::198.51.100.4/304
        *[BGP/170] 01:35:58, localpref 100, from 203.0.113.0
        AS path: I, validation-state: unverified
        > to 198.51.100.11 via ge-2/1/3.0, label-switched-path pe3tope4

```

Meaning The output displays the local and remote IRB interfaces. It also displays the local and remote hosts that are installed in the VRF table.

Release History Table

Release	Description
16.1R4	Starting with Junos OS Release 16.1R4, EVPN multihoming active-active mode is supported on all EX9200 switches.

Related Documentation

Example: Configuring an ESI on a Logical Interface With EVPN Multihoming

When a customer edge (CE) device in an Ethernet VPN-Multiprotocol Label Switching (EVPN-MPLS) environment is multihomed to two or more provider edge (PE) devices, the set of Ethernet links that connect the devices comprise an Ethernet segment. An Ethernet segment identifier (ESI) is a 10-octet integer that identifies this segment. A sample ESI is 00:11:22:33:44:55:66:77:88:99.

In releases before Junos OS Release 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI only on a physical or aggregated Ethernet interface, for example, **set interfaces ae0 esi 00:11:22:33:44:55:66:77:88:99**. If you specify an ESI on a physical or aggregated Ethernet interface, keep in mind that an ESI is a factor in the designated forwarder (DF) election process. For example, assume that you configure EVPN multihoming active-standby on aggregated Ethernet interface ae0, and given the ESI configured on ae0 and other determining factors, the DF election results in ae0 being in the down state. Further, all logical interfaces configured on ae0, for example, **set interfaces ae0 unit 1** and **set interfaces ae0 unit 2** are also in the down state, which renders logical interfaces ae0.1 and ae0.2 unable to provide services to their respective customer sites (VLANs).

Starting with Junos OS Releases 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI on a logical interface. If you specify an ESI on a logical interface, the DF election process now occurs at the individual logical interface level, which enables you to better utilize logical interfaces. For example, assume that you configure logical interfaces ae0.1 and ae0.2 on aggregated Ethernet interface ae0. You configure EVPN multihoming active-standby on both logical interfaces and given the ESI configured on ae0.1 and other determining factors, the DF election results in ae0.1 being in the down state. Despite logical interface ae0.1 being down, logical interface ae0.2 and other logical interfaces configured on ae0 can be in the up state and provide services to their respective customer sites (VLANs).

This topic shows how to configure an ESI on logical interfaces in both EVPN multihoming active-standby and active-active modes.

- [Requirements on page 195](#)
- [Overview and Topology on page 196](#)
- [EVPN Multihoming Active-Standby Configuration on page 198](#)
- [Verification on page 205](#)

Requirements

Both EVPN multihoming active-standby and multihoming active-active examples use the following hardware and software components:

- An EX9200 switch running Junos OS Release 17.3R1 or later (PE1)
- An MX Series router running Junos OS Release 15.1F6 or later, or Junos OS Release 17.1R1 or later (PE2)

Overview and Topology

EVPN Multihoming Active-Standby

Figure 9 on page 196 shows an EVPN-MPLS topology in which CE1 is multihomed to PE1 and PE2 to provide redundant paths to CE2. On CE1, the connections to PE1 and PE2 are configured as separate aggregated Ethernet interfaces. Table 4 on page 196 shows how the connections with CE1 are configured on PE1 and PE2. Note that the EVPN multihoming mode, ESI, and VLAN ID are actually configured on logical interfaces ae0.1 on each PE device.

Figure 9: EVPN-MPLS Topology with Multihoming Active-Standby

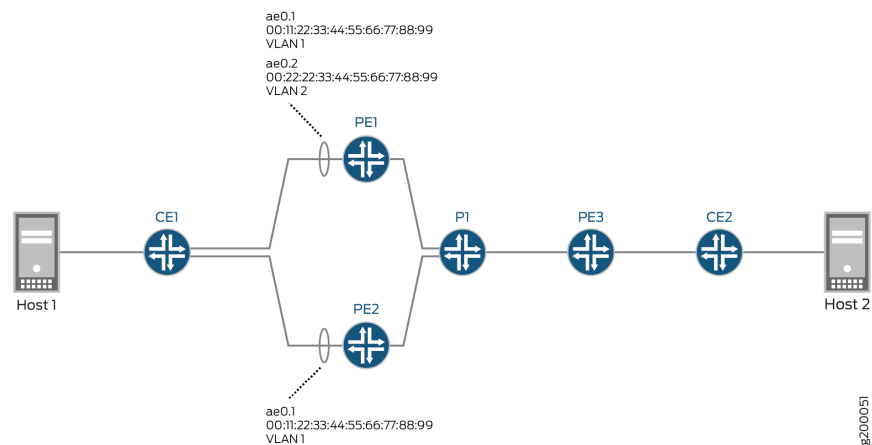


Table 4: EVPN Multihoming Active-Standby: Configuring the Connection with CE1 on PE1 and PE2

Device	Physical Interface	Aggregated Ethernet Interface	Logical Interface	EVPN Multihoming Mode	ESI	VLAN ID
PE1	xe-2/0/0	ae0	ae0.1	single-active	00:11:22:33:44:55:66:77:88:99	1
PE2	xe-3/0/2	ae0	ae0.1	single-active	00:11:22:33:44:55:66:77:88:99	1

Based on the DF election, logical interface ae0.1 on PE2 is up, and logical interface ae0.1 on PE1 is down.

Table 5 on page 197 also shows the configuration for logical interface ae0.2 on PE1. Note that logical interface ae0.2 provides services for a different VLAN and is configured with a different ESI than logical interface ae0.1, which is configured on the same aggregated Ethernet interface. As a result, logical interface ae0.2 is up and providing services to VLAN 2 despite the fact that logical interface ae0.1 is in the down state.

Table 5: Multihoming Active-Standby: Configuring Logical Interface on PE1 that Provides Services to a Different VLAN

Device	Physical Interface	Aggregated Ethernet Interface	Logical Interface	EVPN Multihoming Mode	ESI	VLAN ID
PE1	xe-2/0/0	ae0	ae0.2	single-active	00:22:33:44:55:66:77:88:99	2

EVPN Multihoming Active-Active

Figure 10 on page 197 shows an EVPN-MPLS topology in which CE1 is multihomed to PE1 and PE2 to provide redundant paths to CE2. On CE1, the connections to PE1 and PE2 are configured as one aggregated Ethernet interface. Table 6 on page 197 shows how the connections with CE1 are configured on PE1 and PE2. Note that the EVPN multihoming mode, ESI, and VLAN ID are actually configured on logical interfaces ae0.1 on each device.

Figure 10: EVPN-MPLS Topology with Multihoming Active-Active

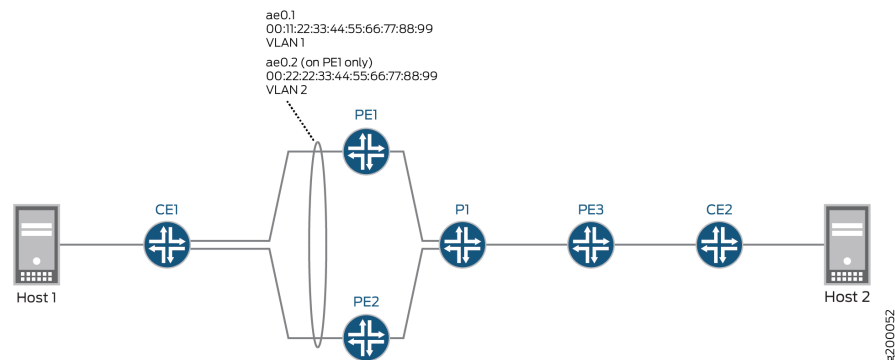


Table 6: Multihoming Active-Active: Configuring the Connection with CE1 on PE1 and PE2

Device	Physical Interface	Aggregated Ethernet Interface	Logical Interface	EVPN Multihoming Mode	ESI	VLAN ID
PE1	xe-2/0/0	ae0	ae0.1	all-active	00:11:22:33:44:55:66:77:88:99	1
PE2	xe-3/0/2	ae0	ae0.1	all-active	00:11:22:33:44:55:66:77:88:99	1

Based on the DF election, logical interface ae0.1 on PE1 is in the up state, and logical interface ae0.1 on PE2 is in the down state.

Table 7 on page 198 also shows the configuration for logical interface ae0.2 on PE1. Note that logical interface ae0.2 provides services for a different VLAN and is configured with a different ESI than logical interface ae0.1, which is in the same aggregated Ethernet interface. As a result, logical interface ae0.2 is down and unable to provide services to VLAN 2 despite the fact that logical interface ae0.1 is in the up state.

Table 7: EVPN Multihoming Active-Active: Configuring Interface on PE1 that Provides Services to a Different VLAN

Device	Physical Interface	Aggregated Ethernet Interface	Logical Interface	EVPN Multihoming Mode	ESI	VLAN ID
PE1	xe-2/0/0	ae0	ae0.2	all-active	00:22:22:33:44:55:66:77:88:99	2

EVPN Multihoming Active-Standby Configuration



NOTE: The configurations for PE1 (EX9200) and PE2 (MX Series router) focus on configuring EVPN multihoming active-standby and ESIs on logical interfaces. The configurations do not include all EVPN-related configurations for physical interfaces, aggregated Ethernet interfaces, logical interfaces, and routing instances. For a more comprehensive configuration of EVPN multihoming active-standby in an EVPN-MPLS environment, see *Example: Configuring EVPN Active-Standby Multihoming*. Note that the referenced example shows how to configure ESIs on physical and aggregated Ethernet interfaces.

CLI Configuration

```

set interfaces xe-2/0/0 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 unit 1 encapsulation vlan-bridge
set interfaces ae0 unit 1 vlan-id 1
set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
set interfaces ae0 unit 1 esi single-active
set interfaces ae0 unit 2 encapsulation vlan-bridge
set interfaces ae0 unit 2 vlan-id 2
set interfaces ae0 unit 2 esi 00:22:22:33:44:55:66:77:88:99
set interfaces ae0 unit 2 esi single-active
set interfaces irb unit 1 family inet address 192.0.2.1/24
set interfaces irb unit 2 family inet address 192.0.2.2/24
set routing-instances blue instance-type evpn
set routing-instances blue vlan-id 1
set routing-instances blue interface ae0.1
set routing-instances blue l3-interface irb.1
...
set routing-instances blue protocols evpn interface ae0.1
set routing-instances green instance-type evpn
set routing-instances green vlan-id 2
set routing-instances green interface ae0.2
set routing-instances green l3-interface irb.2
...
set routing-instances green protocols evpn interface ae0.2
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.1
set routing-instances vrf interface irb.2
...

```

```

PE2  set interfaces xe-3/0/2 gigether-options 802.3ad ae0
      set interfaces ae0 flexible-vlan-tagging
      set interfaces ae0 encapsulation flexible-ethernet-services
      set interfaces ae0 unit 1 encapsulation vlan-bridge
      set interfaces ae0 unit 1 vlan-id 1
      set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
      set interfaces ae0 unit 1 esi single-active
      set interfaces irb unit 1 family inet address 192.0.2.1/24
      set routing-instances blue instance-type evpn
      set routing-instances blue vlan-id 1
      set routing-instances blue interface ae0.1
      set routing-instances blue routing-interface irb.1
      ...
      set routing-instances blue protocols evpn interface ae0.1
      set routing-instances vrf instance-type vrf
      set routing-instances vrf interface irb.1
      ...

```

Step-by-Step Procedure

To configure EVPN multihoming active-standby on PE1:

- Specify an Ethernet interface as a member of aggregated Ethernet interface ae0.

```

[edit interfaces]
user@switch# set xe-2/0/0 gigether-options 802.3ad ae0

```
- Configure aggregated Ethernet interface ae0 to simultaneously transmit 802.1Q VLAN single-tag and dual-tag frames and to support different types of Ethernet encapsulation at the logical interface level.

```

[edit interfaces]
user@switch# set ae0 flexible-vlan-tagging
user@switch# set ae0 encapsulation flexible-ethernet-services

```
- On aggregated Ethernet interface ae0, configure logical interfaces ae0.1 and ae0.2. Configure the logical interfaces to use VLAN bridge encapsulation, and map the logical interfaces to VLANs 1 and 2, respectively. Also, assign an ESI to the logical interfaces, and enable EVPN multihoming active-standby.

```

[edit interfaces]
user@switch# set ae0 unit 1 encapsulation vlan-bridge
user@switch# set ae0 unit 1 vlan-id 1
user@switch# set ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
user@switch# set ae0 unit 1 esi single-active
user@switch# set ae0 unit 2 encapsulation vlan-bridge
user@switch# set ae0 unit 2 vlan-id 2
user@switch# set ae0 unit 2 esi 00:22:22:33:44:55:66:77:88:99
user@switch# set ae0 unit 2 esi single-active

```
- Configure IRB interfaces irb.1 and irb.2, and assign an IP address to each interface.

```

[edit interfaces]
set interfaces irb unit 1 family inet address 192.0.2.1/24
set interfaces irb unit 2 family inet address 192.0.2.2/24

```

5. Configure an EVPN routing instance named blue. Map the routing instance to VLAN 1, logical interface ae0.1, and IRB interface irb.1. Configure EVPN logical interface ae0.1 for the EVPN routing instance.

```
[edit routing-instances]
user@switch# set blue instance-type evpn
user@switch# set blue vlan-id 1
user@switch# set blue interface ae0.1
user@switch# set blue l3-interface irb.1
user@switch# set blue protocols evpn interface ae0.1
```

6. Configure an EVPN routing instance named green. Map the routing instance to VLAN 2, logical interface ae0.2, and IRB interface irb.2. Configure logical interface ae0.2 for the EVPN routing instance.

```
[edit routing-instances]
user@switch# set green instance-type evpn
user@switch# set green vlan-id 2
user@switch# set green interface ae0.2
user@switch# set green l3-interface irb.2
user@switch# set green protocols evpn interface ae0.2
```

7. Configure a VRF routing instance, and add IRB interfaces irb.1 and irb.2 to the routing instance.

```
[edit routing-instances]
set vrf instance-type vrf
set vrf interface irb.1
set vrf interface irb.2
```

Step-by-Step Procedure

To configure EVPN multihoming active-standby on PE2:

1. Specify an Ethernet interface as a member of aggregated Ethernet interface ae0.

```
[edit interfaces]
user@router# set xe-3/0/2 gigether-options 802.3ad ae0
```

2. Configure aggregated Ethernet interface ae0 to simultaneously transmit 802.1Q VLAN single-tag and dual-tag frames and to support different types of Ethernet encapsulation at the logical interface level.

```
[edit interfaces]
user@router# set ae0 flexible-vlan-tagging
user@router# set ae0 encapsulation flexible-ethernet-services
```

3. On aggregated Ethernet interface ae0, configure logical interface ae0.1. Configure the logical interface to use VLAN bridge encapsulation, and map the logical interface to VLAN 1 and 2. Also, assign an ESI to the logical interface, and enable EVPN multihoming active-standby.

```
[edit interfaces]
user@router# set ae0 unit 1 encapsulation vlan-bridge
user@router# set ae0 unit 1 vlan-id 1
```



```
user@router# set ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
user@router# set ae0 unit 1 esi single-active
```

4. Configure IRB interface irb.1, and assign an IP address to the interface.

```
[edit interfaces]
user@router# set irb unit 1 family inet address 192.0.2.1/24
```

5. Configure an EVPN routing instance named blue. Map the routing instance to VLAN 1, logical interface ae0.1, and IRB interface irb.1. Configure EVPN logical interface ae0.1 for the EVPN routing instance.

```
[edit routing-instances]
user@router# set blue instance-type evpn
user@router# set blue vlan-id 1
user@router# set blue interface ae0.1
user@router# set blue routing-interface irb.1
user@router# set blue protocols evpn interface ae0.1
```

6. Configure a VRF routing instance, and add IRB interface irb.1 to the routing instance.

```
[edit routing-instances]
user@router# set vrf instance-type vrf
user@router# set vrf interface irb.1
```

EVPN Multihoming Active-Active Configuration

CLI Quick Configuration



NOTE: The configurations for PE1 (EX9200) and PE2 (MX Series router) focus on configuring EVPN multihoming active-active and ESIs on logical interfaces. The configurations do not include all EVPN-related configurations for physical interfaces, aggregated Ethernet interfaces, logical interfaces, and routing instances. For a more comprehensive configuration of EVPN multihoming active-active, see [“Example: Configuring EVPN Active-Active Multihoming” on page 147](#). Note that the referenced example shows how to configure ESIs on physical and aggregated Ethernet interfaces.

```
PE1 set interfaces xe-2/0/0 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 unit 1 encapsulation vlan-bridge
set interfaces ae0 unit 1 vlan-id 1
set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
set interfaces ae0 unit 1 esi all-active
set interfaces ae0 unit 2 encapsulation vlan-bridge
set interfaces ae0 unit 2 vlan-id 2
set interfaces ae0 unit 2 esi 00:22:22:33:44:55:66:77:88:99
set interfaces ae0 unit 2 esi all-active
set interfaces irb unit 1 family inet address 192.0.2.1/24
```

```
set interfaces irb unit 2 family inet address 192.0.2.2/24
set routing-instances blue instance-type evpn
set routing-instances blue vlan-id 1
set routing-instances blue interface ae0.1
set routing-instances blue l3-interface irb.1
...
set routing-instances blue protocols evpn interface ae0.1
set routing-instances green instance-type evpn
set routing-instances green vlan-id 2
set routing-instances green interface ae0.2
set routing-instances green l3-interface irb.2
...
set routing-instances green protocols evpn interface ae0.2
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.1
set routing-instances vrf interface irb.2
...
```

```
PE2 set interfaces xe-3/0/2 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 unit 1 encapsulation vlan-bridge
set interfaces ae0 unit 1 vlan-id 1
set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
set interfaces ae0 unit 1 esi all-active
set interfaces irb unit 1 family inet address 192.0.2.1/24
set routing-instances blue instance-type evpn
set routing-instances blue vlan-id 1
set routing-instances blue interface ae0.1
set routing-instances blue routing-interface irb.1
...
set routing-instances blue protocols evpn interface ae0.1
set routing-instances vrf instance-type vrf
set routing-instances vrf interface irb.1
...
```

**Step-by-Step
Procedure**

To configure EVPN multihoming active-active on PE1:

1. Specify an Ethernet interface as a member of aggregated Ethernet interface ae0.

[edit interfaces]

```
user@switch# set xe-2/0/0 gigether-options 802.3ad ae0
```

2. Configure aggregated Ethernet interface ae0 to simultaneously transmit 802.1Q VLAN single-tag and dual-tag frames and to support different types of Ethernet encapsulation at the logical interface level.

[edit interfaces]

```
user@switch# set ae0 flexible-vlan-tagging
```

```
user@switch# set ae0 encapsulation flexible-ethernet-services
```

3. On aggregated Ethernet interface ae0, configure logical interfaces ae0.1 and ae0.2. Configure the logical interfaces to use VLAN bridge encapsulation, and map the logical interfaces to VLANs 1 and 2, respectively. Also, assign an ESI to the logical interfaces, and enable EVPN multihoming active-active.

```
[edit interfaces]
user@switch# set ae0 unit 1 encapsulation vlan-bridge
user@switch# set ae0 unit 1 vlan-id 1
user@switch# set ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
user@switch# set ae0 unit 1 esi all-active
user@switch# set ae0 unit 2 encapsulation vlan-bridge
user@switch# set ae0 unit 2 vlan-id 2
user@switch# set ae0 unit 2 esi 00:22:22:33:44:55:66:77:88:99
user@switch# set ae0 unit 2 esi all-active
```

4. Configure IRB interfaces irb.1 and irb.2, and assign an IP address to each interface.

```
[edit interfaces]
set interfaces irb unit 1 family inet address 192.0.2.1/24
set interfaces irb unit 2 family inet address 192.0.2.2/24
```

5. Configure an EVPN routing instance named blue. Map the routing instance to VLAN 1, logical interface ae0.1, and IRB interface irb.1. Configure logical interface ae0.1 for the EVPN routing instance.

```
[edit routing-instances]
user@switch# set blue instance-type evpn
user@switch# set blue vlan-id 1
user@switch# set blue interface ae0.1
user@switch# set blue l3-interface irb.1
user@switch# set blue protocols evpn interface ae0.1
```

6. Configure an EVPN routing instance named green. Map the routing instance to VLAN 2, logical interface ae0.2, and IRB interface irb.2. Configure logical interface ae0.2 for the EVPN routing instance.

```
[edit routing-instances]
user@switch# set green instance-type evpn
user@switch# set green vlan-id 2
user@switch# set green interface ae0.2
user@switch# set green l3-interface irb.2
user@switch# set green protocols evpn interface ae0.2
```

7. Configure a VRF routing instance, and add IRB interfaces irb.1 and irb.2 to the routing instance.

```
[edit routing-instances]
set vrf instance-type vrf
set vrf interface irb.1
set vrf interface irb.2
```

**Step-by-Step
Procedure**

To configure EVPN multihoming active-active on PE2:

1. Specify Ethernet interface xe-3/0/2 as a member of aggregated Ethernet interface ae0.

```
[edit interfaces]
user@router# set xe-3/0/2 aggregated-ethernet ae0
```

2. Configure aggregated Ethernet interface ae0 to simultaneously transmit 802.1Q VLAN single-tag and dual-tag frames and to support different types of Ethernet encapsulation at the logical interface level.

```
[edit interfaces]
user@router# set ae0 flexible-vlan-tagging
user@router# set ae0 encapsulation flexible-ethernet-services
```

3. On aggregated Ethernet interface ae0, configure logical interface ae0.1. Configure the logical interface to use VLAN bridge encapsulation, and map the logical interface to VLANs 1. Also, assign an ESI to the logical interface, and enable EVPN multihoming active-active.

```
[edit interfaces]
user@router# set ae0 unit 1 encapsulation vlan-bridge
user@router# set ae0 unit 1 vlan-id 1
user@router# set ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
user@router# set ae0 unit 1 esi all-active
```

4. Configure IRB interface irb.1, and assign an IP address to the interface.

```
[edit interfaces]
user@router# set irb unit 1 family inet address 192.0.2.1/24
```

5. Configure an EVPN routing instance named blue. Map the routing instance to VLAN 1, logical interface ae0.1, and IRB interface irb.1. Configure logical interface ae0.1 for the EVPN routing instance.

```
[edit routing-instances]
user@router# set blue instance-type evpn
user@router# set blue vlan-id 1
user@router# set blue interface ae0.1
user@router# set blue routing-interface irb.1
user@router# set blue protocols evpn interface ae0.1
```

6. Configure a VRF routing instance, and add IRB interface irb.1 to the routing instance.

```
[edit routing-instances]
user@router# set vrf instance-type vrf
user@router# set vrf interface irb.1
```

Verification

Verifying that a Logical Interface Has Correct ESI and EVPN Multihoming Mode

Purpose Verify that logical interface ae0.1 is configured with the correct ESI and EVPN multihoming mode.

Action From operational mode, enter the **show interfaces ae0.1** command.

```
user@switch> show interfaces ae0.1
  Logical interface ae0.1 (Index 332) (SNMP ifIndex 706)
  Flags: Up SNMP-Traps 0x20004000 VLAN-Tag [ 0x8100.100 ] Encapsulation:
VLAN-Bridge
  Input packets : 0
  Output packets: 0
  Protocol bridge, MTU: 1522
  Flags: Is-Primary
  Ethernet segment value: 00:11:22:33:44:55:66:77:88:99, Mode: Single-active
```

Meaning The output shows that logical interface ae0.1 is configured with ESI 00:11:22:33:44:55:66:77:88:99 and the EVPN multihoming mode is single-active for multihoming active-standby mode. For a scenario with multihoming active-active mode, the output would display all-active.

Verifying the EVPN Routing Instance Status

Purpose Verify the status of the various elements configured in an EVPN routing instance.

Action From operational mode, enter the **show evpn instance extensive** command.

```

user@switch> show evpn instance extensive
Instance: blue
...
Number of local interfaces: (1 up)
  Interface name  ESI                               Mode          Status
AC-Role
  ae0.1           00:11:22:33:44:55:066:77:88:99  single-active  Up
Root
Number of IRB interfaces: 1 (1 up)
  Interface name  VLAN ID  Status  L3 context
  irb.1           1        Up      vrf
Number of protect interfaces: 0
Number of bridge domains: 1
  VLAN  Domain ID  Intfs / up  IRB intf  Mode          MAC sync  IM route
  label SG sync  IM core nexthop
  1      1          0    0          Local switching
...
Number of ethernet segments: 1
ESI: 00:11:22:33:44:55:066:77:88:99
Status: Resolved by IFL ae0.1
Local interface: ae0.1, Status: Up/Forwarding
Number of remote PEs connected: 1
  Remote PE      MAC label  Aliasing label  Mode
  10.255.0.1     0          300928          all-active
DF Election Algorithm: MOD based
Designated forwarder: 10.255.0.1
Backup forwarder: 10.255.0.2
Last designated forwarder update: Jul 28 13:04:29
Advertised split horizon label: 300128

```

Meaning The output shows that for the EVPN routing instance named blue, the logical interface ae0.1 and IRB interface irb.1 are up and running. It also shows that VLAN 1 and Ethernet segment 00:11:22:33:44:55:066:77:88:99 are properly mapped to the routing instance. It also shows the status of the DF election.

Release History Table

Release	Description
15.1F6	Starting with Junos OS Releases 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI on a logical interface.

Related Documentation

- [EVPN Multihoming Overview on page 119](#)

CHAPTER 7

Configuring MAC Mobility

- [Overview of MAC Mobility on page 207](#)
- [Changing Duplicate MAC Address Detection Settings on page 209](#)

Overview of MAC Mobility

MAC mobility describes the scenario where a host moves from one Ethernet segment to another segment in the EVPN network. Provider Edge (PE) devices discover the host MAC address from its local interfaces or from remote PE devices. When a PE device learns of a new local MAC address, it sends a MAC advertisement route message to other devices in the network. During this time, there are two advertised routes and the PE devices in the EVPN network must decide which of the MAC advertisement messages to use.

To determine the correct MAC address location, PE devices use the MAC mobility extended community field, as defined in RFC 7432, in the MAC advertisement route message. The MAC mobility extended community includes a static flag and a sequence number. The static flag identifies pinned MAC addresses that should not be relocated. The sequence number identifies newer MAC advertisement messages. Starting at 0, the sequence number is incremented for every MAC address mobility event. PE devices running Junos OS apply the following precedence order in determining the MAC advertisement route to use:

1. Advertisement routes with a local pinned MAC address (static MAC address).
2. Advertisement routes with a remote pinned MAC address (static MAC address).
3. Advertisement routes with a higher sequence number.



NOTE: When there are two advertisement route messages for pinned MAC addresses with different routes or two advertisement route messages with the same sequence number, the local device chooses the advertisement route message from the PE device with the lower IP address.

Figure 11 on page 208 illustrates a network where a MAC address is relocated from PE1 to PE2. Before the move, a MAC advertisement route message sent by PE1 has the active route for all PE devices in the network. After the relocation, PE2 learns of the new local MAC address and sends an updated MAC advertisement route message.

Table 8 on page 208 lists the action taken by each PE device based on the two MAC advertisements. The PE device generates a syslog message when it encounters conflicts with a pinned MAC address.



NOTE: Table 8 on page 208 includes use cases with pinned MAC addresses. These use cases do not apply to PE devices that do not support MAC pinning.

Figure 11: Mac Mobility in an EVPN Network

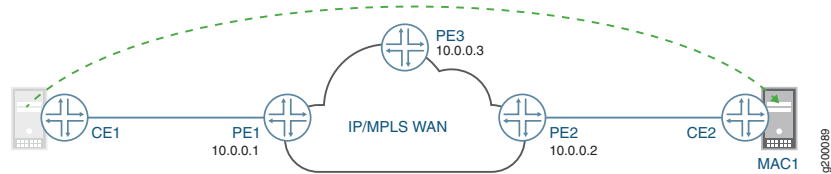


Table 8: MAC Advertisement Routes on PE Devices

MAC Advertisement	PE1	PE2	PE3
PE1: MAC address with a sequence number (n). PE2: MAC address with the sequence number incremented by one (n+1).	Install the remote MAC advertisement route from PE2 because it has a higher sequence number (n+1).	Advertise the local MAC route because it has a higher sequence number (n+1).	Install the remote MAC advertisement route from PE2 because it has a higher sequence number (n+1).
PE1: MAC address with a sequence number (n). PE2: MAC address with the same sequence number (n).	Advertise the local MAC route because PE1 has the lower IP address (10.0.0.1).	Install the remote MAC advertisement route from PE1 because PE1 has the lower IP address (10.0.0.1).	Use the MAC advertisement route from PE1 because PE1 has the lower IP address (10.0.0.1).
PE1: Pinned MAC address with the static bit set. PE2: MAC address and a sequence number (n).	Advertise the local MAC route because it is a pinned MAC address. Generate a syslog message.	Install the remote MAC advertisement route from PE1 because it is a pinned MAC address.	Use the MAC advertisement route from PE1 because it is a pinned MAC address. Generate a syslog message.
PE1: MAC address with a sequence number (n). PE2: Pinned MAC address with the static bit set.	Install the remote the MAC advertisement route from PE2 because it is a pinned MAC address.	Advertise local MAC route because it is a pinned MAC address. Generate a syslog message.	Install the remote MAC advertisement route from PE2 because it is a pinned MAC address.

Table 8: MAC Advertisement Routes on PE Devices (*continued*)

MAC Advertisement	PE1	PE2	PE3
PE1: Pinned MAC address with static bit set.	Advertise the local MAC route because it is a local pinned MAC address.	Advertise the local MAC route because it is a local pinned MAC address.	Use the MAC advertisement route from PE1 because PE1 has the lower IP address (10.0.0.1).
PE2: Pinned MAC address with static bit set.	Generate a syslog message.	Generate a syslog message.	Generate a syslog message.

- Related Documentation**
- [EVPN MAC Pinning Overview](#)
 - [clear evpn duplicate-mac-suppression on page 306](#)
 - [duplicate-mac-detection on page 256](#)

Changing Duplicate MAC Address Detection Settings

When a host is physically moved or when a host is reconfigured on a different Ethernet segment, the PE device sends an updated MAC advertisement route to other PE devices to update their route table. If there is a misconfiguration in the network, MAC advertisement messages oscillate between the different routes causing MAC address flapping. This makes the network more vulnerable and wastes network resources. By default, Junos OS detects and suppresses duplicate MAC addresses. Optionally, you can also configure the length of time that the duplicate MAC address is suppressed. When the PE device encounters duplicate MAC addresses, Junos OS generates a syslog message.

To change the duplicate MAC address detection settings, include the **duplicate-mac-detection** statement at either the **[edit routing-instances routing-instance-name protocols]** hierarchy level or the **[edit logical-systems logical-system-name routing-instances routing-instance-name protocols]** hierarchy level:

```
evpn
  duplicate-mac-detection {
    detection-threshold detection-threshold;
    detection-window seconds;
    auto-recovery-time minutes;
  }
```

You can modify the following options under the **duplicate-mac-detection** statement:

- **detection-window**—The time interval used in detecting a duplicate MAC address. The value can be from 5 through 600 seconds. The default is 180 seconds
- **detection-threshold**—The number of MAC mobility events that are detected for a given MAC address within the **detection-window** before it is identified as a duplicate MAC address. Once the detection threshold is reached, updates for the MAC address are suppressed. The value can be from 2 through 20. The default is 5.
- **auto-recovery-time**—(Optional) The length of time a device suppresses a duplicate MAC address. At the end of this duration, MAC address updates will resume. The value

can be from 5 through 360 minutes. If a value is not specified, then the MAC address continues to be suppressed.



NOTE: To ensure that the mobility advertisements have sufficient time to age out, set an auto-recovery-time greater than the detection-window.

To manually clear the suppression of duplicate MAC address, use the **clear evpn duplicate-mac-suppression** command.

To view MAC duplicate addresses in the EVPN MAC database, use the **show evpn database** command. The following example displays a sample output. The output fields related to duplicate MAC detections are State, Mobility history, and MAC advertisement route status:

```
user@PE1> show evpn database mac-address 00:00:00:00:00:02 extensive
```

Instance: ALPHA

VLAN ID: 100, MAC address: 00:00:00:00:00:02

State: 0x1 <Duplicate-Detected>

Mobility history

	Mobility event time	Type	Source	Seq num
	Aug 03 17:22:28.585619	Local	ge-0/0/2.0	31
	Aug 03 17:22:30.307198	Remote	10.255.0.3	32
	Aug 03 17:22:37.611786	Local	ge-0/0/2.0	33
	Aug 03 17:22:39.289357	Remote	10.255.0.3	34
	Aug 03 17:22:45.609449	Local	ge-0/0/2.0	35

Source: ge-0/0/2.0, Rank: 1, Status: Active

Mobility sequence number: 35 (minimum origin address 10.255.0.2)

Timestamp: Aug 03 17:22:44 (0x5983be54)

State: <Local-MAC-Only Local-To-Remote-Adv-Allowed>

MAC advertisement route status: Not created (duplicate MAC suppression)

IP address: 10.0.0.2

Source: 10.255.0.3, Rank: 2, Status: Inactive

MAC label: 300176

Mobility sequence number: 34 (minimum origin address 10.255.0.3)

Timestamp: Aug 03 17:22:39 (0x5983be4f)

State: <>

MAC advertisement route status: Not created (inactive source)

IP address: 10.0.0.3

- Related Documentation**
- [clear evpn duplicate-mac-suppression on page 306](#)
 - [duplicate-mac-detection on page 256](#)

CHAPTER 8

Configuring EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG

- [Understanding EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG on page 211](#)
- [Example: EVPN-MPLS Interworking With Junos Fusion Enterprise on page 216](#)
- [Example: EVPN-MPLS Interworking With an MC-LAG Topology on page 232](#)

Understanding EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG

Starting with Junos OS Release 17.4R1, you can use Ethernet VPN (EVPN) to extend a Junos Fusion Enterprise or multichassis link aggregation group (MC-LAG) network over an MPLS network to a data center or campus network. With the introduction of this feature, you can now interconnect dispersed campus and data center sites to form a single Layer 2 virtual bridge.

[Figure 12 on page 212](#) shows a Junos Fusion Enterprise topology with two EX9200 switches that serve as aggregation devices (PE2 and PE3) to which the satellite devices are multihomed. The two aggregation devices use an interchassis link (ICL) and the Inter-Chassis Control Protocol (ICCP) protocol from MC-LAG to connect and maintain the Junos Fusion Enterprise topology. PE1 in the EVPN-MPLS environment interworks with PE2 and PE3 in the Junos Fusion Enterprise with MC-LAG.

Figure 12: EVPN-MPLS Interworking with Junos Fusion Enterprise

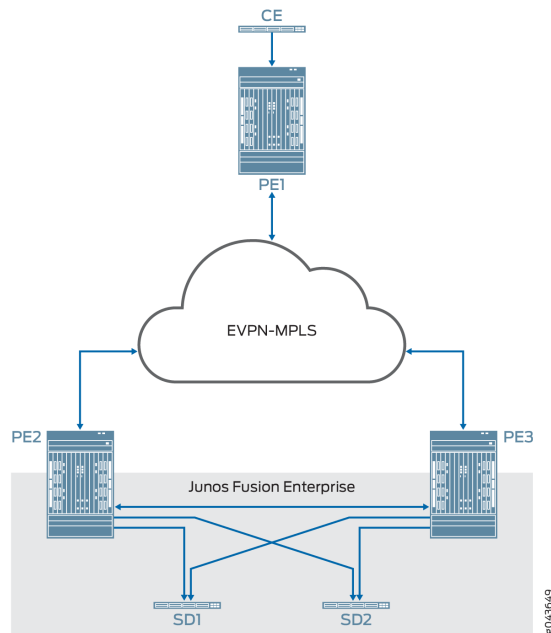
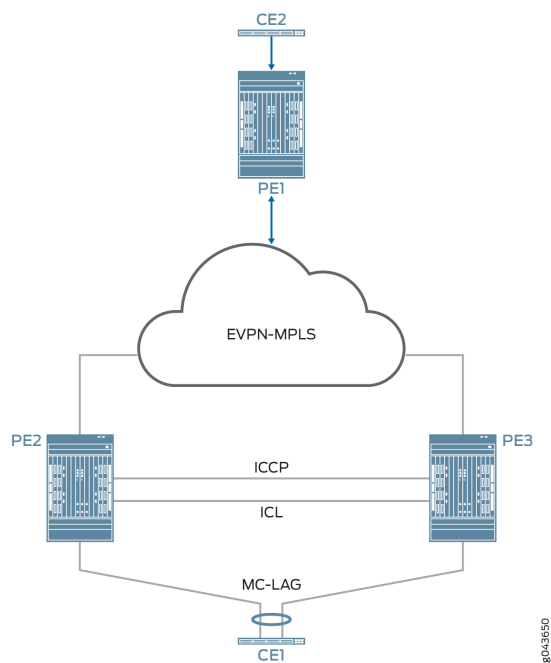


Figure 13 on page 212 shows an MC-LAG topology in which customer edge (CE) device CE1 is multihomed to PE2 and PE3. PE2 and PE3 use an ICL and the ICCP protocol from MC-LAG to connect and maintain the topology. PE1 in the EVPN-MPLS environment interworks with PE2 and PE3 in the MC-LAG environment.

Figure 13: EVPN-MPLS Interworking with MC-LAG



Throughout this topic, [Figure 12 on page 212](#) and [Figure 13 on page 212](#) serve as references to illustrate various scenarios and points.

The use cases depicted in [Figure 12 on page 212](#) and [Figure 13 on page 212](#) require the configuration of both EVPN multihoming in active-active mode and MC-LAG on PE2 and PE3. EVPN with multihoming active-active and MC-LAG have their own forwarding logic for handling traffic, in particular, broadcast, unknown unicast, and multicast (BUM) traffic. At times, the forwarding logic for EVPN with multihoming active-active and MC-LAG contradict each other and causes issues. This topic describes the issues and how the EVPN-MPLS interworking feature resolves these issues.



NOTE:

Other than the EVPN-MPLS interworking-specific implementations described in this topic, EVPN-MPLS, Junos Fusion Enterprise, and MC-LAG offer the same functionality and function the same as the standalone features.

- [BUM Traffic Handling on page 213](#)
- [Split Horizon on page 214](#)
- [MAC Learning on page 215](#)
- [Handling Down Link Between Cascade and Uplink Ports in Junos Fusion Enterprise on page 215](#)
- [Layer 3 Gateway Support on page 216](#)

BUM Traffic Handling

In the use cases shown in [Figure 12 on page 212](#) and [Figure 13 on page 212](#), PE1, PE2, and PE3 are EVPN peers, and PE2 and PE3 are MC-LAG peers. Both sets of peers exchange control information and forward traffic to each other, which causes issues. [Table 9 on page 214](#) outlines the issues that arise, and how Juniper Networks resolves the issues in its implementation of the EVPN-MPLS interworking feature.

Table 9: BUM Traffic: Issues and Resolutions

BUM Traffic Direction	EVPN Interworking with Junos Fusion Enterprise and MC-LAG Logic	Issue	Juniper Networks Implementation Approach
North bound (PE2 receives BUM packet from a locally attached single- or dual-homed interfaces).	<p>PE2 floods BUM packet to the following:</p> <ul style="list-style-type: none"> All locally attached interfaces, including the ICL, for a particular broadcast domain. All remote EVPN peers for which PE2 has received inclusive multicast routes. 	Between PE2 and PE3, there are two BUM forwarding paths—the MC-LAG ICL and an EVPN-MPLS path. The multiple forwarding paths result in packet duplication and loops.	<ul style="list-style-type: none"> BUM traffic is forwarded on the ICL only. Incoming traffic from the EVPN core is not forwarded on the ICL. Incoming traffic from the ICL is not forwarded to the EVPN core.
South bound (PE1 forwards BUM packet to PE2 and PE3).	PE2 and PE3 both receive a copy of the BUM packet and flood the packet out of all of their local interfaces, including the ICL.	PE2 and PE3 both forward the BUM packet out of the ICL, which results in packet duplication and loops.	

Split Horizon

In the use cases shown in [Figure 12 on page 212](#) and [Figure 13 on page 212](#), split horizon prevents multiple copies of a BUM packet from being forwarded to a CE device (satellite device). However, the EVPN-MPLS and MC-LAG split horizon implementations contradict each other, which causes an issue. [Table 10 on page 214](#) explains the issue and how Juniper Networks resolves it in its implementation of the EVPN-MPLS interworking feature.

Table 10: BUM Traffic: Split Horizon-Related Issue and Resolution

BUM Traffic Direction	EVPN Interworking with Junos Fusion Enterprise and MC-LAG Logic	Issue	Juniper Networks Implementation Approach
North bound (PE2 receives BUM packet from a locally attached dual-homed interface).	<ul style="list-style-type: none"> Per EVPN-MPLS forwarding logic: <ul style="list-style-type: none"> Only the designated forwarder (DF) for the Ethernet segment (ES) can forward BUM traffic. The local bias rule, in which the local peer forwards the BUM packet and the remote peer drops it, is not supported. Per MC-LAG forwarding logic, local bias is supported. 	The EVPN-MPLS and MC-LAG forwarding logic contradicts each other and can prevent BUM traffic from being forwarded to the ES.	Support local bias, thereby ignoring the DF and non-DF status of the port for locally switched traffic.

Table 10: BUM Traffic: Split Horizon-Related Issue and Resolution (*continued*)

BUM Traffic Direction	EVPN Interworking with Junos Fusion Enterprise and MC-LAG Logic	Issue	Juniper Networks Implementation Approach
South bound (PE1 forwards BUM packet to PE2 and PE3).	Traffic received from PE1 follows the EVPN DF and non-DF forwarding rules for a multihomed ES.	None.	Not applicable.

MAC Learning

EVPN and MC-LAG use the same method for learning MAC addresses—namely, a PE device learns MAC addresses from its local interfaces and synchronizes the addresses to its peers. However, given that both EVPN and MC-LAG are synchronizing the addresses, an issue arises.

Table 11 on page 215 describes the issue and how the EVPN-MPLS interworking implementation prevents the issue. The use cases shown in Figure 12 on page 212 and Figure 13 on page 212 illustrate the issue. In both use cases, PE1, PE2, and PE3 are EVPN peers, and PE2 and PE3 are MC-LAG peers.

Table 11: MAC Learning: EVPN and MC-LAG Synchronization Issue and Implementation Details

MAC Synchronization Use Case	EVPN Interworking with Junos Fusion Enterprise and MC-LAG Logic	Issue	Juniper Networks Implementation Approach
MAC addresses learned locally on single- or dual-homed interfaces on PE2 and PE3.	<ul style="list-style-type: none"> Between the EVPN peers, MAC addresses are synchronized using the EVPN BGP control plane. Between the MC-LAG peers, MAC addresses are synchronized using the MC-LAG ICCP control plane. 	PE2 and PE3 function as both EVPN peers and MC-LAG peers, which result in these devices having multiple MAC synchronization paths.	<ul style="list-style-type: none"> For PE1: use MAC addresses synchronized by EVPN BGP control plane. For PE2 and PE3: use MAC addresses synchronized by MC-LAG ICCP control plane.
MAC addresses learned locally on single- or dual-homed interfaces on PE1.	Between the EVPN peers, MAC addresses are synchronized using the EVPN BGP control plane.	None.	Not applicable.

Handling Down Link Between Cascade and Uplink Ports in Junos Fusion Enterprise



NOTE: This section applies only to EVPN-MPLS interworking with a Junos Fusion Enterprise.

In the Junos Fusion Enterprise shown in Figure 12 on page 212, assume that aggregation device PE2 receives a BUM packet from PE1 and that the link between the cascade port on PE2 and the corresponding uplink port on satellite device SD1 is down. Regardless of whether the BUM packet is handled by MC-LAG or EVPN multihoming active-active, the

result is the same—the packet is forwarded via the ICL interface to PE3, which forwards it to dual-homed SD1.

To further illustrate how EVPN with multihoming active-active handles this situation with dual-homed SD1, assume that the DF interface resides on PE2 and is associated with the down link and that the non-DF interface resides on PE3. Typically, per EVPN with multihoming active-active forwarding logic, the non-DF interface drops the packet. However, because of the down link associated with the DF interface, PE2 forwards the BUM packet via the ICL to PE3, and the non-DF interface on PE3 forwards the packet to SD1.

Layer 3 Gateway Support

The EVPN-MPLS interworking feature supports the following Layer 3 gateway functionality for extended bridge domains and VLANs:

- Integrated routing and bridging (IRB) interfaces to forward traffic between the extended bridge domains or VLANs.
- Default Layer 3 gateways to forward traffic from a physical (bare-metal) server in an extended bridge domain or VLAN to a physical server or virtual machine in another extended bridge domain or VLAN.

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, you can use Ethernet VPN (EVPN) to extend a Junos Fusion Enterprise or multichassis link aggregation group (MC-LAG) network over an MPLS network to a data center or campus network.

Related Documentation

- [Example: EVPN-MPLS Interworking With Junos Fusion Enterprise on page 216](#)
- [Example: EVPN-MPLS Interworking With an MC-LAG Topology on page 232](#)

Example: EVPN-MPLS Interworking With Junos Fusion Enterprise

This example shows how to use Ethernet VPN (EVPN) to extend a Junos Fusion Enterprise over an MPLS network to a geographically distributed campus or enterprise network.

EVPN-MPLS interworking is supported with a Junos Fusion Enterprise, which is based on a multichassis link aggregation group (MC-LAG) infrastructure to provide redundancy for the EX9200 switches that function as aggregation devices.

The aggregation devices in the Junos Fusion Enterprise are connected to a provider edge (PE) device in an MPLS network. The PE device can be either an MX Series router or an EX9200 switch.

This example shows how to configure the aggregation devices in the Junos Fusion Enterprise and the PE device in the MPLS network to interwork with each other.

- [Requirements on page 217](#)
- [Overview and Topology on page 217](#)
- [Aggregation Device \(PE1 and PE2\) Configuration on page 219](#)
- [PE3 Configuration on page 229](#)

Requirements

This example uses the following hardware and software components:

- Three EX9200 switches:
 - PE1 and PE2, which both function as aggregation devices in the Junos Fusion Enterprise and EVPN BGP peers in the EVPN-MPLS overlay network.
 - PE3, which functions as an EVPN BGP peer in the EVPN-MPLS overlay network.
- The EX9200 switches are running Junos OS Release 17.4R1 or later software.

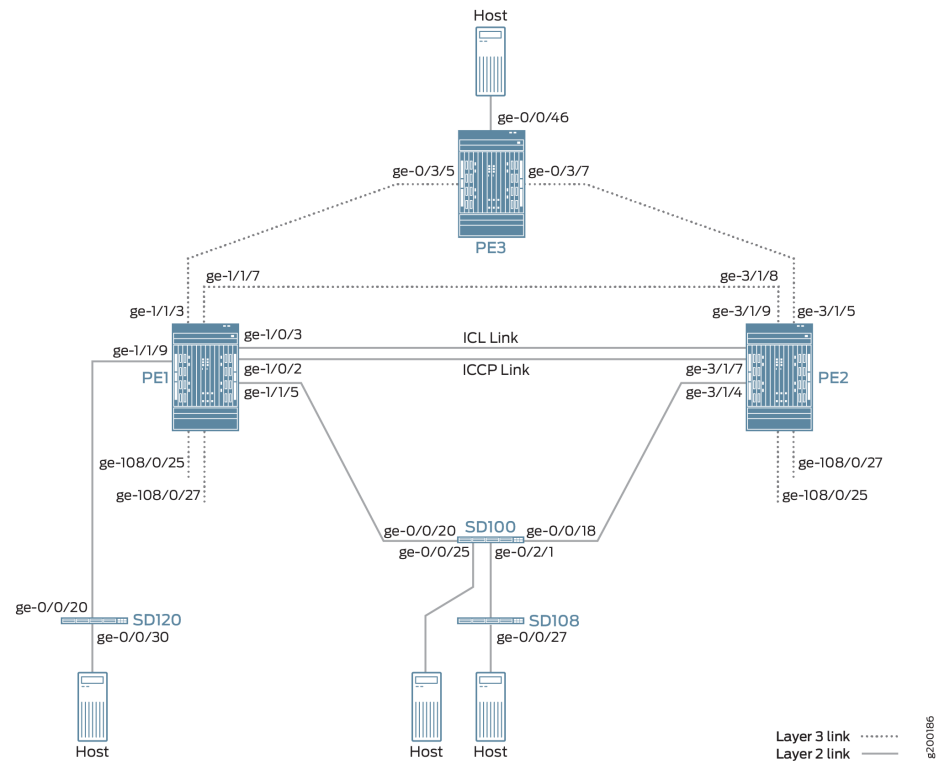


NOTE: Although the Junos Fusion Enterprise includes three satellite devices, this example focuses on the configuration of the PE1, PE2, and PE3. For more information about configuring satellite devices, see *Configuring or Expanding a Junos Fusion Enterprise*.

Overview and Topology

[Figure 14 on page 218](#) shows a Junos Fusion Enterprise with dual aggregation devices PE1 and PE2. The aggregation devices are connected using an interchassis link (ICL) and communicate with each other using the Inter-Chassis Control Protocol (ICCP).

Figure 14: EVPN-MPLS Interworking with Junos Fusion Enterprise



The Junos Fusion Enterprise also includes three satellite devices. Satellite device SD120 is a standalone satellite device that has a single-homed connection to PE1. Satellite devices SD100 and SD108 are included in a cluster named Cluster_100_108. SD100 is the only cluster member with a connection to an aggregation device, in this case, multihomed connections to PE1 and PE2.

The topology in [Figure 14 on page 218](#) also includes PE3, which is positioned at the edge of an MPLS network. PE3 functions as the gateway between the Junos Fusion Enterprise network and a geographically distributed campus or enterprise network. PE1, PE2, and PE3 run EVPN, which enables hosts in the Junos Fusion Enterprise network to communicate with hosts in the campus or enterprise network by way of the intervening MPLS network.

From the perspective of the EVPN-MPLS interworking feature, PE3 functions solely as an EVPN BGP peer, and PE1 and PE2 in the Junos Fusion Enterprise have dual roles:

- Aggregation devices in the Junos Fusion Enterprise.
- EVPN BGP peers in the EVPN-MPLS network.

Because of the dual roles, PE1 and PE2 are configured with Junos Fusion Enterprise, EVPN, BGP, and MPLS attributes.

[Table 12 on page 219](#) outlines key Junos Fusion Enterprise and EVPN (BGP and MPLS) attributes configured on PE1, PE2, and PE3.

Table 12: Key Junos Fusion Enterprise and EVPN (BGP and MPLS) Attributes Configured on PE1, PE2, and PE3

Key Attributes	PE1	PE2	PE3
Junos Fusion Enterprise Attributes			
Interfaces	ICL: ge-1/0/3 ICCP: ge-1/0/2	ICL: ge-3/1/9 ICCP: ge-3/1/7	Not applicable
EVPN-MPLS			
Interfaces	Connection to PE3: ge-1/1/3 Connection to PE2: ge-1/1/7	Connection to PE3: ge-3/1/5 Connection to PE1: ge-3/1/8	Connection to PE1: ge-0/3/5 Connection to PE2: ge-0/3/7
IP addresses	BGP peer address: 10.25.0.1	BGP peer address: 10.25.0.2	BGP peer address: 10.25.0.3
Autonomous system	100	100	100
Virtual switch routing instances	evpn1	evpn1	evpn1

Note the following about the EVPN-MPLS interworking feature and its configuration:

- You must configure Ethernet segment identifiers (ESIs) on the dual-homed extended ports in the Junos Fusion Enterprise. The ESIs enable EVPN to identify the dual-homed extended ports.
- The only type of routing instance that is supported is the virtual switch instance (**set routing-instances *name* instance-type virtual-switch**).
- Only one virtual switch instance is supported with Junos Fusion Enterprise.
- On the aggregation devices in the Junos Fusion Enterprise, you must include the **bgp-peer** configuration statement in the **[edit routing-instances *name* protocols evpn mclag]** hierarchy level. This configuration statement enables the interworking of EVPN-MPLS with Junos Fusion Enterprise on the aggregation devices.
- Address Resolution Protocol (ARP) suppression is not supported.

Aggregation Device (PE1 and PE2) Configuration

To configure aggregation devices PE1 and PE2, perform these tasks.



NOTE: This section focuses on enabling EVPN-MPLS on PE1 and PE2. As a result, the Junos Fusion Enterprise configuration on PE1 and PE2 is performed without the use of the configuration synchronization feature. For information about configuration synchronization, see *Understanding Configuration Synchronization*.

- [PE1: Configuring Junos Fusion Enterprise on page 223](#)
- [PE1: Configuring EVPN-MPLS on page 224](#)
- [PE2: Configuring Junos Fusion Enterprise on page 226](#)
- [PE2: Configuring EVPN-MPLS on page 228](#)

CLI Quick Configuration

PE1: Junos Fusion Enterprise Configuration

```
set interfaces ge-1/1/9 cascade-port
set interfaces ge-1/1/5 cascade-port
set chassis satellite-management fpc 120 cascade-ports ge-1/1/9
set chassis satellite-management cluster Cluster_100_108 cluster-id 2
set chassis satellite-management cluster Cluster_100_108 cascade-ports ge-1/1/5
set chassis satellite-management cluster Cluster_100_108 fpc 100 alias SD100
set chassis satellite-management cluster Cluster_100_108 fpc 100 system-id
    88:e0:f3:1f:3d:50
set chassis satellite-management cluster Cluster_100_108 fpc 108 alias SD108
set chassis satellite-management cluster Cluster_100_108 fpc 108 system-id
    88:e0:f3:1f:c8:d1
set chassis satellite-management cluster Cluster_100_108 fpc 100 member-id 1
set chassis satellite-management cluster Cluster_100_108 fpc 108 member-id 8
set chassis satellite-management upgrade-groups upgrade_120 satellite 120
set chassis satellite-management upgrade-groups upgrade_100 satellite 100
set chassis satellite-management redundancy-groups rg1 redundancy-group-id 2
set chassis satellite-management redundancy-groups chassis-id 1
set chassis satellite-management redundancy-groups rg1 peer-chassis-id 2
    inter-chassis-link ge-1/0/3
set chassis satellite-management redundancy-groups rg1 cluster Cluster_100_108
set interfaces ge-1/0/2 description iccp-link
set interfaces ge-1/0/2 unit 0 family inet address 10.20.20.1/24
set interfaces ge-1/0/3 description icl-link
set interfaces ge-1/0/3 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-1/0/3 unit 0 family ethernet-switching vlan members 100
set switch-options service-id 1
```

PE1: EVPN-MPLS Configuration

```
set interfaces lo0 unit 0 family inet address 10.25.0.1/32
set interfaces ge-1/1/3 unit 0 family inet address 10.0.1.1/30
set interfaces ge-1/1/3 unit 0 family mpls
set interfaces ge-1/1/7 unit 0 family inet address 10.0.3.1/30
set interfaces ge-1/1/7 unit 0 family mpls
set interfaces ge-108/0/25 unit 0 esi 00:01:02:03:04:00:01:02:04:26
set interfaces ge-108/0/25 unit 0 esi all-active
set interfaces ge-108/0/25 unit 0 family ethernet-switching vlan members v100
set interfaces ge-108/0/27 unit 0 esi 00:01:02:03:04:00:01:02:04:28
set interfaces ge-108/0/27 unit 0 esi all-active
```

```

set interfaces ge-108/0/27 unit 0 family ethernet-switching vlan members v100
set routing-options router-id 10.25.0.1
set routing-options autonomous-system 100
set protocols mpls interface lo0.0
set protocols mpls interface ge-1/1/3.0
set protocols mpls interface ge-1/1/7.0
set protocols bgp local-address 10.25.0.1
set protocols bgp peer-as 100
set protocols bgp local-as 100
set protocols bgp group evpn-mes type internal
set protocols bgp group evpn-mes family evpn signaling
set protocols bgp group evpn-mes peer-as 100
set protocols bgp group evpn-mes neighbor 10.25.0.2
set protocols bgp group evpn-mes neighbor 10.25.0.3
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-1/1/3.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-1/1/7.0
set protocols ldp interface lo0.0
set protocols ldp interface ge-1/1/3.0
set protocols ldp interface ge-1/1/7.0
set routing-instances evpn1 instance-type virtual-switch
set routing-instances evpn1 interface ge-108/0/25.0
set routing-instances evpn1 interface ge-108/0/27.0
set routing-instances evpn1 interface ge-1/0/3.0
set routing-instances evpn1 route-distinguisher 10.25.0.1:1
set routing-instances evpn1 vrf-target target:100:1
set routing-instances evpn1 protocols evpn label-allocation per-instance
set routing-instances evpn1 protocols evpn extended-vlan-list 100
set routing-instances evpn1 protocols evpn mclag bgp-peer 10.25.0.2
set routing-instances evpn1 switch-options service-id 2
set routing-instances evpn1 vlans v100 vlan-id 100

```

PE2: Junos Fusion Enterprise Configuration

```

set interfaces ge-3/1/4 cascade-port
set chassis satellite-management cluster Cluster_100_108 cluster-id 2
set chassis satellite-management cluster Cluster_100_108 cascade-ports ge-3/1/4
set chassis satellite-management cluster Cluster_100_108 fpc 100 alias SD100
set chassis satellite-management cluster Cluster_100_108 fpc 100 system-id
88:e0:f3:1f:3d:50
set chassis satellite-management cluster Cluster_100_108 fpc 108 alias SD108
set chassis satellite-management cluster Cluster_100_108 fpc 108 system-id
88:e0:f3:1f:c8:d1
set chassis satellite-management cluster Cluster_100_108 fpc 100 member-id 1
set chassis satellite-management cluster Cluster_100_108 fpc 108 member-id 8
set chassis satellite-management upgrade-groups upgrade_100 satellite 100
set chassis satellite-management redundancy-groups rg1 redundancy-group-id 2
set chassis satellite-management redundancy-groups chassis-id 2
set chassis satellite-management redundancy-groups rg1 peer-chassis-id 1
inter-chassis-link ge-3/1/9
set chassis satellite-management redundancy-groups rg1 cluster Cluster_100_108
set interfaces ge-3/1/7 description iccp-link
set interfaces ge-3/1/7 unit 0 family inet address 10.20.20.2/24
set interfaces ge-3/1/9 description icl-link

```

```
set interfaces ge-3/1/9 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-3/1/9 unit 0 family ethernet-switching vlan members 100
set switch-options service-id 1
```

PE2: EVPN-MPLS Configuration

```
set interfaces lo0 unit 0 family inet address 10.25.0.2/32
set interfaces ge-3/1/5 unit 0 family inet address 10.0.4.2/30
set interfaces ge-3/1/5 unit 0 family mpls
set interfaces ge-3/1/8 unit 0 family inet address 10.0.3.2/30
set interfaces ge-3/1/8 unit 0 family mpls
set interfaces irb unit 0 family inet address 10.5.5.1/24 virtual-gateway-address 10.5.5.5
set interfaces ge-108/0/25 unit 0 esi 00:01:02:03:04:00:01:02:04:26
set interfaces ge-108/0/25 unit 0 esi all-active
set interfaces ge-108/0/25 unit 0 family ethernet-switching vlan members v100
set interfaces ge-108/0/27 unit 0 esi 00:01:02:03:04:00:01:02:04:28
set interfaces ge-108/0/27 unit 0 esi all-active
set interfaces ge-108/0/27 unit 0 family ethernet-switching vlan members v100
set routing-options router-id 10.25.0.2
set routing-options autonomous-system 100
set protocols mpls interface lo0.0
set protocols mpls interface ge-3/1/5.0
set protocols mpls interface ge-3/1/8.0
set protocols bgp local-address 10.25.0.2
set protocols bgp peer-as 100
set protocols bgp local-as 100
set protocols bgp group evpn-mes type internal
set protocols bgp group evpn-mes family evpn signaling
set protocols bgp group evpn-mes peer-as 100
set protocols bgp group evpn-mes neighbor 10.25.0.1
set protocols bgp group evpn-mes neighbor 10.25.0.3
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-3/1/5.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-3/1/8.0
set protocols ldp interface lo0.0
set protocols ldp interface ge-3/1/5.0
set protocols ldp interface ge-3/1/8.0
set routing-instances evpn1 instance-type virtual-switch
set routing-instances evpn1 interface ge-108/0/25.0
set routing-instances evpn1 interface ge-108/0/27.0
set routing-instances evpn1 interface ge-3/1/9.0
set routing-instances evpn1 route-distinguisher 10.25.0.2:1
set routing-instances evpn1 vrf-target target:100:1
set routing-instances evpn1 protocols evpn label-allocation per-instance
set routing-instances evpn1 protocols evpn extended-vlan-list 100
set routing-instances evpn1 protocols evpn mlag bgp-peer 10.25.0.1
set routing-instances evpn1 switch-options service-id 2
set routing-instances evpn1 vlans v100 vlan-id 100
set routing-instances evpn1 vlans v100 l3-interface irb.0
set routing-instances evpn1 vlans v100 no-arp-suppression
```

PE1: Configuring Junos Fusion Enterprise

Step-by-Step Procedure

1. Configure the cascade ports.


```
[edit]
user@switch# set interfaces ge-1/1/9 cascade-port
user@switch# set interfaces ge-1/1/5 cascade-port
```
2. Configure the FPC slot ID for standalone satellite device SD120 and map it to a cascade port.


```
[edit]
user@switch# set chassis satellite-management fpc 120 cascade-ports ge-1/1/9
```
3. Create a satellite device cluster, and assign a name and a cluster ID to it.


```
[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108 cluster-id
2
```
4. Define the cascade ports associated with the satellite device cluster.


```
[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108
cascade-ports ge-1/1/5
user@switch# set chassis satellite-management cluster Cluster_100_108
cascade-ports ge-1/1/9
```
5. Configure the FPC slot ID number, and map it to the MAC address of satellite devices SD100 and SD108, respectively.


```
[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 100
alias SD100
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 100
system-id 88:e0:f3:1f:3d:50
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 108
alias SD108
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 108
system-id 88:e0:f3:1f:c8:d1
```
6. Assign a member ID to each satellite device in the satellite device cluster.


```
[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 100
member-id 1
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 108
member-id 8
```

7. Create two satellite software upgrade groups—one that includes satellite device SD120 and another that includes satellite device SD100.

```
[edit]
user@switch# set chassis satellite-management upgrade-groups upgrade_120
  satellite 120
user@switch# set chassis satellite-management upgrade-groups upgrade_100
  satellite 100
```

8. Create and configure a redundancy group, which includes the aggregation devices and satellite devices in Cluster_100_108.

```
[edit]
user@switch# set chassis satellite-management redundancy-groups rg1
  redundancy-group-id 2
user@switch# set chassis satellite-management redundancy-groups chassis-id 1
user@switch# set chassis satellite-management redundancy-groups rg1
  peer-chassis-id 2 inter-chassis-link ge-1/0/3
user@switch# set chassis satellite-management redundancy-groups rg1 cluster
  Cluster_100_108
```

9. Configure the ICL and ICCP links.

```
[edit]
user@switch# set interfaces ge-1/0/2 description iccp-link
user@switch# set interfaces ge-1/0/2 unit 0 family inet address 10.20.20.1/24
user@switch# set interfaces ge-1/0/3 description icl-link
user@switch# set interfaces ge-1/0/3 unit 0 family ethernet-switching
  interface-mode trunk
user@switch# set interfaces ge-1/0/3 unit 0 family ethernet-switching vlan members
  100
user@switch# set switch-options service-id 1
```



NOTE: While this step shows the configuration of interface ge-1/0/2, which is designated as the ICCP interface, it does not show how to configure the ICCP attributes on interface ge-1/0/2. By default, ICCP is automatically provisioned in a Junos Fusion Enterprise using dual aggregation devices. For more information about the automatic provisioning of ICCP, see *Configuring or Expanding a Junos Fusion Enterprise*.

PE1: Configuring EVPN-MPLS

Step-by-Step Procedure

1. Configure the loopback interface and the interfaces connected to the other PE devices.

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 10.25.0.1/32
user@switch# set interfaces ge-1/1/3 unit 0 family inet address 10.0.1.1/30
user@switch# set interfaces ge-1/1/3 unit 0 family mpls
```



```

user@switch# set interfaces ge-1/1/7 unit 0 family inet address 10.0.3.1/30
user@switch# set interfaces ge-1/1/7 unit 0 family mpls

```

2. Configure the extended ports with EVPN multihoming in active-active mode, an ESI, and map the ports to VLAN v100..

```

[edit]
user@switch# set interfaces ge-108/0/25 unit 0 esi 00:01:02:03:04:00:01:02:04:26
user@switch# set interfaces ge-108/0/25 unit 0 esi all-active
user@switch# set interfaces ge-108/0/25 unit 0 family ethernet-switching vlan
members v100
user@switch# set interfaces ge-108/0/27 unit 0 esi 00:01:02:03:04:00:01:02:04:28
user@switch# set interfaces ge-108/0/27 unit 0 esi all-active
user@switch# set interfaces ge-108/0/27 unit 0 family ethernet-switching vlan
members v100

```

3. Assign a router ID and the autonomous system in which PE1, PE2, and PE3 reside.

```

[edit]
user@switch# set routing-options router-id 10.25.0.1
user@switch# set routing-options autonomous-system 100

```

4. Enable MPLS on the loopback interface and interfaces ge-1/1/3.0 and ge-1/1/7.0.

```

[edit]
user@switch# set protocols mpls interface lo0.0
user@switch# set protocols mpls interface ge-1/1/3.0
user@switch# set protocols mpls interface ge-1/1/7.0

```

5. Configure an IBGP overlay that includes PE1, PE2, and PE3.

```

[edit]
user@switch# set protocols bgp local-address 10.25.0.1
user@switch# set protocols bgp peer-as 100
user@switch# set protocols bgp local-as 100
user@switch# set protocols bgp group evpn-mes type internal
user@switch# set protocols bgp group evpn-mes family evpn signaling
user@switch# set protocols bgp group evpn-mes peer-as 100
user@switch# set protocols bgp group evpn-mes neighbor 10.25.0.2
user@switch# set protocols bgp group evpn-mes neighbor 10.25.0.3

```

6. Configure OSPF as the internal routing protocol for EVPN by specifying an area ID and interfaces on which EVPN-MPLS is enabled.

```

[edit]
user@switch# set protocols ospf traffic-engineering
user@switch# set protocols ospf area 0.0.0.0 interface ge-1/1/3.0
user@switch# set protocols ospf area 0.0.0.0 interface lo0.0
user@switch# set protocols ospf area 0.0.0.0 interface fxp0.0 disable
user@switch# set protocols ospf area 0.0.0.0 interface ge-1/1/7.0

```

7. Configure the Label Distribution Protocol (LDP) on the loopback interface and the interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ldp interface lo0.0
user@switch# set protocols ldp interface ge-1/1/3.0
user@switch# set protocols ldp interface ge-1/1/7.0
```

8. Configure a virtual switch routing instance for VLAN v100, and include the interfaces and other entities associated with the VLAN.

```
[edit]
user@switch# set routing-instances evpn1 instance-type virtual-switch
user@switch# set routing-instances evpn1 interface ge-108/0/25.0
user@switch# set routing-instances evpn1 interface ge-108/0/27.0
user@switch# set routing-instances evpn1 interface ge-1/0/3.0
user@switch# set routing-instances evpn1 route-distinguisher 10.25.0.1:1
user@switch# set routing-instances evpn1 vrf-target target:100:1
user@switch# set routing-instances evpn1 protocols evpn label-allocation
per-instance
user@switch# set routing-instances evpn1 protocols evpn extended-vlan-list 100
user@switch# set routing-instances evpn1 protocols evpn mclag bgp-peer 10.25.0.2
user@switch# set routing-instances evpn1 switch-options service-id 2
user@switch# set routing-instances evpn1 vlans v100 vlan-id 100
```

PE2: Configuring Junos Fusion Enterprise

Step-by-Step Procedure

1. Configure the cascade port.

```
[edit]
user@switch# set interfaces ge-3/1/4 cascade-port
```

2. Create a satellite device cluster, and assign a name and a cluster ID to it.

```
[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108 cluster-id
2
```

3. Define the cascade port associated with the satellite device cluster.

```
[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108
cascade-ports ge-3/1/4
```

4. Configure the FPC slot ID number, and map it to the MAC address of satellite devices SD100 and SD108, respectively.

```
[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 100
alias SD100
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 100
system-id 88:e0:f3:1f:3d:50
```

```

user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 108
alias SD108
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 108
system-id 88:e0:f3:1f:c8:d1

```

5. Assign a member ID to each satellite device in the satellite device cluster.

```

[edit]
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 100
member-id 1
user@switch# set chassis satellite-management cluster Cluster_100_108 fpc 108
member-id 8

```

6. Create a satellite software upgrade group that includes satellite device SD100.

```

[edit]
user@switch# set chassis satellite-management upgrade-groups upgrade_100
satellite 100

```

7. Create and configure a redundancy group, which includes the aggregation devices and satellite devices in Cluster_100_108.

```

[edit]
user@switch# set chassis satellite-management redundancy-groups rg1
redundancy-group-id 2
user@switch# set chassis satellite-management redundancy-groups chassis-id 2
user@switch# set chassis satellite-management redundancy-groups rg1
peer-chassis-id linter-chassis-link ge-3/1/9
user@switch# set chassis satellite-management redundancy-groups rg1 cluster
Cluster_100_108

```

8. Configure the ICL and ICCP links.

```

[edit]
user@switch# set interfaces ge-3/1/7 description iccp-link
user@switch# set interfaces ge-3/1/7 unit 0 family inet address 10.20.20.2/24
user@switch# set interfaces ge-3/1/9 description icl-link
user@switch# set interfaces ge-3/1/9 unit 0 family ethernet-switching
interface-mode trunk
user@switch# set interfaces ge-3/1/9 unit 0 family ethernet-switching vlan members
100
user@switch# set switch-options service-id 1

```



NOTE: While this step shows the configuration of interface ge-3/1/7, which is designated as the ICCP interface, it does not show how to configure the ICCP attributes on interface ge-3/1/7. By default, ICCP is automatically provisioned in a Junos Fusion Enterprise using dual aggregation devices. For more information about the automatic provisioning of ICCP, see *Configuring or Expanding a Junos Fusion Enterprise*.

PE2: Configuring EVPN-MPLS

Step-by-Step Procedure

1. Configure the loopback interface, the interfaces connected to the other PE devices, and an IRB interface that is also configured as a default Layer 3 gateway.

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 10.25.0.2/32
user@switch# set interfaces ge-3/1/5 unit 0 family inet address 10.0.4.2/30
user@switch# set interfaces ge-3/1/5 unit 0 family mpls
user@switch# set interfaces ge-3/1/8 unit 0 family inet address 10.0.3.2/30
user@switch# set interfaces ge-3/1/8 unit 0 family mpls
user@switch# set interfaces irb unit 0 family inet address 10.5.5.1/24
virtual-gateway-address 10.5.5.5
```

2. Configure the extended ports with EVPN multihoming in active-active mode, an ESI, and map the ports to VLAN v100..

```
[edit]
user@switch# set interfaces ge-108/0/25 unit 0 esi 00:01:02:03:04:00:01:02:04:26
user@switch# set interfaces ge-108/0/25 unit 0 esi all-active
user@switch# set interfaces ge-108/0/25 unit 0 family ethernet-switching vlan
members v100
user@switch# set interfaces ge-108/0/27 unit 0 esi 00:01:02:03:04:00:01:02:04:28
user@switch# set interfaces ge-108/0/27 unit 0 esi all-active
user@switch# set interfaces ge-108/0/27 unit 0 family ethernet-switching vlan
members v100
```

3. Assign a router ID and the autonomous system in which PE1, PE2, and PE3 reside.

```
[edit]
user@switch# set routing-options router-id 10.25.0.2
user@switch# set routing-options autonomous-system 100
```

4. Enable MPLS on the loopback interface and interfaces ge-3/1/5.0 and ge-3/1/8.0.

```
[edit]
user@switch# set protocols mpls interface lo0.0
user@switch# set protocols mpls interface ge-3/1/5.0
user@switch# set protocols mpls interface ge-3/1/8.0
```

5. Configure an IBGP overlay that includes PE1, PE2, and PE3.

```
[edit]
user@switch# set protocols bgp local-address 10.25.0.2
user@switch# set protocols bgp peer-as 100
user@switch# set protocols bgp local-as 100
user@switch# set protocols bgp group evpn-mes type internal
user@switch# set protocols bgp group evpn-mes family evpn signaling
user@switch# set protocols bgp group evpn-mes peer-as 100
user@switch# set protocols bgp group evpn-mes neighbor 10.25.0.1
user@switch# set protocols bgp group evpn-mes neighbor 10.25.0.3
```

6. Configure OSPF as the internal routing protocol for EVPN by specifying an area ID and interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ospf traffic-engineering
user@switch# set protocols ospf area 0.0.0.0 interface ge-3/1/5.0
user@switch# set protocols ospf area 0.0.0.0 interface lo0.0
user@switch# set protocols ospf area 0.0.0.0 interface fxp0.0 disable
user@switch# set protocols ospf area 0.0.0.0 interface ge-3/1/8.0
```

7. Configure the LDP on the loopback interface and the interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ldp interface lo0.0
user@switch# set protocols ldp interface ge-3/1/5.0
user@switch# set protocols ldp interface ge-3/1/8.0
```

8. Configure a virtual switch routing instance for VLAN v100, and include the interfaces and other entities associated with the VLAN.

```
[edit]
user@switch# set routing-instances evpn1 instance-type virtual-switch
user@switch# set routing-instances evpn1 interface ge-108/0/25.0
user@switch# set routing-instances evpn1 interface ge-108/0/27.0
user@switch# set routing-instances evpn1 interface ge-3/1/9.0
user@switch# set routing-instances evpn1 route-distinguisher 10.25.0.2:1
user@switch# set routing-instances evpn1 vrf-target target:100:1
user@switch# set routing-instances evpn1 protocols evpn label-allocation
per-instance
user@switch# set routing-instances evpn1 protocols evpn extended-vlan-list 100
user@switch# set routing-instances evpn1 protocols evpn mclag bgp-peer 10.25.0.1
user@switch# set routing-instances evpn1 switch-options service-id 2
user@switch# set routing-instances evpn1 vlans v100 vlan-id 100
user@switch# set routing-instances evpn1 vlans v100 l3-interface irb.0
user@switch# set routing-instances evpn1 vlans v100 no-arp-suppression
```

PE3 Configuration

CLI Quick Configuration

PE3: EVPN-MPLS Configuration

```
set interfaces lo0 unit 0 family inet address 10.25.0.3/32
set interfaces ge-0/3/5 unit 0 family inet address 10.0.1.2/30
set interfaces ge-0/3/5 unit 0 family mpls
set interfaces ge-0/3/7 unit 0 family inet address 10.0.4.1/30
set interfaces ge-0/3/7 unit 0 family mpls
set interfaces ge-0/0/46 unit 0 esi 00:01:02:03:04:00:01:02:04:12
set interfaces ge-0/0/46 unit 0 esi all-active
set interfaces ge-0/0/46 unit 0 family ethernet-switching vlan members 100
set routing-options router-id 10.25.0.3
set routing-options autonomous-system 100
set routing-options forwarding-table export evpn-pplb
set policy-options policy-statement evpn-pplb from protocol evpn
```

```
set policy-options policy-statement evpn-pplb then load-balance per-packet
set protocols mpls interface lo0.0
set protocols mpls interface ge-0/3/5.0
set protocols mpls interface ge-0/3/7.0
set protocols bgp local-address 10.25.0.3
set protocols bgp peer-as 100
set protocols bgp local-as 100
set protocols bgp group evpn-mes type internal
set protocols bgp group evpn-mes family evpn signaling
set protocols bgp group evpn-mes peer-as 100
set protocols bgp group evpn-mes neighbor 10.25.0.2
set protocols bgp group evpn-mes neighbor 10.25.0.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface ge-0/3/5.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/3/7.0
set protocols ldp interface lo0.0
set protocols ldp interface ge-0/3/5.0
set protocols ldp interface ge-0/3/7.0
set routing-instances evpn1 instance-type virtual-switch
set routing-instances evpn1 interface ge-0/0/46.0
set routing-instances evpn1 route-distinguisher 10.25.0.3:1
set routing-instances evpn1 vrf-target target:100:1
set routing-instances evpn1 protocols evpn label-allocation per-instance
set routing-instances evpn1 protocols evpn extended-vlan-list 100
set routing-instances evpn1 switch-options service-id 2
set routing-instances evpn1 vlans v100 vlan-id 100
```

PE3: Configuring EVPN-MPLS

Step-by-Step Procedure

1. Configure the interfaces on EVPN-MPLS interworking occurs.

[edit]
user@switch# set interfaces lo0 unit 0 family inet address 10.25.0.3/32
user@switch# set interfaces ge-0/3/5 unit 0 family inet address 10.0.1.2/30
user@switch# set interfaces ge-0/3/5 unit 0 family mpls
user@switch# set interfaces ge-0/3/7 unit 0 family inet address 10.0.4.1/30
user@switch# set interfaces ge-0/3/7 unit 0 family mpls
2. Configure interface ge-0/0/46 with EVPN multihoming in active-active mode, an ESI, and map the ports to VLAN v100..

[edit]
user@switch# set interfaces ge-0/0/46 unit 0 esi 00:01:02:03:04:00:01:02:04:12
user@switch# set interfaces ge-0/0/46 unit 0 esi all-active
user@switch# set interfaces ge-0/0/46 unit 0 family ethernet-switching vlan members 100
3. Assign a router ID and the autonomous system in which the PE1, PE2, and PE3 reside.

[edit]
user@switch# set routing-options router-id 10.25.0.2
user@switch# set routing-options autonomous-system 100

4. Enable per-packet load-balancing for EVPN routes when EVPN multihoming active-active mode is used.

```
[edit]
user@switch# set routing-options forwarding-table export evpn-pplb
user@switch# set policy-options policy-statement evpn-pplb from protocol evpn
user@switch# set policy-options policy-statement evpn-pplb then load-balance
per-packet
```

5. Enable MPLS on the loopback interface and interfaces ge-0/3/5.0 and ge-0/3/7.0.

```
[edit]
user@switch# set protocols mpls interface lo0.0
user@switch# set protocols mpls interface ge-0/3/5.0
user@switch# set protocols mpls interface ge-0/3/7.0
```

6. Configure an IBGP overlay that includes PE1, PE2, and PE3.

```
[edit]
user@switch# set protocols bgp local-address 10.25.0.3
user@switch# set protocols bgp peer-as 100
user@switch# set protocols bgp local-as 100
user@switch# set protocols bgp group evpn-mes type internal
user@switch# set protocols bgp group evpn-mes family evpn signaling
user@switch# set protocols bgp group evpn-mes peer-as 100
user@switch# set protocols bgp group evpn-mes neighbor 10.25.0.2
user@switch# set protocols bgp group evpn-mes neighbor 10.25.0.1
```

7. Configure OSPF as the internal routing protocol for EVPN by specifying an area ID and interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ospf traffic-engineering
user@switch# set protocols ospf area 0.0.0.0 interface ge-0/3/5.0
user@switch# set protocols ospf area 0.0.0.0 interface lo0.0
user@switch# set protocols ospf area 0.0.0.0 interface fxp0.0 disable
user@switch# set protocols ospf area 0.0.0.0 interface ge-0/3/7.0
```

8. Configure the LDP on the loopback interface and the interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ldp interface lo0.0
user@switch# set protocols ldp interface ge-0/3/5.0
user@switch# set protocols ldp interface ge-0/3/7.0
```

9. Configure a virtual switch routing instance for VLAN v100, and include the interfaces and other entities associated with the VLAN.

```
[edit]
user@switch# set routing-instances evpn1 instance-type virtual-switch
user@switch# set routing-instances evpn1 interface ge-0/0/46.0
user@switch# set routing-instances evpn1 route-distinguisher 10.25.0.3:1
user@switch# set routing-instances evpn1 vrf-target target:100:1
```

```
user@switch# set routing-instances evpn1 protocols evpn label-allocation
per-instance
user@switch# set routing-instances evpn1 protocols evpn extended-vlan-list 100
user@switch# set routing-instances evpn1 switch-options service-id 2
user@switch# set routing-instances evpn1 vlans v100 vlan-id 100
```

- Related Documentation**
- [Understanding EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG on page 211](#)

Example: EVPN-MPLS Interworking With an MC-LAG Topology

This example shows how to use Ethernet VPN (EVPN) to extend a multichassis link aggregation (MC-LAG) network over an MPLS network to a data center network or geographically distributed campus network.

EVPN-MPLS interworking is supported with an MC-LAG topology in which two MX Series routers, two EX9200 switches, or a mix of the two Juniper Networks devices function as MC-LAG peers, which use the Inter-Chassis Control Protocol (ICCP) and an interchassis link (ICL) to connect and maintain the topology. The MC-LAG peers are connected to a provider edge (PE) device in an MPLS network. The PE device can be either an MX Series router or an EX9200 switch.

This example shows how to configure the MC-LAG peers and PE device in the MPLS network to interwork with each other.

- [Requirements on page 232](#)
- [Overview and Topology on page 233](#)
- [PE1 and PE2 Configuration on page 235](#)
- [PE3 Configuration on page 247](#)

Requirements

This example uses the following hardware and software components:

- Three EX9200 switches:
 - PE1 and PE2, which both function as MC-LAG peers in the MC-LAG topology and EVPN BGP peers in the EVPN-MPLS overlay network.
 - PE3, which functions as an EVPN BGP peer in the EVPN-MPLS overlay network.
- The EX9200 switches are running Junos OS Release 17.4R1 or later software.

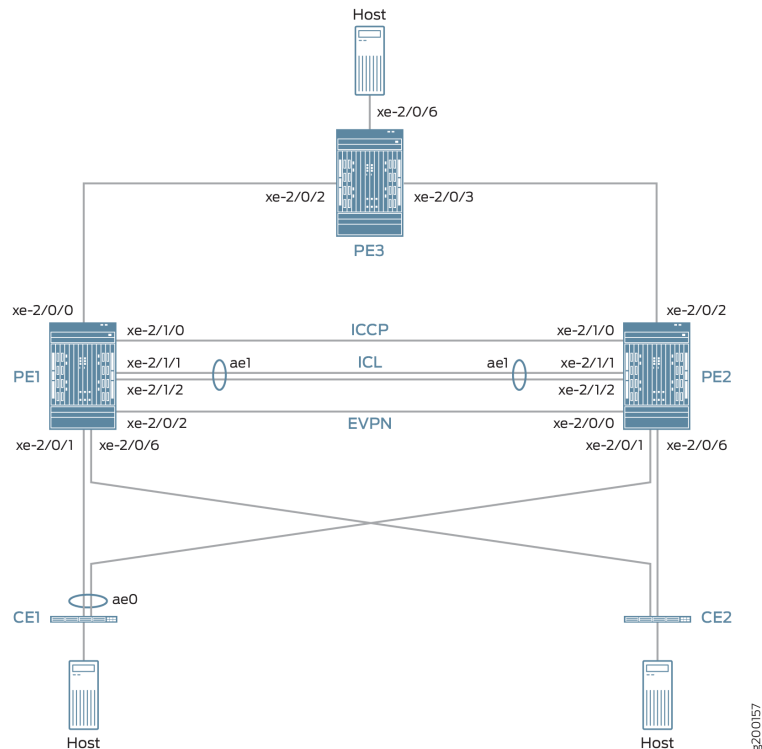


NOTE: Although the MC-LAG topology includes two customer edge (CE) devices, this example focuses on the configuration of the PE1, PE2, and PE3.

Overview and Topology

Figure 13 on page 212 shows an MC-LAG topology with provider edge devices PE1 and PE2 that are configured as MC-LAG peers. The MC-LAG peers exchange control information over an ICCP link and data traffic over an ICL. In this example, the ICL is an aggregated Ethernet interface that is comprised of two interfaces.

Figure 15: EVPN-MPLS Interworking With an MC-LAG Topology



The topology in Figure 13 on page 212 also includes CE devices CE1 and CE2, which are both multihomed to each PE device. The links between CE1 and the two PE devices are bundled as an aggregated Ethernet interface on which MC-LAG in active-active mode is configured.

The topology in Figure 13 on page 212 also includes PE3 at the edge of an MPLS network. PE3 functions as the gateway between the MC-LAG network and either a data center or a geographically distributed campus network. PE1, PE2, and PE3 run EVPN, which enables hosts in the MC-LAG network to communicate with hosts in the data center or other campus network by way of an intervening MPLS network.

From the perspective of the EVPN-MPLS interworking feature, PE3 functions solely as an EVPN BGP peer, and PE1 and PE2 in the MC-LAG topology have dual roles:

- MC-LAG peers in the MC-LAG network.
- EVPN BGP peers in the EVPN-MPLS network.

Because of the dual roles, PE1 and PE2 are configured with MC-LAG, EVPN, BGP, and MPLS attributes.

[Table 12 on page 219](#) outlines key MC-LAG and EVPN (BGP and MPLS) attributes configured on PE1, PE2, and PE3.

Table 13: Key MC-LAG and EVPN (BGP and MPLS) Attributes Configured on PE1, PE2, and PE3

Key Attributes	PE1	PE2	PE3
MC-LAG Attributes			
Interfaces	ICL: aggregated Ethernet interface ae1, which is comprised of xe-2/1/1 and xe-2/1/2 ICCP: xe-2/1/0	ICL: aggregated Ethernet interface ae1, which is comprised of xe-2/1/1 and xe-2/1/2 ICCP: xe-2/1/0	Not applicable
EVPN-MPLS			
Interfaces	Connection to PE3: xe-2/0/0 Connection to PE2: xe-2/0/2	Connection to PE3: xe-2/0/2 Connection to PE1: xe-2/0/0	Connection to PE1: xe-2/0/2 Connection to PE2: xe-2/0/3
IP addresses	BGP peer address: 198.51.100.1	BGP peer address: 198.51.100.2	BGP peer address: 198.51.100.3
Autonomous system	65000	65000	65000
Virtual switch routing instances	evpn1, evpn2, evpn3	evpn1, evpn2, evpn3	evpn1, evpn2, evpn3

Note the following about the EVPN-MPLS interworking feature and its configuration:

- You must configure Ethernet segment identifiers (ESIs) on the dual-homed interfaces in the MC-LAG topology. The ESIs enable EVPN to identify the dual-homed interfaces.
- The only type of routing instance that is supported is the virtual switch instance (**set routing-instances *name* instance-type virtual-switch**).
- On the MC-LAG peers, you must include the **bgp-peer** configuration statement in the **[edit routing-instances *name* protocols evpn mclag]** hierarchy level. This configuration statement enables the interworking of EVPN-MPLS with MC-LAG on the MC-LAG peers.
- Address Resolution Protocol (ARP) suppression is not supported.

PE1 and PE2 Configuration

To configure PE1 and PE2, perform these tasks:

- [PE1: Configuring MC-LAG on page 239](#)
- [PE1: Configuring EVPN-MPLS on page 241](#)
- [PE2: Configuring MC-LAG on page 243](#)
- [PE2: Configuring EVPN-MPLS on page 245](#)

CLI Quick Configuration

PE1: MC-LAG Configuration

```
set chassis aggregated-devices ethernet device-count 3
set interfaces xe-2/0/1 gigether-options 802.3ad ae0
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:11:11:11:11
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 aggregated-ether-options mc-ae mc-ae-id 1
set interfaces ae0 aggregated-ether-options mc-ae redundancy-group 2
set interfaces ae0 aggregated-ether-options mc-ae chassis-id 0
set interfaces ae0 aggregated-ether-options mc-ae mode active-active
set interfaces ae0 aggregated-ether-options mc-ae status-control active
set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
set interfaces ae0 unit 1 esi all-active
set interfaces ae0 unit 1 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 1 family ethernet-switching vlan members 1
set interfaces ae0 unit 2 esi 00:11:11:11:11:11:11:11:11:11
set interfaces ae0 unit 2 esi all-active
set interfaces ae0 unit 2 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 2 family ethernet-switching vlan members 2
set interfaces ae0 unit 3 esi 00:11:22:22:22:22:22:22:22:22
set interfaces ae0 unit 3 esi all-active
set interfaces ae0 unit 3 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 3 family ethernet-switching vlan members 3
set interfaces xe-2/0/6 enable
set interfaces xe-2/0/6 flexible-vlan-tagging
set interfaces xe-2/0/6 encapsulation flexible-ethernet-services
set interfaces xe-2/0/6 unit 1 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 1 family ethernet-switching vlan members 1
set interfaces xe-2/0/6 unit 2 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 2 family ethernet-switching vlan members 2
set interfaces xe-2/0/6 unit 3 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 3 family ethernet-switching vlan members 3
set interfaces xe-2/1/0 unit 0 family inet address 203.0.113.1/24
set interfaces xe-2/1/1 gigether-options 802.3ad ae1
set interfaces xe-2/1/2 gigether-options 802.3ad ae1
set interfaces ae1 flexible-vlan-tagging
set interfaces ae1 encapsulation flexible-ethernet-services
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 unit 1 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 1 family ethernet-switching vlan members 1
```

```
set interfaces ae1 unit 2 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 2 family ethernet-switching vlan members 2
set interfaces ae1 unit 3 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 3 family ethernet-switching vlan members 3
set multi-chassis multi-chassis-protection 203.0.113.2 interface ae1
set protocols iccp local-ip-addr 203.0.113.1
set protocols iccp peer 203.0.113.2 session-establishment-hold-time 600
set protocols iccp peer 203.0.113.2 redundancy-group-id-list 2
set protocols iccp peer 203.0.113.2 liveness-detection minimum-interval 10000
set protocols iccp peer 203.0.113.2 liveness-detection multiplier 3
```

PE1: EVPN-MPLS Configuration

```
set interfaces lo0 unit 0 family inet address 198.51.100.1/32 primary
set interfaces xe-2/0/0 unit 0 family inet address 192.0.2.2/24
set interfaces xe-2/0/0 unit 0 family mpls
set interfaces xe-2/0/2 unit 0 family inet address 192.0.2.111/24
set interfaces xe-2/0/2 unit 0 family mpls
set interfaces irb unit 1 family inet address 10.2.1.1/24 virtual-gateway-address 10.2.1.254
set interfaces irb unit 2 family inet address 10.2.2.1/24 virtual-gateway-address 10.2.2.254
set interfaces irb unit 3 family inet address 10.2.3.1/24 virtual-gateway-address 10.2.3.254
set routing-options router-id 198.51.100.1
set routing-options autonomous-system 65000
set routing-options forwarding-table export evpn-pplb
set protocols mpls interface xe-2/0/0.0
set protocols mpls interface xe-2/0/2.0
set protocols bgp group evpn type internal
set protocols bgp group evpn local-address 198.51.100.1
set protocols bgp group evpn family evpn signaling
set protocols bgp group evpn local-as 65000
set protocols bgp group evpn neighbor 198.51.100.2
set protocols bgp group evpn neighbor 198.51.100.3
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface xe-2/0/0.0
set protocols ospf area 0.0.0.0 interface xe-2/0/2.0
set protocols ldp interface xe-2/0/0.0
set protocols ldp interface xe-2/0/2.0
set protocols ldp interface lo0.0
set policy-options policy-statement evpn-pplb from protocol evpn
set policy-options policy-statement evpn-pplb then load-balance per-packet
set routing-instances evpn1 instance-type virtual-switch
set routing-instances evpn1 interface xe-2/0/6.1
set routing-instances evpn1 interface ae0.1
set routing-instances evpn1 interface ae1.1
set routing-instances evpn1 route-distinguisher 1:10
set routing-instances evpn1 vrf-target target:1:5
set routing-instances evpn1 protocols evpn extended-vlan-list 1
set routing-instances evpn1 protocols evpn mlag bgp-peer 198.51.100.2
set routing-instances evpn1 switch-options service-id 1
set routing-instances evpn1 vlans v1 vlan-id 1
set routing-instances evpn1 vlans v1 l3-interface irb.1
set routing-instances evpn1 vlans v1 no-arp-suppression
set routing-instances evpn2 instance-type virtual-switch
set routing-instances evpn2 interface xe-2/0/6.2
set routing-instances evpn2 interface ae0.2
set routing-instances evpn2 interface ae1.2
```

```

set routing-instances evpn2 route-distinguisher 1:20
set routing-instances evpn2 vrf-target target:1:6
set routing-instances evpn2 protocols evpn extended-vlan-list 2
set routing-instances evpn2 protocols evpn mlag bgp-peer 198.51.100.2
set routing-instances evpn2 switch-options service-id 2
set routing-instances evpn2 vlans v1 vlan-id 2
set routing-instances evpn2 vlans v1 l3-interface irb.2
set routing-instances evpn2 vlans v1 no-arp-suppression
set routing-instances evpn3 instance-type virtual-switch
set routing-instances evpn3 interface xe-2/0/6.3
set routing-instances evpn3 interface ae0.3
set routing-instances evpn3 interface ae1.3
set routing-instances evpn3 route-distinguisher 1:30
set routing-instances evpn3 vrf-target target:1:7
set routing-instances evpn3 protocols evpn extended-vlan-list 3
set routing-instances evpn3 protocols evpn mlag bgp-peer 198.51.100.2
set routing-instances evpn3 switch-options service-id 3
set routing-instances evpn3 vlans v1 vlan-id 3
set routing-instances evpn3 vlans v1 l3-interface irb.3
set routing-instances evpn3 vlans v1 no-arp-suppression

```

PE2: MC-LAG Configuration

```

set chassis aggregated-devices ethernet device-count 3
set interfaces xe-2/0/1 gigether-options 802.3ad ae0
set interfaces xe-2/0/6 enable
set interfaces xe-2/0/6 flexible-vlan-tagging
set interfaces xe-2/0/6 encapsulation flexible-ethernet-services
set interfaces xe-2/0/6 unit 1 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 1 family ethernet-switching vlan members 1
set interfaces xe-2/0/6 unit 2 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 2 family ethernet-switching vlan members 2
set interfaces xe-2/0/6 unit 3 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 3 family ethernet-switching vlan members 3
set interfaces xe-2/1/0 unit 0 family inet address 203.0.113.2/24
set interfaces xe-2/1/1 gigether-options 802.3ad ae1
set interfaces xe-2/1/2 gigether-options 802.3ad ae1
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:11:11:11:11
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 aggregated-ether-options mc-ae mc-ae-id 1
set interfaces ae0 aggregated-ether-options mc-ae redundancy-group 2
set interfaces ae0 aggregated-ether-options mc-ae chassis-id 1
set interfaces ae0 aggregated-ether-options mc-ae mode active-active
set interfaces ae0 aggregated-ether-options mc-ae status-control standby
set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
set interfaces ae0 unit 1 esi all-active
set interfaces ae0 unit 1 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 1 family ethernet-switching vlan members 1
set interfaces ae0 unit 2 esi 00:11:11:11:11:11:11:11:11:11
set interfaces ae0 unit 2 esi all-active
set interfaces ae0 unit 2 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 2 family ethernet-switching vlan members 2

```

```
set interfaces ae0 unit 3 esi 00:11:22:22:22:22:22:22
set interfaces ae0 unit 3 esi all-active
set interfaces ae0 unit 3 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 3 family ethernet-switching vlan members 3
set interfaces ae1 flexible-vlan-tagging
set interfaces ae1 encapsulation flexible-ethernet-services
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 unit 1 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 1 family ethernet-switching vlan members 1
set interfaces ae1 unit 2 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 2 family ethernet-switching vlan members 2
set interfaces ae1 unit 3 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 3 family ethernet-switching vlan members 3
set multi-chassis multi-chassis-protection 203.0.113.1 interface ae1
set protocols iccp local-ip-addr 203.0.113.2
set protocols iccp peer 203.0.113.1 session-establishment-hold-time 600
set protocols iccp peer 203.0.113.1 redundancy-group-id-list 2
set protocols iccp peer 203.0.113.1 liveness-detection minimum-interval 10000
set protocols iccp peer 203.0.113.1 liveness-detection multiplier 3
```

PE2: EVPN-MPLS Configuration

```
set interfaces xe-2/0/0 unit 0 family inet address 192.0.2.222/24
set interfaces xe-2/0/0 unit 0 family mpls
set interfaces xe-2/0/2 unit 0 family inet address 192.0.2.22/24
set interfaces xe-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 198.51.100.2/32 primary
set interfaces irb unit 1 family inet address 10.2.1.2/24 virtual-gateway-address 10.2.1.254
set interfaces irb unit 2 family inet address 10.2.2.2/24 virtual-gateway-address 10.2.2.254
set interfaces irb unit 3 family inet address 10.2.3.2/24 virtual-gateway-address 10.2.3.254
set routing-options router-id 198.51.100.2
set routing-options autonomous-system 65000
set routing-options forwarding-table export evpn-pplb
set protocols mpls interface xe-2/0/2.0
set protocols mpls interface xe-2/0/0.0
set protocols bgp group evpn type internal
set protocols bgp group evpn local-address 198.51.100.2
set protocols bgp group evpn family evpn signaling
set protocols bgp group evpn local-as 65000
set protocols bgp group evpn neighbor 198.51.100.1
set protocols bgp group evpn neighbor 198.51.100.3
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface xe-2/0/0.0
set protocols ospf area 0.0.0.0 interface xe-2/0/2.0
set protocols ldp interface xe-2/0/0.0
set protocols ldp interface xe-2/0/2.0
set protocols ldp interface lo0.0
set policy-options policy-statement evpn-pplb from protocol evpn
set policy-options policy-statement evpn-pplb then load-balance per-packet
set routing-instances evpn1 instance-type virtual-switch
set routing-instances evpn1 interface xe-2/0/6.1
set routing-instances evpn1 interface ae0.1
set routing-instances evpn1 interface ae1.1
set routing-instances evpn1 route-distinguisher 1:11
set routing-instances evpn1 vrf-target target:1:5
set routing-instances evpn1 protocols evpn extended-vlan-list 1
```

```

set routing-instances evpn1 protocols evpn mlag bgp-peer 198.51.100.1
set routing-instances evpn1 switch-options service-id 1
set routing-instances evpn1 vlans v1 vlan-id 1
set routing-instances evpn1 vlans v1 l3-interface irb.1
set routing-instances evpn1 vlans v1 no-arp-suppression
set routing-instances evpn2 instance-type virtual-switch
set routing-instances evpn2 interface xe-2/0/6.2
set routing-instances evpn2 interface ae0.2
set routing-instances evpn2 interface ae1.2
set routing-instances evpn2 route-distinguisher 1:21
set routing-instances evpn2 vrf-target target:1:6
set routing-instances evpn2 protocols evpn extended-vlan-list 2
set routing-instances evpn2 protocols evpn mlag bgp-peer 198.51.100.1
set routing-instances evpn2 switch-options service-id 2
set routing-instances evpn2 vlans v1 vlan-id 2
set routing-instances evpn2 vlans v1 l3-interface irb.2
set routing-instances evpn2 vlans v1 no-arp-suppression
set routing-instances evpn3 instance-type virtual-switch
set routing-instances evpn3 interface xe-2/0/6.3
set routing-instances evpn3 interface ae0.3
set routing-instances evpn3 interface ae1.3
set routing-instances evpn3 route-distinguisher 1:31
set routing-instances evpn3 vrf-target target:1:7
set routing-instances evpn3 protocols evpn extended-vlan-list 3
set routing-instances evpn3 protocols evpn mlag bgp-peer 198.51.100.1
set routing-instances evpn3 switch-options service-id 3
set routing-instances evpn3 vlans v1 vlan-id 3
set routing-instances evpn3 vlans v1 l3-interface irb.3
set routing-instances evpn3 vlans v1 no-arp-suppression

```

PE1: Configuring MC-LAG

Step-by-Step Procedure

1. Set the number of aggregated Ethernet interfaces on PE1.

```

[edit]
user@switch# set chassis aggregated-devices ethernet device-count 3

```
2. Configure aggregated Ethernet interface ae0 on interface xe-2/0/1, and configure LACP and MC-LAG on ae0. Divide aggregated Ethernet interface ae0 into three logical interfaces (ae0.1, ae0.2, and ae0.3). For each logical interface, specify an ESI, place the logical interface in MC-LAG active-active mode, and map the logical interface to a VLAN.

```

[edit]
user@switch# set interfaces xe-2/0/1 gigether-options 802.3ad ae0
user@switch# set interfaces ae0 flexible-vlan-tagging
user@switch# set interfaces ae0 encapsulation flexible-ethernet-services
user@switch# set interfaces ae0 aggregated-ether-options lacp active
user@switch# set interfaces ae0 aggregated-ether-options lacp periodic fast
user@switch# set interfaces ae0 aggregated-ether-options lacp system-id 00:00:11:11:11:11
user@switch# set interfaces ae0 aggregated-ether-options lacp admin-key 1
user@switch# set interfaces ae0 aggregated-ether-options mc-ae mc-ae-id 1
user@switch# set interfaces ae0 aggregated-ether-options mc-ae redundancy-group 2

```

```

user@switch# set interfaces ae0 aggregated-ether-options mc-ae chassis-id 0
user@switch# set interfaces ae0 aggregated-ether-options mc-ae mode
    active-active
user@switch# set interfaces ae0 aggregated-ether-options mc-ae status-control
    active
user@switch# set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
user@switch# set interfaces ae0 unit 1 esi all-active
user@switch# set interfaces ae0 unit 1 family ethernet-switching interface-mode
    trunk
user@switch# set interfaces ae0 unit 1 family ethernet-switching vlan members 1
user@switch# set interfaces ae0 unit 2 esi 00:11:11:11:11:11:11:11:11:11
user@switch# set interfaces ae0 unit 2 esi all-active
user@switch# set interfaces ae0 unit 2 family ethernet-switching interface-mode
    trunk
user@switch# set interfaces ae0 unit 2 family ethernet-switching vlan members 2
user@switch# set interfaces ae0 unit 3 esi 00:11:22:22:22:22:22:22:22:22
user@switch# set interfaces ae0 unit 3 esi all-active
user@switch# set interfaces ae0 unit 3 family ethernet-switching interface-mode
    trunk
user@switch# set interfaces ae0 unit 3 family ethernet-switching vlan members 3

```

3. Configure physical interface xe-2/0/6, and divide it into three logical interfaces (xe-2/0/6.1, xe-2/0/6.2, and xe-2/0/6.3). Map each logical interface to a VLAN.

```

[edit]
user@switch# set interfaces xe-2/0/6 enable
user@switch# set interfaces xe-2/0/6 flexible-vlan-tagging
user@switch# set interfaces xe-2/0/6 encapsulation flexible-ethernet-services
user@switch# set interfaces xe-2/0/6 unit 1 family ethernet-switching
    interface-mode trunk
user@switch# set interfaces xe-2/0/6 unit 1 family ethernet-switching vlan members
    1
user@switch# set interfaces xe-2/0/6 unit 2 family ethernet-switching
    interface-mode trunk
user@switch# set interfaces xe-2/0/6 unit 2 family ethernet-switching vlan members
    2
user@switch# set interfaces xe-2/0/6 unit 3 family ethernet-switching
    interface-mode trunk
user@switch# set interfaces xe-2/0/6 unit 3 family ethernet-switching vlan members
    3

```

4. Configure physical interface xe-2/1/0 as a Layer 3 interface, on which you configure ICCP. Specify the interface with the IP address of 203.0.113.2 on PE2 as the ICCP peer to PE1.

```

[edit]
user@switch# set interfaces xe-2/1/0 unit 0 family inet address 203.0.113.1/24
user@switch# set protocols iccp local-ip-addr 203.0.113.1
user@switch# set protocols iccp peer 203.0.113.2 session-establishment-hold-time
    600
user@switch# set protocols iccp peer 203.0.113.2 redundancy-group-id-list 2
user@switch# set protocols iccp peer 203.0.113.2 liveness-detection
    minimum-interval 10000
user@switch# set protocols iccp peer 203.0.113.2 liveness-detection multiplier 3

```


5. Configure aggregated Ethernet interface ae1 on interfaces xe-2/1/1 and xe-2/1/2, and configure LACP on ae1. Divide aggregated Ethernet interface ae1 into three logical interfaces (ae1.1, ae1.2, and ae1.3), and map each logical interface to a VLAN. Specify ae1 as the multichassis protection link between PE1 and PE2.

```
[edit]
user@switch# set interfaces xe-2/1/1 gigether-options 802.3ad ae1
user@switch# set interfaces xe-2/1/2 gigether-options 802.3ad ae1
user@switch# set interfaces ae1 flexible-vlan-tagging
user@switch# set interfaces ae1 encapsulation flexible-ethernet-services
user@switch# set interfaces ae1 aggregated-ether-options lacp active
user@switch# set interfaces ae1 unit 1 family ethernet-switching interface-mode
trunk
user@switch# set interfaces ae1 unit 1 family ethernet-switching vlan members 1
user@switch# set interfaces ae1 unit 2 family ethernet-switching interface-mode
trunk
user@switch# set interfaces ae1 unit 2 family ethernet-switching vlan members 2
user@switch# set interfaces ae1 unit 3 family ethernet-switching interface-mode
trunk
user@switch# set interfaces ae1 unit 3 family ethernet-switching vlan members 3
user@switch# set multi-chassis multi-chassis-protection 203.0.113.2 interface ae1
```

PE1: Configuring EVPN-MPLS

Step-by-Step Procedure

1. Configure the loopback interface, and the interfaces connected to the other PE devices.

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 198.51.100.1/32 primary
user@switch# set interfaces xe-2/0/0 unit 0 family inet address 192.0.2.2/24
user@switch# set interfaces xe-2/0/0 unit 0 family mpls
user@switch# set interfaces xe-2/0/2 unit 0 family inet address 192.0.2.111/24
user@switch# set interfaces xe-2/0/2 unit 0 family mpls
```

2. Configure IRB interfaces irb.1, irb.2, and irb.3.

```
[edit]
user@switch# set interfaces irb unit 1 family inet address 10.2.1.1/24
virtual-gateway-address 10.2.1.254
user@switch# set interfaces irb unit 2 family inet address 10.2.2.1/24
virtual-gateway-address 10.2.2.254
user@switch# set interfaces irb unit 3 family inet address 10.2.3.1/24
virtual-gateway-address 10.2.3.254
```

3. Assign a router ID and the autonomous system in which PE1, PE2, and PE3 reside.

```
[edit]
user@switch# set routing-options router-id 198.51.100.1
user@switch# set routing-options autonomous-system 65000
```

4. Enable per-packet load-balancing for EVPN routes when EVPN multihoming active-active mode is used.

```
[edit]
user@switch# set routing-options forwarding-table export evpn-pplb
user@switch# set policy-options policy-statement evpn-pplb from protocol evpn
user@switch# set policy-options policy-statement evpn-pplb then load-balance
per-packet
```

5. Enable MPLS on interfaces xe-2/0/0.0 and xe-2/0/2.0.

```
[edit]
user@switch# set protocols mpls interface xe-2/0/0.0
user@switch# set protocols mpls interface xe-2/0/2.0
```

6. Configure an IBGP overlay that includes PE1, PE2, and PE3.

```
[edit]
user@switch# set protocols bgp group evpn type internal
user@switch# set protocols bgp group evpn local-address 198.51.100.1
user@switch# set protocols bgp group evpn family evpn signaling
user@switch# set protocols bgp group evpn local-as 65000
user@switch# set protocols bgp group evpn neighbor 198.51.100.2
user@switch# set protocols bgp group evpn neighbor 198.51.100.3
```

7. Configure OSPF as the internal routing protocol for EVPN by specifying an area ID and interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ospf area 0.0.0.0 interface lo0.0
user@switch# set protocols ospf area 0.0.0.0 interface xe-2/0/0.0
user@switch# set protocols ospf area 0.0.0.0 interface xe-2/0/2.0
```

8. Configure the Label Distribution Protocol (LDP) on the loopback interface and the interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ldp interface lo0.0
user@switch# set protocols ldp interface xe-2/0/0.0
user@switch# set protocols ldp interface xe-2/0/2.0
```

9. Configure virtual switch routing instances for VLAN v1, which is assigned VLAN IDs of 1, 2, and 3, and include the interfaces and other entities associated with the VLAN.

```
[edit]
user@switch# set routing-instances evpn1 instance-type virtual-switch
user@switch# set routing-instances evpn1 interface xe-2/0/6.1
user@switch# set routing-instances evpn1 interface ae0.1
user@switch# set routing-instances evpn1 interface ae1.1
user@switch# set routing-instances evpn1 route-distinguisher 1:10
user@switch# set routing-instances evpn1 vrf-target target:1:5
user@switch# set routing-instances evpn1 protocols evpn extended-vlan-list 1
user@switch# set routing-instances evpn1 protocols evpn mlag bgp-peer
198.51.100.2
```

```

user@switch# set routing-instances evpn1 switch-options service-id 1
user@switch# set routing-instances evpn1 vlans v1 vlan-id 1
user@switch# set routing-instances evpn1 vlans v1 l3-interface irb.1
user@switch# set routing-instances evpn1 vlans v1 no-arp-suppression
user@switch# set routing-instances evpn2 instance-type virtual-switch
user@switch# set routing-instances evpn2 interface xe-2/0/6.2
user@switch# set routing-instances evpn2 interface ae0.2
user@switch# set routing-instances evpn2 interface ae1.2
user@switch# set routing-instances evpn2 route-distinguisher 1:20
user@switch# set routing-instances evpn2 vrf-target target:1:6
user@switch# set routing-instances evpn2 protocols evpn extended-vlan-list 2
user@switch# set routing-instances evpn2 protocols evpn mclag bgp-peer
198.51.100.2
user@switch# set routing-instances evpn2 switch-options service-id 2
user@switch# set routing-instances evpn2 vlans v1 vlan-id 2
user@switch# set routing-instances evpn2 vlans v1 l3-interface irb.2
user@switch# set routing-instances evpn2 vlans v1 no-arp-suppression
user@switch# set routing-instances evpn3 instance-type virtual-switch
user@switch# set routing-instances evpn3 interface xe-2/0/6.3
user@switch# set routing-instances evpn3 interface ae0.3
user@switch# set routing-instances evpn3 interface ae1.3
user@switch# set routing-instances evpn3 route-distinguisher 1:30
user@switch# set routing-instances evpn3 vrf-target target:1:7
user@switch# set routing-instances evpn3 protocols evpn extended-vlan-list 3
user@switch# set routing-instances evpn3 protocols evpn mclag bgp-peer
198.51.100.2
user@switch# set routing-instances evpn3 switch-options service-id 3
user@switch# set routing-instances evpn3 vlans v1 vlan-id 3
user@switch# set routing-instances evpn3 vlans v1 l3-interface irb.3
user@switch# set routing-instances evpn3 vlans v1 no-arp-suppression

```

PE2: Configuring MC-LAG

Step-by-Step Procedure

1. Set the number of aggregated Ethernet interfaces on PE2.
2. Configure aggregated Ethernet interface ae0 on interface xe-2/0/1, and configure LACP and MC-LAG on ae0. Divide aggregated Ethernet interface ae0 into three logical interfaces (ae0.1, ae0.2, and ae0.3). For each logical interface, specify an ESI, place the logical interface in MC-LAG active-active mode, and map the logical interface to a VLAN.

```

[edit]
user@switch# set interfaces xe-2/0/1 gigether-options 802.3ad ae0
user@switch# set interfaces ae0 flexible-vlan-tagging
user@switch# set interfaces ae0 encapsulation flexible-ethernet-services
user@switch# set interfaces ae0 aggregated-ether-options lacp active
user@switch# set interfaces ae0 aggregated-ether-options lacp periodic fast
user@switch# set interfaces ae0 aggregated-ether-options lacp system-id
00:00:11:11:11:11
user@switch# set interfaces ae0 aggregated-ether-options lacp admin-key 1
user@switch# set interfaces ae0 aggregated-ether-options mc-ae mc-ae-id 1

```

```
user@switch# set interfaces ae0 aggregated-ether-options mc-ae redundancy-group
2
user@switch# set interfaces ae0 aggregated-ether-options mc-ae chassis-id 1
user@switch# set interfaces ae0 aggregated-ether-options mc-ae mode
active-active
user@switch# set interfaces ae0 aggregated-ether-options mc-ae status-control
standby
user@switch# set interfaces ae0 unit 1 esi 00:11:22:33:44:55:66:77:88:99
user@switch# set interfaces ae0 unit 1 esi all-active
user@switch# set interfaces ae0 unit 1 family ethernet-switching interface-mode
trunk
user@switch# set interfaces ae0 unit 1 family ethernet-switching vlan members 1
user@switch# set interfaces ae0 unit 2 esi 00:11:11:11:11:11:11:11:11:11
user@switch# set interfaces ae0 unit 2 esi all-active
user@switch# set interfaces ae0 unit 2 family ethernet-switching interface-mode
trunk
user@switch# set interfaces ae0 unit 2 family ethernet-switching vlan members 2
user@switch# set interfaces ae0 unit 3 esi 00:11:22:22:22:22:22:22:22:22
user@switch# set interfaces ae0 unit 3 esi all-active
user@switch# set interfaces ae0 unit 3 family ethernet-switching interface-mode
trunk
user@switch# set interfaces ae0 unit 3 family ethernet-switching vlan members 3
```

3. Configure physical interface xe-2/0/6, and divide it into three logical interfaces (xe-2/0/6.1, xe-2/0/6.2, and xe-2/0/6.3). Map each logical interface to a VLAN.

```
[edit]
set interfaces xe-2/0/6 enable
set interfaces xe-2/0/6 flexible-vlan-tagging
set interfaces xe-2/0/6 encapsulation flexible-ethernet-services
set interfaces xe-2/0/6 unit 1 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 1 family ethernet-switching vlan members 1
set interfaces xe-2/0/6 unit 2 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 2 family ethernet-switching vlan members 2
set interfaces xe-2/0/6 unit 3 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 3 family ethernet-switching vlan members 3
```

4. Configure physical interface xe-2/1/0 as a Layer 3 interface, on which you configure ICCP. Specify the interface with the IP address of 203.0.113.1 on PE1 as the ICCP peer to PE2.

```
[edit]
set interfaces xe-2/1/0 unit 0 family inet address 203.0.113.2/24
set protocols iccp local-ip-addr 203.0.113.2
set protocols iccp peer 203.0.113.1 session-establishment-hold-time 600
set protocols iccp peer 203.0.113.1 redundancy-group-id-list 2
set protocols iccp peer 203.0.113.1 liveness-detection minimum-interval 10000
set protocols iccp peer 203.0.113.1 liveness-detection multiplier 3
```

5. Configure aggregated Ethernet interface ae1 on interfaces xe-2/1/1 and xe-2/1/2, and configure LACP on ae1. Divide aggregated Ethernet interface ae1 into three logical interfaces (ae1.1, ae1.2, and ae1.3), and map each logical interface to a VLAN. Specify ae1 as the multichassis protection link between PE1 and PE2.

```
[edit]
set interfaces xe-2/1/1 gigether-options 802.3ad ae1
set interfaces xe-2/1/2 gigether-options 802.3ad ae1
set interfaces ae1 flexible-vlan-tagging
set interfaces ae1 encapsulation flexible-ethernet-services
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 unit 1 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 1 family ethernet-switching vlan members 1
set interfaces ae1 unit 2 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 2 family ethernet-switching vlan members 2
set interfaces ae1 unit 3 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 3 family ethernet-switching vlan members 3
set multi-chassis multi-chassis-protection 203.0.113.1 interface ae1
```

PE2: Configuring EVPN-MPLS

Step-by-Step Procedure

1. Configure the loopback interface, and the interfaces connected to the other PE devices.

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 198.51.100.2/32 primary
user@switch# set interfaces xe-2/0/0 unit 0 family inet address 192.0.2.222/24
user@switch# set interfaces xe-2/0/0 unit 0 family mpls
user@switch# set interfaces xe-2/0/2 unit 0 family inet address 192.0.2.22/24
user@switch# set interfaces xe-2/0/2 unit 0 family mpls
```

2. Configure IRB interfaces irb.1, irb.2, and irb.3.

```
[edit]
user@switch# set interfaces irb unit 1 family inet address 10.2.1.2/24
virtual-gateway-address 10.2.1.254
user@switch# set interfaces irb unit 2 family inet address 10.2.2.2/24
virtual-gateway-address 10.2.2.254
user@switch# set interfaces irb unit 3 family inet address 10.2.3.2/24
virtual-gateway-address 10.2.3.254
```

3. Assign a router ID and the autonomous system in which PE1, PE2, and PE3 reside.

```
[edit]
user@switch# set routing-options router-id 198.51.100.2
user@switch# set routing-options autonomous-system 65000
```

4. Enable per-packet load-balancing for EVPN routes when EVPN multihoming active-active mode is used.

```
[edit]
user@switch# set routing-options forwarding-table export evpn-pplb
user@switch# set policy-options policy-statement evpn-pplb from protocol evpn
```

```
user@switch# set policy-options policy-statement evpn-pplb then load-balance
per-packet
```

5. Enable MPLS on interfaces xe-2/0/0.0 and xe-2/0/2.0.

```
[edit]
user@switch# set protocols mpls interface xe-2/0/0.0
user@switch# set protocols mpls interface xe-2/0/2.0
```

6. Configure an IBGP overlay that includes PE1, PE2, and PE3.

```
[edit]
user@switch# set protocols bgp group evpn type internal
user@switch# set protocols bgp group evpn local-address 198.51.100.2
user@switch# set protocols bgp group evpn family evpn signaling
user@switch# set protocols bgp group evpn local-as 65000
user@switch# set protocols bgp group evpn neighbor 198.51.100.1
user@switch# set protocols bgp group evpn neighbor 198.51.100.3
```

7. Configure OSPF as the internal routing protocol for EVPN by specifying an area ID and interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ospf area 0.0.0.0 interface lo0.0
user@switch# set protocols ospf area 0.0.0.0 interface xe-2/0/0.0
user@switch# set protocols ospf area 0.0.0.0 interface xe-2/0/2.0
```

8. Configure the Label Distribution Protocol (LDP) on the loopback interface and the interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ldp interface lo0.0
user@switch# set protocols ldp interface xe-2/0/0.0
user@switch# set protocols ldp interface xe-2/0/2.0
```

9. Configure virtual switch routing instances for VLAN v1, which is assigned VLAN IDs of 1, 2, and 3, and include the interfaces and other entities associated with the VLAN.

```
[edit]
user@switch# set routing-instances evpn1 instance-type virtual-switch
user@switch# set routing-instances evpn1 interface xe-2/0/6.1
user@switch# set routing-instances evpn1 interface ae0.1
user@switch# set routing-instances evpn1 interface ae1.1
user@switch# set routing-instances evpn1 route-distinguisher 1:11
user@switch# set routing-instances evpn1 vrf-target target:1:5
user@switch# set routing-instances evpn1 protocols evpn extended-vlan-list 1
user@switch# set routing-instances evpn1 protocols evpn mlag bgp-peer
198.51.100.1
user@switch# set routing-instances evpn1 switch-options service-id 1
user@switch# set routing-instances evpn1 vlans v1 vlan-id 1
user@switch# set routing-instances evpn1 vlans v1 l3-interface irb.1
user@switch# set routing-instances evpn1 vlans v1 no-arp-suppression
user@switch# set routing-instances evpn2 instance-type virtual-switch
user@switch# set routing-instances evpn2 interface xe-2/0/6.2
```

```

user@switch# set routing-instances evpn2 interface ae0.2
user@switch# set routing-instances evpn2 interface ae1.2
user@switch# set routing-instances evpn2 route-distinguisher 1:21
user@switch# set routing-instances evpn2 vrf-target target:1:6
user@switch# set routing-instances evpn2 protocols evpn extended-vlan-list 2
user@switch# set routing-instances evpn2 protocols evpn mclag bgp-peer
198.51.100.1
user@switch# set routing-instances evpn2 switch-options service-id 2
user@switch# set routing-instances evpn2 vlans v1 vlan-id 2
user@switch# set routing-instances evpn2 vlans v1 l3-interface irb.2
user@switch# set routing-instances evpn2 vlans v1 no-arp-suppression
user@switch# set routing-instances evpn3 instance-type virtual-switch
user@switch# set routing-instances evpn3 interface xe-2/0/6.3
user@switch# set routing-instances evpn3 interface ae0.3
user@switch# set routing-instances evpn3 interface ae1.3
user@switch# set routing-instances evpn3 route-distinguisher 1:31
user@switch# set routing-instances evpn3 vrf-target target:1:7
user@switch# set routing-instances evpn3 protocols evpn extended-vlan-list 3
user@switch# set routing-instances evpn3 protocols evpn mclag bgp-peer
198.51.100.1
user@switch# set routing-instances evpn3 switch-options service-id 3
user@switch# set routing-instances evpn3 vlans v1 vlan-id 3
user@switch# set routing-instances evpn3 vlans v1 l3-interface irb.3
user@switch# set routing-instances evpn3 vlans v1 no-arp-suppression

```

PE3 Configuration

CLI Quick Configuration

PE3: EVPN-MPLS Configuration

```

set interfaces lo0 unit 0 family inet address 198.51.100.3/32 primary
set interfaces xe-2/0/2 unit 0 family inet address 192.0.2.1/24
set interfaces xe-2/0/2 unit 0 family mpls
set interfaces xe-2/0/3 unit 0 family inet address 192.0.2.11/24
set interfaces xe-2/0/3 unit 0 family mpls
set interfaces xe-2/0/6 enable
set interfaces xe-2/0/6 flexible-vlan-tagging
set interfaces xe-2/0/6 encapsulation flexible-ethernet-services
set interfaces xe-2/0/6 unit 1 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 1 family ethernet-switching vlan members 1
set interfaces xe-2/0/6 unit 2 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 2 family ethernet-switching vlan members 2
set interfaces xe-2/0/6 unit 3 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/6 unit 3 family ethernet-switching vlan members 3
set interfaces irb unit 1 family inet address 10.2.1.3/24 virtual-gateway-address 10.2.1.254
set interfaces irb unit 2 family inet address 10.2.2.3/24 virtual-gateway-address 10.2.2.254
set interfaces irb unit 3 family inet address 10.2.3.3/24 virtual-gateway-address 10.2.3.254
set routing-options router-id 198.51.100.3
set routing-options autonomous-system 65000
set routing-options forwarding-table export evpn-pplb
set protocols mpls interface xe-2/0/2.0
set protocols mpls interface xe-2/0/3.0
set protocols bgp group evpn type internal
set protocols bgp group evpn local-address 198.51.100.3
set protocols bgp group evpn family evpn signaling

```

```

set protocols bgp group evpn local-as 65000
set protocols bgp group evpn neighbor 198.51.100.1
set protocols bgp group evpn neighbor 198.51.100.2
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface xe-2/0/2.0
set protocols ospf area 0.0.0.0 interface xe-2/0/3.0
set protocols ldp interface lo0.0
set protocols ldp interface xe-2/0/2.0
set protocols ldp interface xe-2/0/3.0
set policy-options policy-statement evpn-pplb from protocol evpn
set policy-options policy-statement evpn-pplb then load-balance per-packet
set routing-instances evpn1 instance-type virtual-switch
set routing-instances evpn1 interface xe-2/0/6.1
set routing-instances evpn1 route-distinguisher 1:12
set routing-instances evpn1 vrf-target target:1:5
set routing-instances evpn1 protocols evpn extended-vlan-list 1
set routing-instances evpn1 switch-options service-id 1
set routing-instances evpn1 vlans v1 vlan-id 1
set routing-instances evpn1 vlans v1 l3-interface irb.1
set routing-instances evpn1 vlans v1 no-arp-suppression
set routing-instances evpn2 instance-type virtual-switch
set routing-instances evpn2 interface xe-2/0/6.2
set routing-instances evpn2 route-distinguisher 1:22
set routing-instances evpn2 vrf-target target:1:6
set routing-instances evpn2 protocols evpn extended-vlan-list 2
set routing-instances evpn2 switch-options service-id 2
set routing-instances evpn2 vlans v1 vlan-id 2
set routing-instances evpn2 vlans v1 l3-interface irb.2
set routing-instances evpn2 vlans v1 no-arp-suppression
set routing-instances evpn3 instance-type virtual-switch
set routing-instances evpn3 interface xe-2/0/6.3
set routing-instances evpn3 route-distinguisher 1:32
set routing-instances evpn3 vrf-target target:1:7
set routing-instances evpn3 protocols evpn extended-vlan-list 3
set routing-instances evpn3 switch-options service-id 3
set routing-instances evpn3 vlans v1 vlan-id 3
set routing-instances evpn3 vlans v1 l3-interface irb.3
set routing-instances evpn3 vlans v1 no-arp-suppression

```

PE3: Configuring EVPN-MPLS

Step-by-Step Procedure

1. Configure the loopback interface, and the interfaces connected to the other PE devices.


```

[edit]
user@switch# set interfaces lo0 unit 0 family inet address 198.51.100.3/32 primary
user@switch# set interfaces xe-2/0/2 unit 0 family inet address 192.0.2.1/24
user@switch# set interfaces xe-2/0/2 unit 0 family mpls
user@switch# set interfaces xe-2/0/3 unit 0 family inet address 192.0.2.11/24
user@switch# set interfaces xe-2/0/3 unit 0 family mpls

```
2. Configure interface xe-2/0/6, which is connected to the host.


```

[edit]
user@switch# set interfaces xe-2/0/6 enable

```



```

user@switch# set interfaces xe-2/0/6 flexible-vlan-tagging
user@switch# set interfaces xe-2/0/6 encapsulation flexible-ethernet-services
user@switch# set interfaces xe-2/0/6 unit 1 family ethernet-switching
  interface-mode trunk
user@switch# set interfaces xe-2/0/6 unit 1 family ethernet-switching vlan members
  1
user@switch# set interfaces xe-2/0/6 unit 2 family ethernet-switching
  interface-mode trunk
user@switch# set interfaces xe-2/0/6 unit 2 family ethernet-switching vlan members
  2
user@switch# set interfaces xe-2/0/6 unit 3 family ethernet-switching
  interface-mode trunk
user@switch# set interfaces xe-2/0/6 unit 3 family ethernet-switching vlan members
  3

```

3. Configure IRB interfaces irb.1, irb.2, and irb.3.

```

[edit]
user@switch# set interfaces irb unit 1 family inet address 10.2.1.3/24
  virtual-gateway-address 10.2.1.254
user@switch# set interfaces irb unit 2 family inet address 10.2.2.3/24
  virtual-gateway-address 10.2.2.254
user@switch# set interfaces irb unit 3 family inet address 10.2.3.3/24
  virtual-gateway-address 10.2.3.254

```

4. Assign a router ID and the autonomous system in which PE1, PE2, and PE3 reside.

```

[edit]
user@switch# set routing-options router-id 198.51.100.3
user@switch# set routing-options autonomous-system 65000

```

5. Enable per-packet load-balancing for EVPN routes when EVPN multihoming active-active mode is used.

```

[edit]
user@switch# set routing-options forwarding-table export evpn-pplb
user@switch# set policy-options policy-statement evpn-pplb from protocol evpn
user@switch# set policy-options policy-statement evpn-pplb then load-balance
  per-packet

```

6. Enable MPLS on interfaces xe-2/0/2.0 and xe-2/0/3.0.

```

[edit]
user@switch# set protocols mpls interface xe-2/0/2.0
user@switch# set protocols mpls interface xe-2/0/3.0

```

7. Configure an IBGP overlay that includes PE1, PE2, and PE3.

```

[edit]
user@switch# set protocols bgp group evpn type internal
user@switch# set protocols bgp group evpn local-address 198.51.100.3
user@switch# set protocols bgp group evpn family evpn signaling
user@switch# set protocols bgp group evpn local-as 65000
user@switch# set protocols bgp group evpn neighbor 198.51.100.1

```

```
user@switch# set protocols bgp group evpn neighbor 198.51.100.2
```

8. Configure OSPF as the internal routing protocol for EVPN by specifying an area ID and interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ospf area 0.0.0.0 interface lo0.0
user@switch# set protocols ospf area 0.0.0.0 interface xe-2/0/2.0
user@switch# set protocols ospf area 0.0.0.0 interface xe-2/0/3.0
```

9. Configure the LDP on the loopback interface and the interfaces on which EVPN-MPLS is enabled.

```
[edit]
user@switch# set protocols ldp interface lo0.0
user@switch# set protocols ldp interface xe-2/0/2.0
user@switch# set protocols ldp interface xe-2/0/3.0
```

10. Configure virtual switch routing instances for VLAN v1, which is assigned VLAN IDs of 1, 2, and 3, and include the interfaces and other entities associated with the VLAN.

```
[edit]
user@switch# set routing-instances evpn1 instance-type virtual-switch
user@switch# set routing-instances evpn1 interface xe-2/0/6.1
user@switch# set routing-instances evpn1 route-distinguisher 1:12
user@switch# set routing-instances evpn1 vrf-target target:1:5
user@switch# set routing-instances evpn1 protocols evpn extended-vlan-list 1
user@switch# set routing-instances evpn1 switch-options service-id 1
user@switch# set routing-instances evpn1 vlans v1 vlan-id 1
user@switch# set routing-instances evpn1 vlans v1 l3-interface irb.1
user@switch# set routing-instances evpn1 vlans v1 no-arp-suppression
user@switch# set routing-instances evpn2 instance-type virtual-switch
user@switch# set routing-instances evpn2 interface xe-2/0/6.2
user@switch# set routing-instances evpn2 route-distinguisher 1:22
user@switch# set routing-instances evpn2 vrf-target target:1:6
user@switch# set routing-instances evpn2 protocols evpn extended-vlan-list 2
user@switch# set routing-instances evpn2 switch-options service-id 2
user@switch# set routing-instances evpn2 vlans v1 vlan-id 2
user@switch# set routing-instances evpn2 vlans v1 l3-interface irb.2
user@switch# set routing-instances evpn2 vlans v1 no-arp-suppression
user@switch# set routing-instances evpn3 instance-type virtual-switch
user@switch# set routing-instances evpn3 interface xe-2/0/6.3
user@switch# set routing-instances evpn3 route-distinguisher 1:32
user@switch# set routing-instances evpn3 vrf-target target:1:7
user@switch# set routing-instances evpn3 protocols evpn extended-vlan-list 3
user@switch# set routing-instances evpn3 switch-options service-id 3
user@switch# set routing-instances evpn3 vlans v1 vlan-id 3
user@switch# set routing-instances evpn3 vlans v1 l3-interface irb.3
user@switch# set routing-instances evpn3 vlans v1 no-arp-suppression
```

Related Documentation • [Understanding EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG on page 211](#)

PART 3

Configuration Statements and Operational Commands

- Configuration Statements on page 253
- Operational Commands on page 305

CHAPTER 9

Configuration Statements

- [bgp-peer](#) on page 254
- [decapsulate-inner-vlan](#) on page 255
- [duplicate-mac-detection](#) on page 256
- [encapsulate-inner-vlan](#) on page 257
- [encapsulation \(Physical Interface\)](#) on page 258
- [encapsulation vxlan](#) on page 264
- [esi](#) on page 265
- [evpn](#) on page 267
- [extended-vlan-list](#) on page 269
- [extended-vni-list](#) on page 270
- [global-mac-ip-limit](#) on page 271
- [global-mac-ip-table-aging-time](#) on page 272
- [ingress-node-replication \(EVPN\)](#) on page 273
- [instance-type](#) on page 274
- [interface \(EVPN Routing Instances\)](#) on page 277
- [interface \(Routing Instances\)](#) on page 278
- [interface-mac-ip-limit](#) on page 279
- [interface-mac-limit \(VPLS\)](#) on page 280
- [ip-prefix-routes](#) on page 282
- [l2-learning](#) on page 285
- [mac-statistics](#) on page 286
- [mac-ip-table-size](#) on page 287
- [mac-table-size](#) on page 288
- [mclag](#) on page 289
- [multicast-mode \(EVPN\)](#) on page 290
- [no-arp-suppression](#) on page 291
- [no-mac-learning](#) on page 292
- [packet-action](#) on page 294

- [routing-instances on page 297](#)
- [switch-options on page 298](#)
- [traceoptions \(Protocols EVPN\) on page 299](#)
- [vlan-id \(routing instance\) on page 301](#)
- [vlans on page 302](#)
- [vni-options on page 303](#)
- [vni on page 304](#)

bgp-peer

Syntax	<code>bgp-peer <i>ip-address</i>;</code>
Hierarchy Level	[edit routing-instances <i>name</i> protocols evpn mclag]
Release Information	Statement introduced in Junos OS Release 17.4R1 on MX Series routers, EX Series switches, and Junos Fusion Enterprise.
Description	Configure an aggregation device in a Junos Fusion Enterprise or a multichassis link aggregation group (MC-LAG) topology to interwork with an Ethernet VPN-MPLS (EVPN-MPLS) device.
Options	<i>ip-address</i> —IP address of the BGP peer. Typically, a BGP peer is identified by the IP address of the device's loopback interface.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Understanding EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG on page 211

decapsulate-inner-vlan

Syntax	decapsulate-inner-vlan
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> vxlan], [edit vlans <i>vlan-name</i> vxlan]
Release Information	Statement introduced in Junos OS Release 14.1X53-D10. Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.
Description	Configure the switch to de-encapsulate a preserved original VLAN tag (in the inner Ethernet packet) from a VXLAN encapsulated packet.
Default	A preserved VLAN tag is dropped when the packet is de-encapsulated.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• <i>Understanding VXLANs</i>• <i>Manually Configuring VXLANs on a QFX5100 Switch</i>• <i>Examples: Manually Configuring VXLANs on QFX Switches</i>• encapsulate-inner-vlan on page 257

duplicate-mac-detection

Syntax	<pre>duplicate-mac-detection { auto-recovery-time <i>minutes</i>; detection-threshold <i>detection-threshold</i>; detection-window <i>seconds</i>; }</pre>
Hierarchy Level	<pre>[edit logical-systems <i>logical-systems-name</i> protocols evpn], [edit logical-systems <i>logical-systems-name</i> routing-instances <i>routing-instance-name</i> protocols evpn], [edit protocols evpn] [edit routing-instances <i>routing-instance-name</i> protocols evpn],</pre>
Release Information	Statement introduced in Junos OS Release 17.4R1 on MX series routers, EX Series switches, and QFX Series switches.
Description	Duplicate MAC address detection settings. You can configure a threshold for MAC address mobility events and the window for detecting the number of MAC address mobility event. In addition, you can also configure the optional recovery time that the router waits before the duplicate MAC address is unsurpressed.
Options	<p>auto-recovery-time—The length of time a device suppresses a duplicate MAC address. At the end of this duration, MAC address updates will resume. When an auto-recovery-time is not specified, duplicate MAC address updates will continue to be suppressed. To manually clear the suppression of duplicate MAC address, use the clear evpn duplicate-mac-suppression command.</p> <p>Range: 5-360 minutes</p> <p>detection-threshold—Number of MAC mobility events detected for a given MAC address before it is identified as a duplicate MAC address. Once the threshold is reached, updates for this MAC address is suppressed.</p> <p>Default: 5</p> <p>Range: 2-20</p> <p>detection-window—The time interval used in detecting a duplicate MAC address. When the number of MAC mobility events for a MAC address exceeds the detection-threshold within the detection window, the MAC address is identified as a duplicate MAC address.</p> <p>Default: 180 seconds</p> <p>Range: 5-600 seconds</p>



NOTE: To ensure that the mobility advertisements have sufficient time to age out, set an auto-recovery-time greater than the detection window.

Required Privilege Level routing

Related Documentation

- [Overview of MAC Mobility on page 207](#)
- [Changing Duplicate MAC Address Detection Settings on page 209](#)

encapsulate-inner-vlan

Syntax encapsulate-inner-vlan

Hierarchy Level [edit routing-instances *routing-instance-name* vlan vlans *vlan-name* vxlan],
[edit vlans *vlan-name* vxlan]

Release Information Statement introduced in Junos OS Release 14.1R2 for MX Series Routers.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series.
Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.

Description Configure the switch to preserve the original VLAN tag (in the inner Ethernet packet) when performing Virtual Extensible LAN (VXLAN) encapsulation.



Default The original tag is dropped when the packet is encapsulated.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- [Understanding VXLANs](#)
- [Manually Configuring VXLANs on a QFX5100 Switch](#)
- [Examples: Manually Configuring VXLANs on QFX Switches](#)
- [decapsulate-accept-inner-vlan](#)

encapsulation (Physical Interface)

Syntax	encapsulation (atm-ccc-cell-relay atm-pvc cisco-hdlc cisco-hdlc-ccc cisco-hdlc-tcc ethernet-bridge ethernet-ccc ethernet-over-atm ethernet-tcc ethernet-vpls ethernet-vpls-fr ether-vpls-over-atm-llc ethernet-vpls-ppp extended-frame-relay-ccc extended-frame-relay-ether-type-tcc extended-frame-relay-tcc extended-vlan-bridge extended-vlan-ccc extended-vlan-tcc extended-vlan-vpls flexible-ethernet-services flexible-frame-relay frame-relay frame-relay-ccc frame-relay-ether-type frame-relay-ether-type-tcc frame-relay-port-ccc frame-relay-tcc generic-services multilink-frame-relay-uni-nni ppp ppp-ccc ppp-tcc vlan-ccc vlan-vci-ccc vlan-vpls);
Hierarchy Level	[edit interfaces <i>interface-name</i>], [edit interfaces rlsq <i>number:number</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 11.1 for EX Series switches. Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers (flexible-ethernet-services , ethernet-ccc , and ethernet-tcc options only).
Description	Specify the physical link-layer encapsulation type.
	<div>  <p>NOTE: Not all encapsulation types are supported on the switches. See the switch CLI.</p> </div>
Default	ppp—Use serial PPP encapsulation.
Options	<div>  <p>NOTE: Frame Relay, ATM, PPP, SONET, and SATSOP options are not supported on the EX Series switches.</p> </div> <p>atm-ccc-cell-relay—Use ATM cell-relay encapsulation.</p> <p>atm-pvc—Defined in RFC 2684, <i>Multiprotocol Encapsulation over ATM Adaptation Layer 5</i>. When you configure physical ATM interfaces with ATM PVC encapsulation, an RFC 2684-compliant ATM Adaptation Layer 5 (AAL5) tunnel is set up to route the ATM cells over a Multiprotocol Label Switching (MPLS) path that is typically established between two MPLS-capable routers using the Label Distribution Protocol (LDP).</p> <p>cisco-hdlc—Use Cisco-compatible High-Level Data Link Control (HDLC) framing. E1, E3, SONET/SDH, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:</p>

- CCC version (**cisco-hdlc-ccc**)—The logical interface does not require an encapsulation statement. When you use this encapsulation type, you can configure the **ccc** family only.
- TCC version (**cisco-hdlc-tcc**)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

cisco-hdlc-ccc—Use Cisco-compatible HDLC framing on CCC circuits.

cisco-hdlc-tcc—Use Cisco-compatible HDLC framing on TCC circuits for connecting different media.

ethernet-bridge—Use Ethernet bridge encapsulation on Ethernet interfaces that have bridging enabled and that must accept all packets.

ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, CCC is not supported.

ethernet-over-atm—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, this encapsulation type allows ATM interfaces to connect to devices that support only bridge protocol data units (BPDUs). Junos OS does not completely support bridging, but accepts BPDU packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or ARP in the payload, and drops the rest. For packets destined to the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and MAC header, and the packet is forwarded to the ATM interface.

ethernet-tcc—For interfaces that carry IPv4 traffic, use Ethernet TCC encapsulation on interfaces that must accept packets carrying standard TPID values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, TCC is not supported.

ethernet-vpls—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard TPID values. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

ethernet-vpls-fr—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use the MAC address to forward the packet into a given VPLS instance.

ethernet-vpls-ppp—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 PPP connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.

ether-vpls-over-atm-llc—For ATM intelligent queuing (IQ) interfaces only, use the Ethernet virtual private LAN service (VPLS) over ATM LLC encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3 encapsulated Ethernet frames with the frame check sequence (FCS) field removed.

extended-frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to CCC. When you use this encapsulation type, you can configure the **ccc** family only.

extended-frame-relay-ether-type-tcc—Use extended Frame Relay ether type TCC for Cisco-compatible Frame Relay for DLCIs 1 through 1022. This encapsulation type is used for circuits with different media on either side of the connection.

extended-frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits to connect different media. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC.

extended-vlan-bridge—Use extended VLAN bridge encapsulation on Ethernet interfaces that have IEEE 802.1Q VLAN tagging and bridging enabled and that must accept packets carrying TPID 0x8100 or a user-defined TPID.

extended-vlan-ccc—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values. Extended VLAN CCC encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. When you use this encapsulation type, you can configure the **ccc** family only. For 8-port, 12-port, and 48-port Fast Ethernet PICs, extended VLAN CCC is not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC is not supported.

extended-vlan-tcc—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. For 4-port Gigabit Ethernet PICs, extended VLAN TCC is not supported.

extended-vlan-vpls—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.



NOTE: The built-in Gigabit Ethernet PIC on an M7i router does not support extended VLAN VPLS encapsulation.

flexible-ethernet-services—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), and for Gigabit Ethernet interfaces, use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. Aggregated Ethernet bundles can use this encapsulation type. This encapsulation type allows you to configure any combination of route, TCC, CCC, Layer 2 virtual private networks (VPNs), and VPLS encapsulations on a single physical port. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

flexible-frame-relay—For IQ interfaces only, use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.

frame-relay—Use Frame Relay encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E1, E3, link services, SONET/SDH, T1, T3, and voice services interfaces can use Frame Relay encapsulation.

frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. This encapsulation is same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC. The logical interface must also have **frame-relay-ccc** encapsulation. When you use this encapsulation type, you can configure the **ccc** family only.

frame-relay-ether-type—Use Frame Relay ether type encapsulation for compatibility with the Cisco Frame Relay. IETF frame relay encapsulation identifies the payload format using NLPID and SNAP formats. Cisco-compatible Frame Relay encapsulation uses the Ethernet type to identify the type of payload.



NOTE: When the encapsulation type is set to Cisco-compatible Frame Relay encapsulation, ensure that the LMI type is set to ANSI or Q933-A.

frame-relay-ether-type-tcc—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media. This encapsulation is Cisco-compatible Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to TCC.

frame-relay-port-ccc—Use Frame Relay port CCC encapsulation to transparently carry all the DLCIs between two customer edge (CE) routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. The connection between the two CE routers can be either user-to-network interface (UNI) or network-to-network interface (NNI); this is completely transparent to the PE routers. When you use this encapsulation type, you can configure the **ccc** family only.

frame-relay-tcc—This encapsulation is similar to Frame Relay CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

generic-services—Use generic services encapsulation for services with a hierarchical scheduler.

multilink-frame-relay-uni-nni—Use MLFR UNI NNI encapsulation. This encapsulation is used on link services, voice services interfaces functioning as FRF.16 bundles, and their constituent T1 or E1 interfaces, and is supported on LSQ and redundant LSQ interfaces.

ppp—Use serial PPP encapsulation. This encapsulation is defined in RFC 1661, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. E1, E3, SONET/SDH, T1, and T3 interfaces can use PPP encapsulation.

ppp-ccc—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only.

ppp-tcc—Use serial PPP encapsulation on TCC circuits for connecting different media. When you use this encapsulation type, you can configure the **tcc** family only.

vlan-ccc—Use Ethernet VLAN encapsulation on CCC circuits. VLAN CCC encapsulation supports TPID 0x8100 only. When you use this encapsulation type, you can configure the **ccc** family only.

vlan-vci-ccc—Use ATM-to-Ethernet interworking encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only. All logical interfaces configured on the Ethernet interface must also have the encapsulation type set to **vlan-vci-ccc**.

vlan-vpls—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.



NOTE:

- Label-switched interfaces (LSIs) do not support VLAN VPLS encapsulation. Therefore, you can only use VLAN VPLS encapsulation on a PE-router-to-CE-router interface and not a core-facing interface.
 - Starting with Junos OS release 13.3, a commit error occurs when you configure **vlan-vpls** encapsulation on a physical interface and configure family **inet** on one of the logical units. Previously, it was possible to commit this invalid configuration.
-

Required Privilege interface—To view this statement in the configuration.
Level interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Interface Encapsulation on Physical Interfaces*
- *Configuring CCC Encapsulation for Layer 2 VPNs*
- *Configuring Layer 2 Switching Cross-Connects Using CCC*
- *Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits*
- *Configuring ATM Interface Encapsulation*
- *Configuring ATM-to-Ethernet Interworking*
- *Configuring VLAN and Extended VLAN Encapsulation*
- *Configuring VLAN and Extended VLAN Encapsulation*
- *Configuring Encapsulation for Layer 2 Wholesale VLAN Interfaces*
- *Configuring Interfaces for Layer 2 Circuits*
- *Configuring Interface Encapsulation on PTX Series Packet Transport Routers*
- *Configuring MPLS LSP Tunnel Cross-Connects Using CCC*
- *Configuring TCC*
- *Configuring VPLS Interface Encapsulation*
- *Configuring Interfaces for VPLS Routing*
- *Defining the Encapsulation for Switching Cross-Connects*
- *Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)*

encapsulation vxlan

Syntax	encapsulation vxlan;
Hierarchy Level	{edit protocols evpn }, [edit routing-instances <i>routing-instance-name</i> protocols evpn]
Release Information	Statement introduced in Junos OS Release 16.1. Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.
Description	Configure a VXLAN encapsulation type. This statement is required for a VXLAN EVPN instance.



NOTE: If you configure the `encapsulation vxlan` statement, then you must also configure the [extended-vni-list](#) statement.



NOTE: The `encapsulation vxlan` statement is an exclusive command. You cannot configure the `encapsulation vxlan` statement with the [extended-vlan-list](#) statement, or other commands associated with MPLS EVPN instances.

Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
---------------------------------	---

Related Documentation	<ul style="list-style-type: none">• Understanding EVPN with VXLAN Data Plane Encapsulation on page 5• EVPN Over VXLAN Encapsulation Configuration Overview for QFX5100 Series• Example: Configuring an EVPN Control Plane and VXLAN Data Plane on page 29
------------------------------	---

esi

Syntax	<pre>esi identifier { all-active single-active; }</pre>
Hierarchy Level	<pre>[edit interfaces interface-name] [edit interfaces interface-name unit logical-unit-number]</pre>
Release Information	<p>Statement introduced in Junos OS Release 14.1.</p> <p>Statement introduced in Junos OS Releases 15.1F6 and 17.1 on MX Series routers with MPCs.</p>
Description	<p>Configure an Ethernet Segment Identifier (ESI) on a physical, aggregated Ethernet, or logical interface in either EVPN multihoming active-standby or active-active mode.</p> <p>In releases before Junos OS Release 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI only on a physical or aggregated Ethernet interface, for example, set interfaces ae0 esi 00:11:22:33:44:55:66:77:88:99. If you specify an ESI on a physical or aggregated Ethernet interface, keep in mind that an ESI is a factor in the designated forwarder (DF) election process. For example, assume that you configure EVPN multihoming active-standby on aggregated Ethernet interface ae0, and given the ESI configured on ae0 and other determining factors, the DF election results in ae0 being in the down state. Further, all logical interfaces configured on aggregated Ethernet interface ae0, for example, set interfaces ae0 unit 1 and set interfaces ae0 unit 2 are also in the down state, which renders logical interfaces ae0.1 and ae0.2 unable to provide services to their respective customer sites (VLANs).</p> <p>Starting with Junos OS Releases 15.1F6 and 17.1R1 for MX Series routers and Junos OS Release 17.3R1 for EX9200 switches, you can specify an ESI on a logical interface. If you specify an ESI on a logical interface, the DF election process now occurs at the individual logical interface level, which enables you to better utilize logical interfaces. For example, assume that you configure logical interfaces ae0.1 and ae0.2 on aggregated Ethernet interface ae0. You configure EVPN multihoming active-standby on both logical interfaces and given the ESI configured on ae0.1 and other determining factors, the DF election results in ae0.1 being in the down state. Despite logical interface ae0.1 being down, logical interface ae0.2 and other logical interfaces configured on ae0 can be in the up state and provide services to their respective customer sites (VLANs).</p>
Options	<p>esi—Ten octet value. ESI value 0 and all fixed filters are reserved, and not used for configuring a multihomed Ethernet segment.</p>



NOTE:

- Two interfaces (physical, logical, or aggregated Ethernet) cannot be configured with the same ESI value.

- The left most octet must be configured as 00. The other 9 octets are fully configurable.

.....

all-active—Configure the EVPN active-active multihoming mode of operation.

single-active—Configure the EVPN active-standby multihoming mode of operation.

Required Privilege	interface—To view this statement in the configuration.
Level	interface-control—To add this statement to the configuration.

Related Documentation	<ul style="list-style-type: none">• evpn on page 267• Example: Configuring an ESI on a Logical Interface With EVPN Multihoming on page 195
------------------------------	---

evpn

Syntax	<pre> evpn { designated-forwarder-election-hold-timesseconds; designated-forwarder-preference-least; extended-vni-list { vni-options } no-default-gateway-ext-comm encapsulation <i>encapsulation-type</i>; extended-vni-list; multicast-mode client ingress-replication; vni-options { vni xxx vrf-target export target:xxx:xx vni xxx vrf-export <i>name</i> } extended-vni-all; no-default-gateway-ext-comm; } extended-vlan-list <i>vlan-id</i> [<i>vlan-id set</i>]; extended-isid-list (<i>single-isid</i> <i>isid-list</i> <i>isid-range</i> all); pbb-evpn-core; service-type <i>service-type</i>; interface <i>interface-name</i> { ignore-encapsulation-mismatch; interface-mac-limit <i>limit</i> { packet-action drop; } mac-pinning (EVPN Routing Instances) no-mac-learning; static-mac <i>mac-address</i>; } interface-mac-limit <i>limit</i> { packet-action drop; } label-allocation per-instance; mac-statistics; mac-table-size <i>limit</i> { packet-action drop; } no-mac-learning; traceoptions { file <i>filename</i> <files <i>number</i>> <size <i>size</i>> <world-readable no-world-readable>; flag <i>flag</i> <flag-modifier>; } } </pre>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols]</p> <p>[edit protocols]</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols]</p>
Release Information	Statement introduced in Junos OS Release 13.2 for EVPNs on MX 3D Series routers.

designated-forwarder-election-hold-time *seconds* statement introduced in Junos OS Release 14.1.

extended-vlan-list *vlan-id* | [*vlan-id set*] statement introduced in Junos OS Release 14.1. Statement introduced in Junos OS Release 14.1-X53-D30 for QFX Series switches.




NOTE: The **extended-vlan-list** statement is not supported on QFX10000 switches.

designated-forwarder-preference-least and **service-type** statements introduced in Junos OS Release 17.3 for MX Series routers with MPC and MIC interfaces.



Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.

Description	Enable an Ethernet VPN (EVPN) on the routing instance.
Options	<p>designated-forwarder-election-hold-time <i>seconds</i>—Time in seconds to wait before electing a designated forwarder (DF). Range: 1 through 1800 seconds</p> <p>designated-forwarder-preference-least—Use least preference value for DF election. By default, the preference-based DF election is based on the highest preference value configured.</p> <p>evpn-etree—Configure an Ethernet VPN E-TREE service.</p> <p>service-type <i>service-type</i>—Service interface type:</p> <ul style="list-style-type: none"> • vlan-based—VLAN based service interface. • vlan-bundle—VLAN bundle service interface. • vlan-aware bundle—VLAN-aware bundle service interface <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Implementing EVPN-VXLAN for Data Centers</i> • <i>Configuring EVPN Routing Instances</i> • Tracing EVPN Traffic and Operations on page 27 • <i>Implementing EVPN-VXLAN for Data Centers</i> • <i>Provider Backbone Bridging (PBB) and EVPN Integration Overview</i>

extended-vlan-list

Syntax	<code>extended-vlan-list <i>vlan-id</i> [<i>vlan-id set</i>];</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols] [edit routing-instances <i>routing-instance-name</i> instance-type virtual-switch protocols evpn]
Release Information	Statement introduced in Junos OS Release 14.1. Statement introduced in Junos OS Release 14.2 on EX Series switches. Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.
Description	Specify the VLAN or range of VLANs that are extended over the WAN, wherein all the single VLAN bridge domains corresponding to these VLANs are stretched.
<div>  <p>NOTE: The <code>extended-vni-list</code> statement is an exclusive command. You cannot configure the <code>extended-vni-list</code> statement with either the <code>extended-vlan-list</code> or <code>extended-vni-all</code> statements.</p> </div>	
Options	<p><i>vlan-id</i>—VLAN ID to be EVPN extended.</p> <p><i>vlan-id set</i>—List of VLAN IDs to be EVPN extended. Range: 1 through 4094 VLANs</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • evpn on page 267

extended-vni-list

Syntax	<code>extended-vni-list [<i>list of VNIs</i> all];</code>
Hierarchy Level	<p>For MX Series routers, EX9200, and QFX Series switches:</p> <pre>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols], [edit routing-instances <i>routing-instance-name</i> protocols evpn]</pre>
Release Information	<p>Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.</p> <p>Statement introduced in Junos OS Release 16.1 for MX Series.</p> <p>Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.</p> <p>Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.</p>
Description	<p>Establishes which VXLAN Network Identifiers (VNI) will be part of the Virtual Switch (VS) instance. When you issue the commit check command to verify the candidate configuration syntax without committing it, it also checks if the specified VNI(s) is associated with a bridge domain (BD) (bridge-domain <i>name</i> vxlan <i>vni</i>).</p> <p>There are different broadcast, unknown unicast, and multicast (BUM) replication options available in Ethernet Virtual Private Network (EVPN). By using the extended-vni-list statement, you forgo a multicast underlay in favor of EVPN and VXLAN ingress-replication.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> NOTE: The extended-vni-list statement is an exclusive command. You cannot configure the extended-vni-list statement with either the extended-vlan-list or extended-vni-all statements.</p> </div> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> NOTE: If you configure the extended-vni-list statement, then you must also configure the encapsulation vxlan statement.</p> </div>
Options	<p>list of VNIs—Specify a single VNI or list of VNIs as part of the VS instance, for example extended-vni-list [10-50 60 70].</p> <p>all—Include all VNIs as part of the VS instance. By specifying all, you bypass the commit check process and all configured BDs in the EVPN are considered extended.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring an EVPN Control Plane and VXLAN Data Plane on page 29 • Understanding EVPN with VXLAN Data Plane Encapsulation on page 5

- *EVPN Over VXLAN Encapsulation Configuration Overview for QFX5100 Series*
- *show configuration protocols evpn*
- *Implementing EVPN-VXLAN for Data Centers*

global-mac-ip-limit

Syntax	global-mac-ip-limit { <code>number</code> ; }
Hierarchy Level	[edit logical-systems <i>name</i> protocols l2-learning], [edit protocols l2-learning]
Release Information	Statement introduced in Junos OS Release 17.4R1 for MX Series routers and EX9200 switches.
Description	Limit the number of entries that can be added systemwide to the MAC-IP bindings database.
Options	<code>number</code> —Specify the maximum number of entries that can added systemwide to the MAC-IP bindings database.. When the specified maximum is reached, no new entries are added to the database. Range: 20 through 1048575.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression on page 14 • global-mac-ip-table-aging-time on page 272 • interface-mac-ip-limit on page 279 • mac-ip-table-size on page 287

global-mac-ip-table-aging-time

Syntax	global-mac-ip-table-aging-time <i>seconds</i> ;
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols l2-learning], [edit protocols l2-learning]
Release Information	Statement introduced in Junos OS Release 17.4R1 for MX Series routers and EX9200 switches.
Description	Configure the timeout interval systemwide for entries in the MAC-IP address bindings database.
Options	<p>seconds—Specify the time that is elapsed before entries in the MAC-IP bindings database are timed out and deleted.</p> <p>Range: For MX Series routers, 10 through 1 million; for EX9200 switches, 60 through 1 million.</p> <p>Default: 1200 (20 minutes)</p>
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression on page 14• global-mac-ip-limit on page 271• interface-mac-ip-limit on page 279• mac-ip-table-size on page 287

ingress-node-replication (EVPN)

Syntax	<code>ingress-node-replication;</code>
Hierarchy Level	<code>[edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> vxlan],</code> <code>[edit vlans <i>vlan-name</i> vxlan]</code>
Release Information	Statement introduced in Junos OS Release 14.1X53-D30 for QFX Series switches. Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.
Description	Ingress replication is supported only for EVPN-VXLAN and enabled by default; no configuration is needed. EVPN A/A with VXLAN encapsulation is based on the local bias for traffic coming from the access layer (redundant Layer 2 gateway function through a pair of top-of-rack switches). Because the traffic has no MPLS label, the split-horizon filtering rule for multi-home Ethernet segment is modified to be based on the IP address of the EVPN provider edge (PE) instead of the MPLS ES-label. This is called local bias for EVPN-VXLAN. Each EVPN PE tracks the IP address of its peer multihomed EVPN PE that share the same Ethernet segment. This is the source VTEP IP address (outer SIP) for each VXLAN packet received from other EVPN PE. The local bias filtering rule is enforced on both ingress and egress PEs for the multidestination traffic. For egress traffic, there is no forwarding of any multidestination packets to the same multihomed Ethernet segment that an egress PE shares with its ingress PE regardless of the egress PE's DF election status for that Ethernet segment. Ingress traffic is responsible for forwarding multi-destination packets coming from any directly attached access interfaces to the rest of the multi-home Ethernet segments associated with it regardless of the ingress PE's designated forwarder election status on the connected physical device segment.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Understanding EVPN with VXLAN Data Plane Encapsulation on page 5

instance-type

Syntax	<code>instance-type type;</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i>], [edit routing-instances <i>routing-instance-name</i>]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>virtual-switch and layer2-control options introduced in Junos OS Release 8.4.</p> <p>Statement introduced in Junos OS Release 9.2 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 11.3 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 12.3 for ACX Series routers.</p> <p>mpls-internet-multicast option introduced in Junos OS Release 11.1 for the EX Series, M Series, MX Series, and T Series.</p> <p>evpn option introduced in Junos OS Release 13.2 for MX 3D Series routers.</p> <p>evpn option introduced in Junos OS Release 17.3 for the QFX Series.</p> <p>forwarding option introduced in Junos OS Release 14.2 for the PTX Series.</p> <p>mpls-forwarding option introduced in Junos OS Release 16.1 for the MX Series.</p> <p>evpn-vpws option introduced in Junos OS Release 17.1 for MX Series routers.</p> <p>Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.</p>
Description	Define the type of routing instance.

Options



NOTE: On ACX Series routers, you can configure only the forwarding, virtual router, and VRF routing instances.

type—Can be one of the following:

- **evpn**—(MX 3D Series routers, QFX switches, and EX9200 switches)—Enable an Ethernet VPN (EVPN) on the routing instance.
hierarchy level.
- **evpn-vpws**—Enable an Ethernet VPN (EVPN) Virtual Private Wire Service (VPWS) on the routing instance.
- **forwarding**—Provide support for filter-based forwarding, where interfaces are not associated with instances. All interfaces belong to the default instance. Other instances are used for populating RPD learned routes. For this instance type, there is no one-to-one mapping between an interface and a routing instance. All interfaces belong to the default instance inet.0.
- **l2backhaul-vpn**—Provide support for Layer 2 wholesale VLAN packets with no existing corresponding logical interface. When using this instance, the router learns both the outer tag and inner tag of the incoming packets, when the **instance-role** statement is defined as **access**, or the outer VLAN tag only, when the **instance-role** statement is defined as **nni**.

- **l2vpn**—Enable a Layer 2 VPN on the routing instance. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- **layer2-control**—(MX Series routers only) Provide support for RSTP or MSTP in customer edge interfaces of a VPLS routing instance. This instance type cannot be used if the customer edge interface is multihomed to two provider edge interfaces. If the customer edge interface is multihomed to two provider edge interfaces, use the default BPDU tunneling.
- **mpls-forwarding**—(MX Series routers only) Allow filtering and translation of route distinguisher (RD) values in IPv4 and IPv6 VPN address families on both routes received and routes sent for selected BGP sessions. In particular, for Inter-AS VPN Option-B networks, this option can prevent the malicious injection of VPN labels from one peer AS boundary router to another.
- **mpls-internet-multicast**—(EX Series, M Series, MX Series, and T Series routers only) Provide support for ingress replication provider tunnels to carry IP multicast data between routers through an MPLS cloud, using MBGP or next-generation MVPN.
- **no-forwarding**—This is the default routing instance. Do not create a corresponding forwarding instance. Use this routing instance type when a separation of routing table information is required. There is no corresponding forwarding table. All routes are installed into the default forwarding table. IS-IS instances are strictly nonforwarding instance types.
- **virtual-router**—Enable a virtual router routing instance. This instance type is similar to a VPN routing and forwarding instance type, but used for non-VPN-related applications. You must configure the **interface** statement for this type of routing instance. You do not need to configure the **route-distinguisher**, **vrf-import**, and **vrf-export** statements.
- **virtual-switch**—(MX Series routers, EX9200 switches, and QFX switches only) Provide support for Layer 2 bridging. Use this routing instance type to isolate a LAN segment with its Spanning Tree Protocol (STP) instance and to separate its VLAN identifier space.
- **vpls**—Enable VPLS on the routing instance. Use this routing instance type for point-to-multipoint LAN implementations between a set of sites in a VPN. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- **vrf**—VPN routing and forwarding (VRF) instance. Provides support for Layer 3 VPNs, where interface routes for each instance go into the corresponding forwarding table only. Required to create a Layer 3 VPN. Create a VRF table (**instance-name.inet.0**) that contains the routes originating from and destined for a particular Layer 3 VPN. For this instance type, there is a one-to-one mapping between an interface and a routing instance. Each VRF instance corresponds with a forwarding table. Routes on an interface go into the corresponding forwarding table. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.

Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Configuring Routing Instances on PE Routers in VPNs</i> • <i>Configuring EVPN Routing Instances</i> • Configuring EVPN Routing Instances on EX9200 Switches on page 25 • <i>Configuring Virtual Router Routing Instances</i> • <i>Example: Configuring Filter-Based Forwarding on the Source Address</i> • <i>Example: Configuring Filter-Based Forwarding on Logical Systems</i> • <i>vpws-service-id</i>

interface (EVPN Routing Instances)

Syntax	<pre> interface <i>interface-name</i> { ignore-encapsulation-mismatch; interface-mac-limit <i>limit</i> { packet-action drop; } mac-pinning (EVPN Routing Instances) no-mac-learning; protect-interface static-mac <i>mac-address</i>; vpws-service-id }</pre>
Hierarchy Level	<pre> [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols] [edit routing-instances <i>routing-instance-name</i> protocols evpn]</pre>
Release Information	<p>Statement introduced in Junos OS Release 13.2 for EVPNs on MX 3D Series routers.</p> <p>Statement introduced in Junos OS Release 14.2 on EX Series switches.</p> <p>Statement (mac-pinning) introduced in Junos OS Release 16.2 on MX Series routers.</p> <p>vpws-service-id statement introduced in Junos OS Release 17.1 on MX Series routers.</p> <p>Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.</p>
Description	Specify each interface over which the Ethernet VPN (EVPN) traffic travels between the PE device and customer edge (CE) device. The interfaces are bound to the EVPN routing instance.
Options	<p><i>interface-name</i>—Name of the interface.</p> <p>The remaining statements are explained separately. See CLI Explorer.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring EVPN Routing Instances • Configuring EVPN Routing Instances on EX9200 Switches on page 25 • evpn on page 267 • instance-type on page 274 • vpws-service-id


interface (Routing Instances)

Syntax	<pre>interface <i>interface-name</i> { description <i>text</i>; }</pre>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i>], [edit routing-instances <i>routing-instance-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.2 for EX Series switches. Statement introduced in Junos OS Release 12.3 for ACX Series routers. Statement introduced in Junos OS Release 13.2 for MX 3D Series routers.
Description	Specify the interface over which the VPN traffic travels between the PE device and CE device. You configure the interface on the PE device. If the value vrf is specified for the instance-type statement included in the routing instance configuration, this statement is required.
Options	<p><i>interface-name</i>—Name of the interface.</p> <p>The remaining statement is explained separately. See CLI Explorer.</p>
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring Routing Instances on PE Routers in VPNs• Configuring EVPN Routing Instances• Configuring EVPN Routing Instances on EX9200 Switches on page 25• interface (VPLS Routing Instances)

interface-mac-ip-limit

Syntax	<code>interface-mac-ip-limit <i>number</i>;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> switch-options],</p> <p>[edit logical-systems <i>logical-system-name</i> vlans <i>vlan-name</i> switch-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols evpn],</p> <p>[edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> switch-options],</p> <p>[edit vlans <i>vlan-name</i> switch-options]</p>
Release Information	Statement introduced in Junos OS Release 17.4R1.
Description	<p>Limit the number of entries that can be learned on an interface through the MAC-IP bindings database. On MX routers and EX9200 switches, this limit can be applied to EVPN routing instances. On MX routers, you can also apply this limit to bridge-domains configured in a virtual switch routing instance. On EX9200 switches, this limit can also be applied to VLANs configured in a routing instance.</p> <p>To apply this limit systemwide, use the global-mac-ip-limit statement at the [edit protocols l2-learning] hierarchy level.</p>
Options	<p><i>number</i>—Specify the maximum number of MAC(IP) bindings per interface. After that maximum is reached, no additional MAC(IP) entries are added to the database.</p> <p>Range: 1 through 1024.</p> <p>Default: 124</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression on page 14 • mac-ip-table-size on page 287

interface-mac-limit (VPLS)

Syntax	<pre>interface-mac-limit <i>limit</i> { packet-action drop; }</pre>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls site <i>site-name</i> interfaces <i>interface-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols evpn],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols evpn interface <i>interface-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols vpls site <i>site-name</i> interfaces <i>interface-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Support for EVPNs introduced in Junos OS Release 13.2 on MX 3D Series routers.</p> <p>Support for EVPNs introduced in Junos OS Release 14.2 on EX Series switches.</p>
Description	<p>Specify the maximum number of media access control (MAC) addresses that can be learned by the EVPN or VPLS routing instance. You can configure the same limit for all interfaces configured for a routing instance. You can also configure a limit for a specific interface.</p> <p>Starting with Junos OS Release 12.3R4, if you do not configure the parameter to limit the number of MAC addresses to be learned by a VPLS instance, the default value is not effective. Instead, if you do not include the interface-mac-limit option at the [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls site <i>site-name</i> interfaces <i>interface-name</i>], hierarchy level, this setting is not present in the configuration with the default value of 1024 addresses. If you upgrade a router running a Junos OS release earlier than Release 12.3R4 to Release 12.3R4 or later, you must configure the interface-mac-limit option with a valid value for it to be saved in the configuration.</p>
Options	<p>limit—Number of MAC addresses that can be learned from each interface.</p> <p>Range: 1 through 131,071 MAC addresses</p>
<div>  NOTE: For M120 devices only, the range is 16 through 65,536 MAC addresses. </div>	
<p>Default: 1024 addresses</p> <p>The remaining statement is explained separately. See CLI Explorer.</p>	
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring EVPN Routing Instances

- [Configuring EVPN Routing Instances on EX9200 Switches on page 25](#)
- *Configuring VPLS Routing Instances*
- [interface on page 277](#)
- [mac-table-size on page 288](#)

ip-prefix-routes

Syntax	<pre>ip-prefix-routes { advertise (direct-nexthop gateway-address); encapsulation (vxlan mpls); <export routing-policy-name>; gateway-interface interface-name; vni number; }</pre>
Hierarchy Level	<pre>[edit logical-systems logical-system-name routing-instances routing-instance-name protocols evpn] [edit routing-instances routing-instance-name protocols evpn]</pre>
Release Information	<p>Statement introduced in Junos OS Release 15.1X53-D60 and Junos OS Release 17.2R1 for QFX10000 switches.</p> <p>Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.</p> <p>Statement introduced in Junos OS Release 17.4R1 for QFX5110 switches.</p> <p>Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.</p>
Description	<p>In an Ethernet VPN (EVPN) environment, enable the switch to advertise the IP prefix associated with a specified customer domain as a type-5 route to remote data centers or in a metro transport network. You use this feature when the Layer 2 domain does not exist at the remote data centers or metro network peering points.</p> <p>Two models for implementing type-5 routes are supported:</p> <ul style="list-style-type: none">• Pure type-5 route without overlay next hop and type-2 route• Type-5 route with gateway IRB interface as overlay next hop and type-2 route <p>A pure type-5 route advertises the summary IP prefix and includes a BGP extended community called a router MAC, which is used to carry the MAC address of the sending switch and to provide next-hop reachability information for the prefix. In contrast, a standard type-5 route requires a gateway IP address as a next-hop overlay and a supporting type-2 route to provide recursive route resolution.</p> <p>On QFX5110 and QFX10000 switches, only fully resolved next-hop—that is, EVPN pure type-5—routes are currently supported. QFX10000 switches support only EVPN Virtual Extensible LAN (VXLAN). MPLS encapsulation is not supported.</p> <p>On QFX5110 switches, you must also configure the <i>overlay-ecmp</i> statement at the [edit forwarding-options vxlan-routing] hierarchy level with pure type-5 routes in an overlay EVPN-VXLAN network. We strongly recommend also configuring the overlay-ecmp statement whenever you enable pure type-5 routes on a QFX5110 switch. Configuring this statement causes the Packet Forwarding Engine to restart. Restarting the Packet Forwarding Engine interrupts all forwarding operations. Therefore, we strongly recommend configuring the overlay-ecmp statement before the EVPN-VXLAN network becomes operational.</p>

On EX9200 switches, both type-5 routes are supported. Only MPLS encapsulation is supported with pure type-5 routes. Both MPLS and VXLAN encapsulation are supported with the standard type-5 route with a gateway IP address as next-hop overlay.



CAUTION: Pure type-5 routing for EVPN-VXLAN was introduced in Junos OS Release 15.1X53-D30 for QFX10002 switches only. In that release, this statement is `ip-prefix-support forwarding-mode symmetric`. Starting with Junos OS Release 15.1X53-D60, the statement is `ip-prefix-routes advertise direct-nexthop`. Any configuration with the original `ip-prefix-support` statement is automatically upgraded to the new `ip-prefix-routes` statement when you upgrade to Junos OS Release 15.1X53-D60 or later.



NOTE: Pure type-5 routing is supported on all QFX10000 switches starting in Junos OS Release 15.1X53-D60.



NOTE: QFX10000 switches do not support advertising an IP prefix with a mask length of /32 as a pure type-5 route.

Options The **advertise**, **encapsulation** and **vni *number*** options are required. The **export *routing-policy-name*** option is optional.

advertise direct-nexthop—Enable the switch to send IP prefix information using an EVPN pure type-5 route, which includes a router MAC extended community used to send the MAC address of the switch. This router MAC extended community provides next-hop reachability without requiring an overlay next-hop or supporting type-2 route.



NOTE: For pure route type-5, QFX5110 and QFX10000 switches support only VXLAN encapsulation, and EX9200 switches support only MPLS encapsulation.

advertise gateway-address (EX9200 switches only)—Enable the switch to advertise a gateway address in exported IP prefix routes. This gateway address provides overlay next-hop reachability.



NOTE: You must also specify a gateway address by including the **gateway-interface *interface-name*** statement.

encapsulation (vxlan | mpls)—Specify to encapsulate forwarded traffic in VXLAN or MPLS for transmission to the remote data center.



NOTE: The same type of encapsulation must be used end to end. Only VXLAN encapsulation is supported on QFX10000 and QFX5110 switches.

export *routing-policy-name*—(Optional) Specify the name of the routing policy configured at the **[edit policy-options policy-statement *policy-statement-name*]** hierarchy level to apply to the routes for the specified customer domain. Applying an export policy allows you to further control the IP prefixes to advertise or to suppress through EVPN type-5 routes for each customer. You can apply a separate export routing policy to one or more customer domains. This allows each customer to each have its own policy.

gateway-interface *interface-name*—Specify the gateway interface to use as a next-hop overlay for a standard type-5 route. You must use this option in conjunction with the **advertise gateway-address** option.

vni *number*—Specify the identifier associated with a customer domain. Each customer domain must have a unique identifier.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Understanding EVPN Pure Route Type-5 on QFX Switches*
- [EVPN Overview for Switches on page 3](#)
- *policy-statement*

l2-learning

Syntax

```
l2-learning {
    global-le-bridge-domain-aging-time;
    global-mac-ip-limit number;
    global-mac-ip-table-aging-time seconds;
    global-mac-limit limit;
    global-mac-statistics;
    global-mac-table-aging-time seconds;
    global-no-mac-learning;
    global-mac-move;
}
```

Hierarchy Level [edit protocols]

Release Information

Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D10 for QFX Series.
global-le-bridge-domain-aging-time option introduced in Junos OS Release 14.2R5 for the MX Series.
global-mac-ip-limit and **global-mac-ip-table-aging-time** options introduced in Junos OS Release 17.4R1 for MX Series routers and EX9200 switches.

Description (MX Series routers, EX Series switches, and QFX Series switches only) Configure Layer 2 address learning and forwarding properties globally.

The remaining statements are explained separately. See [CLI Explorer](#).

Options

global-le-bridge-domain-aging-time—Specify the aging time of LE bridge-domain. The MAC address is learnt after next hop(NH) and bridge-domain(BD), also called NHBD. This aging time delays the deletion of NHBD. Configuring lesser time, in seconds, results in faster deletion of NHBD.
Range: 120 to 1000000 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Understanding Layer 2 Learning and Forwarding*

mac-statistics


Syntax	mac-statistics;
Hierarchy Level	<p>[edit bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> switch-options],</p> <p>[edit logical-systems <i>logical-system-name</i> switch-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> switch-options],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols evpn],</p> <p>[edit switch-options],</p> <p>[edit switch-options],</p> <p>[edit vlans <i>vlan-name</i> switch-options]</p>
Release Information	<p>Statement introduced in Junos OS Release 8.4.</p> <p>Support for the switch-options statement added in Junos OS Release 9.2.</p> <p>Support for top-level configuration for the virtual-switch type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.</p> <p>Support for logical systems added in Junos OS Release 9.6.</p> <p>[edit switch-options] and [edit vlans <i>vlan-name</i> switch-options] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.</p> <p>Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.</p> <p>[edit switch-options] and [edit vlans <i>vlan-name</i> switch-options] hierarchy levels introduced in Junos OS Release 13.2 for the QFX Series.</p>
Description	(MX Series routers, EX Series switches, and QFX Series only) For bridge domains or VLANs, enable MAC accounting either for a specific bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port.
Default	disabled
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Understanding Layer 2 Learning and Forwarding for Bridge Domains</i> • <i>Layer 2 Learning and Forwarding for VLANs Overview</i> • <i>Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports</i>

- [Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port](#)
- [Configuring EVPN Routing Instances](#)
- [Configuring EVPN Routing Instances on EX9200 Switches on page 25](#)

mac-ip-table-size

Syntax	<code>mac-ip-table-size <i>number</i>;</code>
Hierarchy Level	[edit logical-systems <i>name</i> routing-instances <i>routing-instance-name</i> protocols evpn], [edit routing-instances <i>routing-instance-name</i> protocols evpn], [edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> bridge-options], [edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> switch-options]
Release Information	Statement introduced in Junos OS Release 17.4R1.
Description	<p>Limit the number of entries that can be added to the MAC-IP address bindings database. On MX routers and EX9200 switches, this limit can be applied to EVPN routing instances. On MX routers, you can also apply this limit to bridge-domains configured in a virtual switch routing instance. On EX9200 switches, this limit can also be applied to VLANs configured in a routing instance. When the specified maximum number of entries is reached, no additional entries are added to the MAC-IP bindings database.</p> <p>To apply this limit systemwide, use the global-mac-ip-limit statement at the [edit protocols l2-learning] hierarchy level.</p>
Options	<p><i>number</i>—Maximum number of entries that can be added to the MAC+IP table.</p> <p>Range: 16 through 1048575.</p> <p>Default: 8192</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression on page 14 • interface-mac-ip-limit on page 279

mac-table-size

Syntax	<pre>mac-table-size size { packet-action drop; }</pre>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols vpls], [edit routing-instances <i>routing-instance-name</i> protocols evpn], [edit routing-instances <i>routing-instance-name</i> protocols vpls]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 13.2 for EVPNs on MX 3D Series routers. Statement introduced in Junos OS Release 14.2 for EX Series switches.</p>
Description	Specify the size of the MAC address table.
Options	<p>size—Size of the MAC address table.</p> <p>Range:</p> <ul style="list-style-type: none"> • (M Series and T Series routers only) 16 through 65,536 MAC addresses • (MX Series routers only) 16 through 1,048,575 MAC addresses • (T4000 routers with Type 5 FPCs only) 16 through 262,143 MAC addresses
	<p> NOTE: Before modifying the size of the MAC address table (to 262,143 addresses), you must enable network services mode by including the enhanced-mode statement at the [edit chassis network-services] hierarchy level and then reboot the router.</p>
	<p>Default: 512 MAC addresses</p> <p>The remaining statement is explained separately. See CLI Explorer.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • <i>Configuring EVPN Routing Instances</i> • Configuring EVPN Routing Instances on EX9200 Switches on page 25 • <i>Configuring VPLS Routing Instances</i> • <i>Configuring Improved VPLS MAC Address Learning on T4000 Routers with Type 5 FPCs</i> • <i>enhanced-mode</i> • evpn on page 267

mclag

Syntax	<pre>mclag { bgp-peer <i>ip-address</i>; }</pre>
Hierarchy Level	[edit routing-instances <i>name</i> protocols evpn]
Release Information	Statement introduced in Junos OS Release 17.4R1 on MX Series routers, EX Series switches, and Junos Fusion Enterprise.
Description	<p>Configure parameters that enable the interworking of Ethernet VPN-MPLS (EVPN-MPLS) with a Junos Fusion Enterprise or a multichassis link aggregation group (MC-LAG) topology.</p> <p>The remaining statements are explained separately. See CLI Explorer.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Understanding EVPN-MPLS Interworking with Junos Fusion Enterprise and MC-LAG on page 211

multicast-mode (EVPN)

Syntax	<code>multicast-mode client ingress-replication;</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols], [edit protocols evpn], [edit routing-instances <i>routing-instance-name</i> protocols evpn]
Release Information	Statement introduced in Junos OS Release 14.1X53-D30 for QFX Series switches. Statement introduced in Junos OS Release 17.3R1 for EX9200 switches. Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.
Description	Configure the multicast server mode for delivering traffic and packets for Ethernet VPN (EVPN). This statement is required for a VXLAN EVPN instance.



NOTE: If you configure the `multicast-mode` statement, then you must also configure the [encapsulation vxlan](#) statement.

Options	client —Use the client as the multicast mode for delivering traffic and multicast packets across routers and switches. ingress-replication —Use ingress replication as the multicast mode for delivering broadcast, unknown unicast, and multicast (BUM) traffic and multicast packets across routers and switches. Default: ingress-replication
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Understanding EVPN with VXLAN Data Plane Encapsulation on page 5• EVPN Over VXLAN Encapsulation Configuration Overview for QFX5100 Series• Using a Default Layer 3 Gateway to Route Traffic Between Virtual Networks in an EVPN-VXLAN Topology• Example: Configuring an EVPN Control Plane and VXLAN Data Plane on page 29

no-arp-suppression

Syntax	no-arp-suppression;
Hierarchy Level	<p>[edit bridge-domains <i>bridge-domain-name</i>] [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols] [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i>] [edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i>] [edit vlans <i>vlan-name</i>] [edit routing-instances <i>instance-name</i> protocols evpn] [edit routing-instances <i>instance-name</i> bridge-domains <i>domain-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 17.2. Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.</p>
Description	<p>Disable the suppression of Address Resolution Protocol (ARP) requests from a customer edge (CE) device to a provider edge (PE) device that acts as ARP proxy in an Ethernet VPN-MPLS (EVPN-MPLS) or Ethernet VPN-Virtual Extensible LAN (EVPN-VXLAN) environment.</p> <p>When disabling ARP suppression, be aware of the following implications;</p> <ul style="list-style-type: none"> • If the PE device does not find the MAC-IP address binding in its database, it cannot forward the ARP request. Instead, the device discards the ARP request, and the packets flood through the Layer 2 domain because the ARP request is considered to be Layer 2 broadcast traffic. Therefore, we recommend that ARP suppression remains enabled. • Disabling the suppression of ARP packets on a PE device essentially disables the proxy ARP functionality. <p>Therefore, we recommend that ARP suppression remains enabled.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • EVPN Proxy ARP and ARP Suppression and Network Discovery Protocol and Network Discovery Protocol Suppression on page 14

no-mac-learning

Syntax no-mac-learning;

Hierarchy Level [edit bridge-domains *bridge-domain-name* bridge-options],
 [edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
 [edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options],
 [edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options],
 [edit logical-systems *logical-system-name* switch-options],
 [edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
 [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
 [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
 [edit routing-instances *routing-instance-name* protocols *evpn*],
 [edit routing-instances *routing-instance-name* protocols *evpn* *interface* *interface-name*],
 [edit routing-instances *routing-instance-name* switch-options],
 [edit switch-options],
 [edit switch-options],
 [edit switch-options interface *interface-name*],
 [set vlans *vlan-name* switch-options]

Release Information Statement introduced in Junos OS Release 8.4.
 Support for the **switch-options** statement added in Junos OS Release 9.2.
 Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or bridge domain configured within a virtual switch.
 Support for logical systems added in Junos OS Release 9.6.
[edit switch-options], **[edit switch-options interface *interface-name*]**, **[edit vlans *vlan-name* switch-options]**, and **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy levels introduced in Junos OS Release 12.3 R2 for EX Series switches.
 Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.
 Hierarchy levels **[edit switch-options interface *interface-name*]** and **[edit vlans *vlan-name* switch-options]** introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description For MX Series routers and EX Series switches, disable MAC learning for a virtual switch, for a bridge domain or VLAN, for a specific logical interface in a bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port. On platforms that support EVPNs, you can disable MAC learning on an EVPN.



NOTE: When MAC learning is disabled for a VPLS routing instance, traffic is not load-balanced and only one of the equal-cost next hops is used.

Default	MAC learning is enabled.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • <i>Configuring EVPN Routing Instances</i> • Configuring EVPN Routing Instances on EX9200 Switches on page 25 • <i>Understanding Layer 2 Learning and Forwarding for Bridge Domains</i> • <i>Layer 2 Learning and Forwarding for VLANs Overview</i> • <i>Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports</i> • <i>Understanding Bridging and VLANs on EX Series Switches</i> • <i>Understanding Q-in-Q Tunneling on EX Series Switches with ELS Support</i>

packet-action

Syntax `packet-action action;`

Hierarchy Level [edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
 [edit bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
 [edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
 [edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface *interface-name* interface-mac-limit *limit*],
 [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface-mac-limit *limit*],
 [edit logical-systems *logical-system-name* switch-options interface-mac-limit *limit*],
 [edit protocols [l2-learning](#) global-mac-limit *limit*],
 [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
 [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
 [edit routing-instances *routing-instance-name* protocols evpn [interface-mac-limit \(VPLS\)](#)],
 [edit routing-instances *routing-instance-name* protocols evpn interface *interface-name* [interface-mac-limit \(VPLS\)](#)],
 [edit routing-instances *routing-instance-name* protocols evpn [mac-table-size limit](#)],
 [edit routing-instances *routing-instance-name* switch-options interface *interface-name* interface-mac-limit *limit*],
 [edit routing-instances *routing-instance-name* switch-options interface-mac-limit *limit*],
 [edit switch-options interface *interface-name* interface-mac-limit *limit*],
 [edit switch-options interface-mac-limit *limit*],
 [edit switch-options interface *interface-name* interface-mac-limit *limit*],
 [edit switch-options interface-mac-limit *limit*],
 [edit switch-options mac-table-size *limit*],
 [edit switch-options interface *interface-name* interface-mac-limit *limit*],
 [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*],
 [edit vlans *vlan-name* switch-options interface-mac-limit *limit*],
 [edit vlans *vlan-name* switch-options mac-table-size *limit*]
 [edit vlans *vlan-name* switch-options interface-mac-limit *limit*],
 [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*],
 [edit vlans *vlan-name* switch-options mac-table-size *limit*]

Release Information Statement introduced in Junos OS Release 8.4.
 Support for the **switch-options** statement added in Junos OS Release 9.2.
 Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy

supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.

Support for logical systems added in Junos OS Release 9.6.

[edit switch-options interface *interface-name* interface-mac-limit *limit*], [edit switch-options interface-mac-limit *limit*], [edit switch-options mac-table-size *limit*], [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*], [edit vlans *vlan-name* switch-options interface-mac-limit *limit*], and [edit vlans *vlan-name* switch-options mac-table-size *limit*] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.

Support for EVPNs introduced in Junos OS Release 13.2 on MX Series 3D Universal Edge Routers.

Support at the [edit switch-options interface *interface-name* interface-mac-limit *limit*] hierarchy level and hierarchy levels under [edit vlans *vlan-name*] introduced in Junos OS Release 13.2X50-D10 for EX Series switches and Junos OS Release 13.2 for the QFX Series.

Description Specify the action taken when packets with new source MAC addresses are received after the MAC address limit is reached. If this statement is not configured, packets with new source MAC addresses are forwarded by default.

Default



NOTE: On a QFX Series Virtual Chassis, if you include the shutdown option at the [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit packet-action] hierarchy level and issue the commit operation, the system generates a commit error. The system does not generate an error if you include the shutdown option at the [edit switch-options interface *interface-name* interface-mac-limit packet-action] hierarchy level.

Disabled. The default is for packets for new source MAC addresses to be forwarded after the MAC address limit is reached.

Options **drop**—Drop packets with new source MAC addresses, and do not learn the new source MAC addresses.



NOTE: On QFX10000 switches, if you include the drop option, you cannot configure unicast reverse-path forwarding (URFP) on integrated routing and bridging (IRB) and MAC limiting on the same interface. If you have an MC-LAG configuration, you cannot configure MAC limiting on the interchassis link (ICL) interface.

drop-and-log—(EX Series switches and QFX Series only) Drop packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

log—(EX Series switches and QFX Series only) Hold packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

none—(EX Series switches and QFX Series only) Forward packets with new source MAC addresses, and learn the new source MAC address.

shutdown—(EX Series switches and QFX Series only) Disable the specified interface, and generate an alarm, an SNMP trap, or a system log entry.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

- Related Documentation**
- *Configuring EVPN Routing Instances*
 - [Configuring EVPN Routing Instances on EX9200 Switches on page 25](#)
 - *Configuring MAC Limiting (CLI Procedure)*
 - *Configuring Persistent MAC Learning (CLI Procedure)*
 - *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
 - *Layer 2 Learning and Forwarding for VLANs Overview*
 - *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
 - *Layer 2 Learning and Forwarding for VLANs Overview*
 - *Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port*

routing-instances

Syntax	<code>routing-instances <i>routing-instance-name</i> { ... }</code>
Hierarchy Level	[edit], [edit logical-systems <i>logical-system-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Configure an additional routing entity for a router or switch. You can create multiple instances of BGP, IS-IS, OSPF, OSPF version 3 (OSPFv3), and RIP for a router or switch.
Default	Routing instances are disabled for the router or switch.
Options	<i>routing-instance-name</i> —Name of the routing instance, a maximum of 31 characters. The remaining statements are explained separately.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• <i>Configuring EVPN Routing Instances</i>• <i>Configuring Routing Instances on PE Routers in VPNs</i>

switch-options

Syntax	<pre> switch-options { interface <i>interface-name</i> { interface-mac-limit <i>limit</i> { packet-action drop; } mac-pinning (EVPN Routing Instances) no-mac-learning; static-mac <i>static-mac-address</i> { vlan-id <i>number</i>; } } interface-mac-limit <i>limit</i> { packet-action drop; } mac-statistics; mac-ip-table-size <i>number</i>; mac-table-size <i>limit</i> { packet-action drop; } no-mac-learning; service-id <i>number</i>; vtep-source-interface } </pre>
Hierarchy Level	<pre> [edit], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i>], [edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i>], [edit vlans <i>vlan-name</i>] </pre>
Release Information	<p>Statement introduced in Junos OS Release 12.3R2 for EX Series switches and MX Series routers.</p> <p>Statement introduced in Junos OS Release 13.2 for the QFX Series.</p> <p>Statement (mac-pinning) introduced in Junos OS 16.2 for MX Series routers.</p> <p>mac-ip-table-size statement introduced in Junos OS 17.4 Release for MX Series routers and EX9200 switches.</p>
Description	<p>Configure Layer 2 learning and forwarding properties for a VLAN or a virtual switch.</p> <p>The remaining statements are explained separately. See CLI Explorer.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>

traceoptions (Protocols EVPN)

Syntax	<pre> traceoptions { file <i>filename</i> <files <i>number</i>> <size <i>size</i>> <world-readable no-world-readable>; flag <i>flag</i> <flag-modifier>; } </pre>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols] [edit routing-instances <i>routing-instance-name</i> protocols evpn]
Release Information	<p>Statement introduced in Junos OS Release 13.2 for MX 3D Series routers.</p> <p>Statement introduced in Junos OS Release 14.2 for EX Series switches.</p> <p>Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.</p>
Description	Trace traffic flowing through an EVPN routing instance.
Options	<p>file <i>filename</i>—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").</p> <p>files <i>number</i>—(Optional) Maximum number of trace files. When a trace file named <i>trace-file</i> reaches the maximum size as specified by the size option, it is renamed <i>trace-file.0</i>. When <i>trace-file</i> again reaches the maximum size, <i>trace-file.0</i> is renamed <i>trace-file.1</i> and <i>trace-file</i> is renamed <i>trace-file.0</i>. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.</p> <p>If you specify a maximum number of files, you also must specify a maximum file size with the size option.</p> <p>Range: 2 through 1000 files</p> <p>Default: 2 files</p> <p>flag <i>flag</i>—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can specify the following tracing flags:</p> <ul style="list-style-type: none"> • all—All EVPN tracing options • error—Error conditions • general—General events • mac-database—MAC route database in the EVPN routing instance • nlri—EVPN advertisements received or sent by means of BGP • normal—Normal events • oam—OAM messages • policy—Policy processing • route—Routing information • state—State transitions

- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing
- **topology**—EVPN topology changes caused by reconfiguration or advertisements received from other provider edge (PE) routers using BGP

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the following modifiers:

- **detail**—Provide detailed trace information.
- **disable**—Disable this trace flag.
- **receive**—Trace received packets.
- **send**—Trace sent packets.

no-world-readable—Do not allow any user to read the log file.

size size—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named **trace-file** reaches this size, it is renamed **trace-file.0**. When **trace-file** again reaches the maximum size, **trace-file.0** is renamed **trace-file.1** and **trace-file** is renamed **trace-file.0**. This renaming scheme continues until the maximum number of trace files (as specified by the **files** option) is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: **xk** to specify kilobytes, **xm** to specify megabytes, or **xg** to specify gigabytes

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—Allow any user to read the log file.

Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
---------------------------------	---

Related Documentation	<ul style="list-style-type: none">• Tracing EVPN Traffic and Operations on page 27
------------------------------	--

vlan-id (routing instance)

Syntax	<code>vlan-id (vlan-id all none);</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i>], [edit routing-instances <i>routing-instance-name</i>] [edit routing-instances <i>routing-instance-name</i> instance-type]
Release Information	Statement introduced in Junos OS Release 13.2. Statement introduced in Junos OS Release 14.2 for EX Series switches. Statement introduced in Junos OS Release 17.1 for the QFX Series.
Description	Specify 802.1Q VLAN tag IDs to a routing instance.
Options	<p>vlan-id—A valid VLAN identifier.</p> <p>Range: For 4-port Fast Ethernet PICs, 512 through 1023. For 1-port and 10-port Gigabit Ethernet PICs configured to handle VPLS traffic, 512 through 4094.</p> <p>all—Include all VLAN identifiers specified on the logical interfaces included in the routing instance.</p> <p>none—Include no VLAN identifiers for the routing instance.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring EVPN Routing Instances • Configuring EVPN Routing Instances on EX9200 Switches on page 25

vlan

Syntax	<pre> vlan { vlan-name { description <i>text-description</i>; dot1q-tunneling { customer-vlans (<i>id</i> <i>range</i>) layer2-protocol-tunneling all <i>protocol-name</i> { drop-threshold <i>number</i>; shutdown-threshold <i>number</i>; } } filter input <i>filter-name</i>; filter output <i>filter-name</i>; interface <i>interface-name</i> { egress; ingress; mapping (native (push swap) policy tag (push swap)); pvlan-trunk; } isolation-id <i>id-number</i>; l3-interface <i>l3-interface-name.logical-interface-number</i>; l3-interface-ingress-counting <i>layer-3-interface-name</i>; mac-limit <i>limit</i> action <i>action</i>; mac-table-aging-time <i>seconds</i>; no-local-switching; no-mac-learning; primary-vlan <i>vlan-name</i>; vlan-id <i>number</i>; vlan-prune; vlan-range <i>vlan-id-low-vlan-id-high</i>; } } </pre>
Hierarchy Level	<pre> [edit], [edit routing-instances <i>routing-instance-name</i>] </pre>
Release Information	Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>Configure VLAN properties on EX Series switches. The following configuration guidelines apply:</p> <ul style="list-style-type: none"> Only private VLAN (PVLAN) firewall filters can be used when the VLAN is enabled for Q-in-Q tunneling. An S-VLAN tag is added to the packet if the VLAN is Q-in-Q-tunneled and the packet is arriving from an access interface. You cannot use a firewall filter to assign an integrated routing and bridging (IRB) interface or a routed VLAN interface (RVI) to a VLAN. VLAN assignments performed using a firewall filter override all other VLAN assignments.

Options *vlan-name*—Name of the VLAN. The name can include letters, numbers, hyphens (-), and periods (.) and can contain up to 255 characters long.

The remaining statements are explained separately. See [CLI Explorer](#).

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- [Configuring VLANs for EX Series Switches \(CLI Procedure\)](#)
- [Configuring VLANs for EX Series Switches with ELS Support \(CLI Procedure\)](#)
- [Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support \(CLI Procedure\)](#)
- [Configuring Integrated Routing and Bridging Interfaces on Switches \(CLI Procedure\)](#)
- [Understanding Bridging and VLANs on EX Series Switches](#)

vni-options

Syntax

```
vni-options vni vxlan-network-identifier {
    designated-forwarder-election-hold-time seconds;
    vrf-target {
        community;
        auto;
        import community-name;
        export community-name;
    }
}
```

Hierarchy Level [edit protocols evpn]
[edit routing-instances *routing-instance-name* protocols evpn]

Release Information Statement introduced in Junos OS Release 14.1X53-D30 for QFX Series switches.
Statement introduced in Junos OS Release 16.1 for MX Series.
Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.

Description Configure a designated forwarder election hold time and specific route targets (RTs) for each VXLAN network identifier (VNI).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- [EVPN Multihoming Overview on page 119](#)
- [Understanding EVPN with VXLAN Data Plane Encapsulation on page 5](#)
- [Example: Configuring an EVPN Control Plane and VXLAN Data Plane on page 29](#)
- [extended-vni-list on page 270](#)

vni

Syntax	vni [1–16777214]
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> vxlan] [edit vlans <i>vlan-name</i> vxlan]
Release Information	Statement introduced in Junos OS Release 14.1R2 for MX Series routers. Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.
Description	Assign a numeric value to identify a Virtual Extensible LAN (VXLAN). All members of a VXLAN must use the same VNI.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• <i>Understanding VXLANs</i>• <i>Manually Configuring VXLANs on a QFX5100 Switch</i>• <i>Examples: Manually Configuring VXLANs on QFX Switches</i>

CHAPTER 10

Operational Commands

- `clear evpn duplicate-mac-suppression`
- `clear evpn nd-table`
- `clear ethernet-switching evpn nd-statistics`
- `clear ethernet-switching evpn nd-table`
- `show ethernet-switching evpn nd-statistics`
- `show evpn arp-table`
- `show evpn database`
- `show evpn flood`
- `show evpn instance`
- `show evpn ip-prefix-database`
- `show evpn l3-context`
- `show evpn mac-table`
- `show evpn nd-table`
- `show evpn peer-gateway-macs`
- `show vlans evpn nd-table`

clear evpn duplicate-mac-suppression

Syntax	<code>clear evpn duplicate-mac-suppression</code> <code><instance <i>instance</i> l2-domain-id <i>l2-domain-id</i> logical-system (all <i>logical-system-name</i>)</code> <code> mac-address <i>mac-address</i>></code>
Release Information	Command introduced in Junos OS Release 17.4 on MX Series routers, EX Series switches, and QFX Series switches.
Description	Clear suppressed duplicate MAC address in the EVPN network.
Options	<p>instance <i>instance</i>—(Optional) Clear suppressed MAC address for a specific routing instance. On MX Series routers, the routing instance can be an EVPN instance or virtual-switch instance. On QFX Series switches, the routing instance can be an EVPN instance, virtual-switch instance or an implicit default-switch instance.</p> <p>l2-domain-id <i>l2-domain-id</i>—(Optional) Clear suppressed MAC address for a specific L2 domain.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Clear suppressed MAC address on all logical systems or on a specific logical system.</p> <p>mac-address <i>mac-address</i>—(Optional) Clear a specific suppressed MAC address.</p>
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none">• Overview of MAC Mobility on page 207• Changing Duplicate MAC Address Detection Settings on page 209

clear evpn nd-table

Syntax	clear evpn nd-table
Release Information	Command introduced in Junos OS Release 16.2. Command introduced in Junos OS Release 17.1 for MX series Routers. Command introduced in Junos OS Release 17.3R1 for EX series switches.
Description	Clear the Neighbor Discovery (ND) proxy table from the Ethernet VPN (EVPN). ND proxy is a kernel module that implements IPv6 Neighbor Discovery proxying over Ethernet-like access networks.
Options	none —Clear learned ND proxy tables from the EVPN.
Required Privilege Level	clear
List of Sample Output	clear evpn nd-table on page 307
Output Fields	When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear evpn nd-table

```
user@host> clear evpn nd-table
```

clear ethernet-switching evpn nd-statistics

Syntax	clear ethernet-switching evpn nd-statistics
Release Information	Command introduced in Junos OS Release 17.3R1 for EX series switches.
Description	Clear the Neighbor Discovery (ND) proxy table statistics for integrated routing and bridging (IRB) interfaces participating in the Ethernet VPN (EVPN). ND proxy is a kernel module that implements IPv6 Neighbor Discovery proxying over Ethernet-like access networks.
Options	none —Clear the ND proxy table statistics for IRB interfaces participating in the EVPN.
Required Privilege Level	clear
List of Sample Output	clear ethernet-switching evpn nd-statistics on page 308
Output Fields	When you enter this command, the ND proxy table statistics for IRB interfaces participating in the EVPN are cleared.

Sample Output

clear ethernet-switching evpn nd-statistics

```
user@host> clear ethernet-switching evpn nd-statistics
```

clear ethernet-switching evpn nd-table

Syntax	clear ethernet-switching evpn nd-table
Release Information	Command introduced in Junos OS Release 17.3R1 for the EX Series.
Description	Clear the Neighbor Discovery (ND) proxy table from the Ethernet VPN (EVPN) for integrated routing and bridging (IRB) interfaces. This command applies to EVPN instances of type virtual-switch .
Options	none —Clear information about the ND proxy tables for an EVPN.
Required Privilege Level	clear
List of Sample Output	clear ethernet-switching evpn nd-table on page 309
Output Fields	When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear ethernet-switching evpn nd-table

```
user@host> clear ethernet-switching evpn nd-table
```

show ethernet-switching evpn nd-statistics

Syntax `show ethernet-switching evpn nd-statistics`
`<brief | count | detail | extensive | summary | display xml>`

Release Information Command introduced in Junos OS Release 17.3R1 for EX series switches.

Description Show Neighbor Discovery (ND) proxy table statistics for integrated routing and bridging (IRB) interfaces participating in the Ethernet VPN (EVPN). ND proxy is a kernel module that implements IPv6 Neighbor Discovery proxying over Ethernet-like access networks.

Options **none**—Display information about ND proxy table statistics for all IRB interfaces participating in the EVPN.

brief | count | detail | extensive | summary | display xml—(Optional) Display the specified level of output.

Required Privilege Level view

List of Sample Output [show ethernet-switching evpn nd-statistics on page 311](#)

Output Fields [Table 14 on page 310](#) lists the output fields for the **show ethernet-switching evpn nd-statistics** command. Output fields are listed in the approximate order in which they appear.

Table 14: show ethernet-switching evpn nd-statistics Output Fields

Field Name	Field Description	Level of Output
Interface	IRB interface on which the statistics has been applied	All levels
EVPN	Name of the EVPN	All levels
Bridge domain	Name of the bridge domain for the EVPN	All levels
ND routes add received	Total number of additional ND routes received	All levels
ND routes del received	Total number of deleted ND routes received	All levels
ND routes dropped	Total number of ND routes dropped	All levels
MAC+IPv6s sent to peer	Total number of MAC and IPv6 addresses sent to peer device	All levels

Sample Output

show ethernet-switching evpn nd-statistics

```
user@host> show ethernet-switching evpn nd-statistics interface irb.0

Interface : irb.0      EVPN : evpn1      Bridge domain: vlan10
ND routes add received      : 2
ND routes del received      : 0
ND routes dropped           : 0
MAC+IPv6s sent to peer      : 3
```

show evpn arp-table

Syntax	show evpn arp-table <address> <brief count detail extensive> <instance <i>instance-name</i> >
Release Information	Command introduced in Junos OS Release 15.1 for EX Series switches.
Description	Show Ethernet VPN (EVPN) Address Resolution Protocol (ARP) entries associated with learned MAC addresses.
Options	<p>none—Display brief information about the EVPN ARP table.</p> <p>address—(Optional) Display ARP information for the specified MAC address.</p> <p>brief count detail extensive—(Optional) Display the specified level of output.</p> <p>instance <instance-name>—oot(Optional) Display ARP information for the specified routing instance .</p>
Required Privilege Level	view
List of Sample Output	show evpn arp-table on page 313 show evpn arp-table 00:05:86:a0:dc:f0 (MAC address) on page 313 show evpn arp-table brief on page 313 show evpn arp-table detail on page 313 show evpn arp-table count on page 313 show evpn arp-table extensive on page 313 show evpn arp-table instance evpn1 on page 314
Output Fields	Table 15 on page 312 lists the output fields for the show evpn arp-table command. Output fields are listed in the approximate order in which they appear.

Table 15: show evpn arp-table Output Fields

Field Name	Field Description	Level of Output
INET address	The INET address related to the INET entries that are added to the ARP table.	All levels
MAC address	MAC addresses learned through ARP.	brief, detail, extensive, instance, mac-address,,
Logical Interface	Logical interface associated with the routing instance in which the ARP INET address is learned.	brief, instance, mac-address,,

Table 15: show evpn arp-table Output Fields (*continued*)

Field Name	Field Description	Level of Output
Routing instance	Routing instance in which the ARP INET address is learned.	all levels
Bridging domain	Bridging domain in which the ARP INET address is learned.	all levels
Learning interface	Interface on which the ARP INET address is learned.	detail, extensive
Count	Indicates the number of ARP INET addresses learned in a routing instance in a bridge domain.	count

Sample Output

show evpn arp-table

```
user@host> show evpn arp-table
INET          MAC          Logical      Routing      Bridging
address       address      interface    instance     domain
203.0.113.2   00:05:86:a0:dc:f0  irb.0        evpn1        __evpn1__
```

show evpn arp-table 00:05:86:a0:dc:f0 (MAC address)

```
user@host> show evpn arp-table 00:05:86:a0:dc:f0
INET          MAC          Logical      Routing      Bridging
address       address      interface    instance     domain
203.0.113.2   00:05:86:a0:dc:f0  irb.0        evpn1        __evpn1__
```

show evpn arp-table brief

```
user@host> show evpn arp-table brief
INET          MAC          Logical      Routing      Bridging
address       address      interface    instance     domain
203.0.113.2   00:05:86:a0:dc:f0  irb.0        evpn1        __evpn1__
```

show evpn arp-table detail

```
user@host> show evpn arp-table detail

INET address: 203.0.113.2
MAC address: 00:05:86:a0:dc:f0
Routing instance: evpn1
Bridging domain: __evpn1__
Learning interface: irb.0
```

show evpn arp-table count

```
user@switch> show evpn arp-table count
1 ARP INET addresses learned in routing instance evpn1 bridge domain __evpn1__
```

show evpn arp-table extensive

```
user@host> show evpn arp-table extensive

INET address: 203.0.113.2
```

```

MAC address: 00:05:86:a0:dc:f0
Routing instance: evpn1
  Bridging domain: __evpn1__
  Learning interface: irb.0

```

show evpn arp-table instance evpn1

```

user@host> show evpn arp-table instance evpn1
INET          MAC          Logical      Routing      Bridging
address       address       interface    instance     domain
203.0.113.2   00:05:86:a0:dc:f0  irb.0        evpn1        __evpn1__

```

show evpn database

Syntax	<pre>show evpn database <extensive> <instance <i>instance-name</i>> <interface <i>interface-name</i>> <logical-system <i>logical-system-name</i>> <mac-address <i>address</i>> <neighbor <i>neighbor-name</i>> <origin <i>origin-name</i>> <state <i>state-name</i>> <vlan-id <i>vlan-id</i>></pre>
Release Information	<p>Command introduced in Junos OS Release 14.2 for EX Series switches.</p> <p>Command introduced in Junos OS Release 16.1 for MX Series routers.</p> <p>Command introduced in Junos OS Release 17.1 on MX Series routers with MPCs.</p>
Description	Show Ethernet VPN (EVPN) database information.
Options	<p>none—Display brief information about the EVPN database.</p> <p>extensive —(Optional) Display detailed information about the EVPN database.</p> <p>instance <i>instance-name</i>—(Optional) Display MAC addresses from the specified routing instance.</p> <p>interface <i>interface-name</i>—(Optional) Display the MAC address learned from the specified interface.</p> <p>logical-system <i><logical-system-name></i>—(Optional) Display database information for the specified logical system or all logical systems.</p> <p>mac-address <i>address</i>—(Optional) Display the specified MAC address.</p> <p>neighbor <i>neighbor-name</i>—(Optional) Display the MAC address learned from the specified neighbor.</p> <p>origin <i>origin-name</i>—(Optional) Display the MAC address with the specified origin.</p> <p>state <i>state-name</i>—(Optional) Display the MAC address with the specified state.</p> <p>vlan-id <i>vlan-id</i>—(Optional) Display the MAC address with the specified VLAN.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 106
List of Sample Output	show evpn database on page 317

[show evpn database extensive on page 317](#)

[show evpn database extensive instance on page 318](#)

[show evpn database extensive \(Duplicate MAC Address\) on page 318](#)

Output Fields Table 16 on page 316 lists the output fields for the **show evpn database** command. Output fields are listed in the approximate order in which they appear.

Table 16: show evpn database Output Fields

Field Name	Field Description	Level of Output
Instance	Name of the routing instances.	All Levels
VNI	VXLAN network identifier.	All Levels
MAC address	MAC address of the routing instance.	All Levels
Origin	Specific origin of the MAC address.	All Levels
Timestamp	Month, day and time of generated command output.	All Levels
IP address	IP address.	All Levels
Prefix	Prefix address of the VXLAN network identifier.	extensive
Source	Source address of the VXLAN network identifier.	extensive
Rank	Level number of the VXLAN network identifier.	extensive
Status	Status of the of the VXLAN network identifier.	extensive
Remote origin	Specific remote origin of the VXLAN network identifier.	extensive
Mobility History	Information on a MAC address mobility history: <ul style="list-style-type: none"> • Mobility event time—Date and time of the MAC mobility event • Type—Origin of the learned MAC address (Local or Remote). • Source—Source of the learned MAC address (Local Interface, Remote PE IP address, or ESI) • Seq num—Sequence number associated with the mobility event 	extensive
Mac label	Label advertised by the remote PE device when forwarding unicast traffic. In EVPN-MPLS, it is the MPLS label and in EVPN-VXLAN, it is the VNI .	extensive
Mobility Sequence number	Current sequence number associated with the MAC address. The minimum origin address is the IP address of the PE device with the lowest IP address that is sending a MAC advertisement.	extensive

Table 16: show evpn database Output Fields (*continued*)

Field Name	Field Description	Level of Output
State	Information collected in the EVPN database from different flag states including Duplicate-Detected, Local-Pinned, and Remote-Pinned.	extensive
IP address	IPv4 or IPv6 address for the MAC address.	extensive
L3 route	Route installed on the EVPN IRB interface.	extensive
L3 context	Name of the routing instance that has the Layer 3 routes installed for an EVPN IRB interface, typically a virtual routing and forward (VRF) routing instance.	extensive

Sample Output

show evpn database

```

user@host> show evpn database
Instance: bd_red
VNI  MAC address      Origin      Timestamp      IP address
50    1a:1b:1b:1b:1b:1b  192.0.2.3   Jun 12 21:57:17
50    1a:1c:1c:1c:1c:1c  ge-2/3/0.0  Jun 12 22:05:37
50    b0:c6:9a:e9:cd:d8  192.0.2.3   Jun 12 21:57:17
50    b0:c6:9a:ea:87:42  ge-2/3/0.0  Jun 12 22:05:38

Instance: evpn-0
VNI  MAC address      Origin      Timestamp      IP address
4000 1a:1a:1a:1a:01:01  ge-2/3/0.10 Jun 12 21:53:01
4000 1a:1a:1a:1a:01:02  ge-2/3/0.10 Jun 12 21:53:01

```

show evpn database extensive

```

user@host> show evpn database extensive
Instance: ALPHA

VN Identifier: 9100, Prefix: 10.0.0.0/24
Source: 10.255.0.2, Rank: 1, Status: Active
Timestamp: May 22 17:16:12 (0x555fc6cc)

VN Identifier: 9100, Prefix: 198.51.100.0/24
Source: 05:00:00:00:64:00:00:23:8c:00, Rank: 1, Status: Active
Remote origin: 10.255.0.2
Remote origin: 10.255.0.3
Timestamp: May 22 17:14:20 (0x555fc65c)

VN Identifier: 9100, Prefix: 192.0.2.0/24
Source: irb.0, Rank: 1, Status: Active
Timestamp: May 22 17:16:12 (0x555fc6cc)

VN Identifier: 9100, Prefix: 203.0.113.0/24
Source: 10.255.0.3, Rank: 1, Status: Active
Timestamp: May 22 17:16:12 (0x555fc6cc)

```

show evpn database extensive instance

```
user@PE1> show evpn database instance ALPHA mac-address 00:00:00:00:00:01 extensive
Instance: ALPHA
```

```
VLAN ID: 100, MAC address: 00:00:00:00:00:01
Nexthop ID: 1048575
Mobility history
  Mobility event time      Type      Source                               Seq
num
  Jul 10 16:26:14.920136 Remote  10.255.0.3                          13
  Jul 10 16:26:16.174769 Local   ge-0/0/2.0                         14
  Jul 10 16:26:17.868187 Remote  10.255.0.3                          15
  Jul 10 16:26:19.129879 Local   ge-0/0/2.0                         16
  Jul 10 16:26:25.972747 Remote  10.255.0.3                          17
Source: 10.255.0.3, Rank: 1, Status: Active
MAC label: 299776
Mobility sequence number: 17 (minimum origin address 10.255.0.3)
Timestamp: Jul 10 16:26:25 (0x59640d21)
State: <Remote-To-Local-Adv-Done>
IP address: 10.0.0.3
  L3 route: 10.0.0.3/32, L3 context: DELTA (irb.0)
```

show evpn database extensive (Duplicate MAC Address)

```
user@PE1> show evpn database instance ALPHA mac-address 00:00:00:00:00:012extensive
Instance: ALPHA
```

```
VLAN ID: 100, MAC address: 00:00:00:00:00:02
State: 0x1 <Duplicate-Detected>
Mobility history
  Mobility event time      Type      Source                               Seq
num
  Aug 03 17:22:28.585619 Local   ge-0/0/2.0                         31
  Aug 03 17:22:30.307198 Remote  10.255.0.3                          32
  Aug 03 17:22:37.611786 Local   ge-0/0/2.0                         33
  Aug 03 17:22:39.289357 Remote  10.255.0.3                          34
  Aug 03 17:22:45.609449 Local   ge-0/0/2.0                         35
Source: ge-0/0/2.0, Rank: 1, Status: Active
Mobility sequence number: 35 (minimum origin address 10.255.0.2)
Timestamp: Aug 03 17:22:44 (0x5983be54)
State: <Local-MAC-Only Local-To-Remote-Adv-Allowed>
MAC advertisement route status: Not created (duplicate MAC suppression)
IP address: 10.0.0.2
Source: 10.255.0.3, Rank: 2, Status: Inactive
MAC label: 300176
Mobility sequence number: 34 (minimum origin address 10.255.0.3)
Timestamp: Aug 03 17:22:39 (0x5983be4f)
State: <>
MAC advertisement route status: Not created (inactive source)
IP address: 10.0.0.3
```

show evpn flood

Syntax	<pre>show evpn flood event-queue <instance <i>instance-name</i>> <logical-system <i>logical-system-name</i>> <route <i>route-name</i>></pre>
Release Information	Command introduced in Junos OS Release 14.2 for EX Series switches.
Description	Show Ethernet VPN (EVPN) flooding information.
Options	<p>none—Display brief information about EVPN flooding.</p> <p>brief detail extensive summary—(Optional) Display the specified level of output.</p> <p>event-queue—(Optional) Display the queue of pending EVPN flood events.</p> <p>instance <i>instance-name</i>—(Optional) Display flooding information for the specified routing instance.</p> <p>logical-system <i><logical-system-name></i>—(Optional) Display flooding information for the specified logical system or all logical systems.</p> <p>route <i>route-name</i>—(Optional) Display flooding information for the specified route.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 106

show evpn instance

Syntax	<code>show evpn instance</code> <code><brief extensive></code> <code><backup-forwarder></code> <code><designated-forwarder></code> <code><esi esi></code> <code><instance-name></code> <code><neighbor neighbor-address></code>
Release Information	Command introduced in Junos OS Release 14.1 for MX series Routers. Command introduced in Junos OS Release 14.2 for EX Series switches. Command introduced in Junos OS Release 17.4R1 for QFX Series switches.
Description	Show Ethernet VPN (EVPN) routing instance information.
Options	none —Display brief information about the EVPN routing instance. brief extensive —(Optional) Display the specified level of output. backup-forwarder —(Optional) Display IP addresses of all the backup designated forwarders for the Ethernet segment. designated-forwarder —(Optional) Display the IP address of the designated forwarder for the Ethernet segment. esi esi —(Optional) Display brief information about the routing instance associated with the specified Ethernet segment identifier (ESI) value. instance-name —(Optional) Display information about the specified routing instance. neighbor neighbor-address —(Optional) Display the IP address of the EVPN neighbor.
Required Privilege Level	view
List of Sample Output	show evpn instance brief on page 323 show evpn instance on page 323 show evpn instance extensive (In Junos OS Release 16.1 and earlier) on page 324 show evpn instance extensive (In Junos OS Release 16.2 and later) on page 325 show evpn instance extensive (Preference-based DF Election) on page 326 show evpn instance extensive (Duplicate MAC Address) on page 328 show evpn instance extensive (Protected Interface) on page 328 show evpn instance esi backup-forwarder (Instance Name with Ethernet Segment Identifier) on page 328 show evpn instance esi designated-forwarder (Instance Name with Ethernet Segment Identifier) on page 328

Output Fields Table 17 on page 321 lists the output fields for the **show evpn instance** command. Output fields are listed in the approximate order in which they appear.

Table 17: show evpn instance Output Fields

Field Name	Field Description	Level of Output
Instance	Names of the routing instances.	All levels
Intfs	Total number of interfaces participating in each routing instance, and number of interfaces that are up.	brief
IRB intfs	Statistics on the number of integrated routing and bridging (IRB) interfaces for each routing instance: <ul style="list-style-type: none"> • Total—Total number of IRB interfaces. • Up—Number of active IRB interfaces. • Nbrs—Number of neighbor IRB interfaces. 	brief
MH ESIs	Number of Ethernet segments per routing instance that connect to a multihomed customer site.	brief
MAC addresses	Number of local and remote MAC addresses for each routing instance.	brief
Route Distinguisher	Unique route distinguisher associated with this routing instance.	none
VLAN ID	VLAN identifier.	none
Label allocation mode	Label allocation policy for the routing instance.	none
Encapsulation type	Encapsulation type for VXLAN EVPN instances (EVIs).	extensive
Per-instance MAC route label	Label of MAC route for each routing instance.	none
Duplicate MAC detection threshold	Number of MAC mobility events detected for a given MAC address before it is identified as a duplicate MAC address.	extensive
Duplicate MAC detection window	The time interval used in detecting a duplicate MAC address.	extensive
Duplicate MAC auto-recovery time	Length of time a device suppresses a duplicate MAC address. At the end of this duration, MAC address updates will resume.	extensive
DF Election preference	Preference value used for the designated forwarder (DF) election: <ul style="list-style-type: none"> • Highest preference—Default DF election preference. • Lowest preference—Based on the configuration of the designated-forwarder-preference-least statement at the [edit routing-instance routing-instance-name protocols evpn] hierarchy level. 	extensive

Table 17: show evpn instance Output Fields (*continued*)

Field Name	Field Description	Level of Output
Per-instance multicast route label	Label of the multicast route for each routing instance.	none
Total MAC addresses	Total number of local and remote MAC addresses received for each routing instance.	extensive
Default gateway MAC addresses	Number of local and remote MAC addresses serving as a default gateway in this routing instance.	extensive
Number of local interfaces	Number of local interfaces belonging to this routing instance.	extensive
Number of local interfaces up	Number of active local interfaces belonging to this routing instance.	extensive
Interface name	Name of interfaces that belong to this routing instance.	extensive
ESI	Ethernet segment identifier (ESI) value of the interfaces belonging to this routing instance.	extensive
DF Election Algorithm	DF election type: <ul style="list-style-type: none"> • MOD based—Default DF election algorithm based on the modulo operation. • Preference based—DF election based on the configured preference values for an ESI. 	extensive
Preference	Preference value for EVPN Multihoming DF election.	extensive
Mode	Mode of operation for each routing instance: <ul style="list-style-type: none"> • single-homed—Default mode and does not require Ethernet segment values to be configured. • single-active—EVPN active-standby multihoming mode of operation. 	extensive
SH label	Split horizon label used for the active-standby multihoming mode of operation.	extensive
Number of IRB interfaces	Number of IRB interfaces that belong to this routing instance.	extensive
Number of protect interfaces	Number of protect interfaces that belong to this routing instance.	extensive
Interface name	Name of the primary interface.	extensive
Protect interface name	Name of the protect interface.	extensive

Table 17: show evpn instance Output Fields (*continued*)

Field Name	Field Description	Level of Output
Status	Protected status of the primary interface. The status is either Protect-inactive or Protect-active.	extensive
L3 context	Names of routing instances that have the Layer 3 routes installed for an EVPN IRB interface, typically a virtual routing and forwarding (VRF) routing instance.	extensive
Number of neighbors	Number of neighbors connected to this routing instance and their IP addresses.	extensive
MAC address advertisement	Number of MAC address advertisements received from the neighbor.	extensive
MAC+IP address advertisement	Number of MAC and IP address advertisements received from the neighbor.	extensive
Inclusive multicast	Number of inclusive multicast routes received from the neighbor.	extensive
Ethernet auto-discovery	Number of autodiscovery routes per Ethernet segment received from the neighbor.	extensive
Number of ethernet segments	Total number of Ethernet segments for the routing instance.	extensive
Designated forwarder	IP address of the designated forwarder (DF) for the Ethernet segment.	extensive
Backup forwarder	IP address of all the backup designated forwarders or routers that are not designated forwarders for the Ethernet segment. NOTE: Immediately after an EVPN interface Ethernet segment identifier value is changed and the new configuration is committed, the Designated forwarder information changes to DF not elected yet and the backup forwarder information is not displayed until after the election is complete.	extensive

Sample Output

show evpn instance brief

```

user@host> show evpn instance brief

```

Instance	Intfs		IRB intfs			MH ESI	MAC addresses	
	Total	Up	Total	Up	Nbrs		Local	Remote
ALPHA	2	2	1	1	2	1	3	4
BETA	2	2	1	1	2	1	2	4
__default__evpn__	0	0	0	0	1	0	0	0

show evpn instance

```

user@host> show evpn instance

```

```

Instance: black
Route Distinguisher: 101:101
VLAN ID: 100
Label allocation mode: Per-instance
Per-instance MAC route label: 299776
Per-instance multicast route label: 299792
Number of local interfaces: 1
Number of local interfaces up: 1
  Interface name    Static MACs    ESI
  cbp-0.0           0            0
Number of neighbors: 1
  192.0.2.1
  Received routes
    MAC address advertisement:      1
    Ethernet auto-discovery:        0
    Inclusive multicast:            1

```

show evpn instance extensive (In Junos OS Release 16.1 and earlier)

```

user@host> show evpn instance extensive
Instance: ALPHA
Route Distinguisher: 10.255.0.1:100
Encapsulation type: VXLAN
Per-instance MAC route label: 300144
Per-instance multicast route label: 300160
MAC database status
  Total MAC addresses:      Local Remote
  Default gateway MAC addresses: 1      2
Number of local interfaces: 2 (2 up)
  Interface name  ESI                                     Mode      SH label
  ae0.0           00:11:22:33:44:55:66:77:88:99  single-active
  ge-0/0/2.0      00:00:00:00:00:00:00:00:00:00  single-homed
Number of IRB interfaces: 1 (1 up)
  Interface name  L3 context
  irb.0           DELTA
Number of neighbors: 2
  10.255.0.2
    Received routes
      MAC address advertisement:      2
      MAC+IP address advertisement:    3
      Inclusive multicast:            1
      Ethernet auto-discovery:        1
  10.255.0.3
    Received routes
      MAC address advertisement:      2
      MAC+IP address advertisement:    2
      Inclusive multicast:            1
      Ethernet auto-discovery:        0
Number of ethernet segments: 1
  ESI: 00:11:22:33:44:55:66:77:88:99
  Designated forwarder: 10.255.0.1
  Backup forwarder: 10.255.0.2

Instance: BETA
Route Distinguisher: 10.255.0.1:300
Encapsulation type: VXLAN
VLAN ID: 300
Per-instance MAC route label: 300176
Per-instance multicast route label: 300192
MAC database status
  Total MAC addresses:      Local Remote

```

```

    Default gateway MAC addresses:      1      2
Number of local interfaces: 2 (2 up)
  Interface name  ESI                               Mode          SH label
  ae1.0           00:00:00:00:00:00:00:00:00:00    single-homed
  ge-0/0/4.0      00:22:44:66:88:00:22:44:66:88    single-active
Number of IRB interfaces: 1 (1 up)
  Interface name  L3 context
  irb.1           DELTA
Number of neighbors: 2
  10.255.0.2
    Received routes
      MAC address advertisement:      2
      MAC+IP address advertisement:   3
      Inclusive multicast:            1
      Ethernet auto-discovery:        1
  10.255.0.3
    Received routes
      MAC address advertisement:      2
      MAC+IP address advertisement:   2
      Inclusive multicast:            1
      Ethernet auto-discovery:        0
Number of ethernet segments: 1
  ESI: 00:22:44:66:88:00:22:44:66:88
  Designated forwarder: 10.255.0.1
  Backup forwarder: 10.255.0.2

Instance: __default_evpn__
Route Distinguisher: 10.255.0.1:0
Encapsulation type: VXLAN
VLAN ID: 0
Per-instance MAC route label: 300208
Per-instance multicast route label: 300224
MAC database status          Local  Remote
  Total MAC addresses:      0      0
  Default gateway MAC addresses: 0      0
Number of local interfaces: 0 (0 up)
Number of IRB interfaces: 0 (0 up)
Number of neighbors: 1
  10.255.0.2
    Received routes
      Ethernet auto-discovery:      0
      Ethernet Segment:             2
Number of ethernet segments: 0

```

show evpn instance extensive (In Junos OS Release 16.2 and later)

```
user@host> show evpn instance extensive
```

```
user@host> show evpn instance extensive
```

```

Instance: ALPHA
Route Distinguisher: 10.255.0.1:100
Encapsulation type: VXLAN
Per-instance MAC route label: 300144
Per-instance multicast route label: 300160
MAC database status          Local  Remote
  Total MAC addresses:      3      4
  Default gateway MAC addresses: 1      2
Number of local interfaces: 2 (2 up)
  Interface name  ESI                               Mode          SH label
  ae0.0           00:11:22:33:44:55:66:77:88:99    single-active

```

```

ge-0/0/2.0      00:00:00:00:00:00:00:00:00:00 single-homed
Number of IRB interfaces: 1 (1 up)
Interface name  L3 context
irb.0           DELTA
Number of neighbors: 2
Address          MAC      MAC+IP      AD      IM      ES
10.255.0.2       2        3          1       1       0
10.255.0.3       2        2          0       1       0
Number of ethernet segments: 1
ESI: 00:11:22:33:44:55:66:77:88:99
Designated forwarder: 10.255.0.1
Backup forwarder: 10.255.0.2

Instance: BETA
Route Distinguisher: 10.255.0.1:300
Encapsulation type: VXLAN
VLAN ID: 300
Per-instance MAC route label: 300176
Per-instance multicast route label: 300192
MAC database status          Local Remote
Total MAC addresses:         3       4
Default gateway MAC addresses: 1       2
Number of local interfaces: 2 (2 up)
Interface name  ESI                      Mode      SH label
ae1.0           00:00:00:00:00:00:00:00:00:00 single-homed
ge-0/0/4.0       00:22:44:66:88:00:22:44:66:88 single-active
Number of IRB interfaces: 1 (1 up)
Interface name  L3 context
irb.1           DELTA
Number of neighbors: 2
Address          MAC      MAC+IP      AD      IM      ES
10.255.0.2       2        3          1       1       0
10.255.0.3       2        2          0       1       0
Number of ethernet segments: 1
ESI: 00:22:44:66:88:00:22:44:66:88
Designated forwarder: 10.255.0.1
Backup forwarder: 10.255.0.2

Instance: __default_evpn__
Route Distinguisher: 10.255.0.1:0
Encapsulation type: VXLAN
VLAN ID: 0
Per-instance MAC route label: 300208
Per-instance multicast route label: 300224
MAC database status          Local Remote
Total MAC addresses:         0       0
Default gateway MAC addresses: 0       0
Number of local interfaces: 0 (0 up)
Number of IRB interfaces: 0 (0 up)
Number of neighbors: 1
Address          MAC      MAC+IP      AD      IM      ES
10.255.0.2       0        0          0       0       2
Number of ethernet segments: 0

```

show evpn instance extensive (Preference-based DF Election)

```

user@host> show evpn instance EVPN_1 extensive
Instance: EVPN_1
Route Distinguisher: 1:101
Per-instance MAC route label: 299792
+ DF Election preference: Lowest preference

```

```

MAC database status
MAC advertisements:
MAC+IP advertisements:
Default gateway MAC advertisements:
Number of local interfaces: 2 (2 up)
Interface name  ESI
ae101.0         00:11:11:11:11:11:11:11:11:01
ae102.0         00:11:11:11:11:11:11:11:02
Number of IRB interfaces: 4 (4 up)
Interface name  VLAN  VNI  Status  L3 context
irb.101         101      Up    master
irb.102         102      Up    master
irb.103         103      Up    master
irb.104         104      Up    master
Number of bridge domains: 4
VLAN  Domain ID  Intfs / up  IRB intf  Mode  MAC sync  IM route
Label
101      1  1  irb.101  Extended  Enabled  299776
102      1  1  irb.102  Extended  Enabled  299776
103      1  1  irb.103  Extended  Enabled  299776
104      1  1  irb.104  Extended  Enabled  299776
Number of neighbors: 2
100.0.0.2
Received routes
MAC address advertisement:
MAC+IP address advertisement:
Inclusive multicast:
Ethernet auto-discovery:
100.0.0.4
Received routes
MAC address advertisement:
MAC+IP address advertisement:
Inclusive multicast:
Ethernet auto-discovery:
Number of ethernet segments: 2
ESI: 00:11:11:11:11:11:11:11:11
Status: Resolved by IFL ae0.110
Local interface: ae0.110, Status: Up/Forwarding
Number of remote PEs connected: 2
Remote PE      MAC label  Aliasing label  Mode
100.100.100.3  0          0                single-active
100.100.100.2  0          0                single-active
DF Election Algorithm: Preference based
Designated forwarder: 100.100.100.3, Preference: 200
Backup forwarder: 100.100.100.1, Preference: 800
Backup forwarder: 100.100.100.2, Preference: 400
ESI: 00:11:11:11:11:11:11:11:02
Status: Resolved by IFL ae102.0
Local interface: ae102.0, Status: Up/Forwarding
Number of remote PEs connected: 1
Remote PE      MAC label  Aliasing label  Mode
100.0.0.2      299792     299792          all-active
DF Election Algorithm: MOD based
Designated forwarder: 100.0.0.2
Backup forwarder: 100.0.0.1
Last designated forwarder update: Feb 21 22:23:32
Advertised split horizon label: 300800

```

show evpn instance extensive (Duplicate MAC Address)

```

user@host> show evpn instance ALPHA extensive
Instance: ALPHA
  Route Distinguisher: 10.255.0.2:100
  Per-instance MAC route label: 304192
  Duplicate MAC detection threshold: 5
  Duplicate MAC detection window: 180
  Duplicate MAC auto-recovery time: 10
...

```

show evpn instance extensive (Protected Interface)

```

user@host> show evpn instance blue extensive
Instance: blue
  Route Distinguisher: 10.255.255.1:100
  Per-instance MAC route label: 299776
  MAC database status
    MAC advertisements:          Local Remote
    MAC+IP advertisements:      0      0
    Default gateway MAC advertisements: 0      0
  Number of local interfaces: 5 (5 up)
    Interface name  ESI                               Mode          Status
  AC-Role
    ae0.0           00:11:22:33:44:55:66:77:88:99  all-active    Up
  Root
    ge-0/0/3.0      00:00:00:00:00:00:00:00:00  single-homed  Up
  Root
    ge-0/0/4.0      00:11:11:11:44:55:66:77:88:99  all-active    Up
  Root
    ge-0/0/4.1      00:22:22:22:44:55:66:77:88:99  all-active    Up
  Root
    ge-0/0/4.50     00:00:00:00:00:00:00:00:00  single-homed  Up
  Root
  Number of IRB interfaces: 1 (0 up)
    Interface name  VLAN  VNI  Status  L3 context
    irb.1           25    Down  vrf
  Number of protect interfaces: 1
    Interface name  protect-interface  active-interface
    ge-0/0/3.1      ge-0/0/4.50       ge-0/0/3.1

```

show evpn instance esi backup-forwarder (Instance Name with Ethernet Segment Identifier)

```

user@host> show evpn instance ALPHA esi 00:11:22:33:44:55:66:77:88:99 backup-forwarder
Instance: ALPHA
  Number of ethernet segments: 1
  ESI: 00:11:22:33:44:55:66:77:88:99
  Backup forwarder: 10.255.0.2

```

show evpn instance esi designated-forwarder (Instance Name with Ethernet Segment Identifier)

```

user@host> show evpn instance ALPHA esi 00:11:22:33:44:55:66:77:88:99 designated-forwarder
Instance: ALPHA
  Number of ethernet segments: 1
  ESI: 00:11:22:33:44:55:66:77:88:99
  Designated forwarder: 10.255.0.1

```


show evpn ip-prefix-database

Syntax	<pre>show evpn ip-prefix-database <extensive> <direction (imported exported)> <esi <i>number</i>> <ethernet-tag <i>number</i>> <family (inet inet6)> <gateway <i>ip-address</i>> <l3-context <i>routing-instance-name</i>> <logical-system <i>logical-system-name</i>> <nexthop <i>ip-address</i>> <prefix <i>ip-prefix</i>></pre>
Release Information	<p>Command introduced in Junos OS Release 15.1X53-D30 for QFX10002 switches.</p> <p>Statement introduced in Junos OS Release 15.1X53-D60 for QFX10008 and QFX10016 switches.</p> <p>Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.</p>
Description	Display Ethernet VPN (EVPN) database information for imported and exported IP prefixes.
Options	<p>none—Display standard information for all EVPN imported and exported IP prefixes.</p> <p>extensive—Display the specified level of output.</p> <p>direction (imported exported)—(Optional) Display the specified subset of IP prefixes.</p> <p>esi <i>number</i>—(Optional) Display the IP prefix associated with the specified Ethernet segment identifier.</p> <p>ethernet-tag <i>number</i>—(Optional) Display the IP prefix associated with the specified Ethernet tag.</p> <p>family (inet inet6)—(Optional) Display IP prefixes for the specified protocol family.</p> <p>gateway <i>ip-address</i>—(Optional) Display the IP Prefix associated with the overlay gateway IP address.</p> <p>l3-context <i>routing-instance-name</i>—(Optional) Display EVPN IP prefix information for the specified Layer 3 virtual routing and forwarding (VRF) instance.</p> <p>logical-system <i>logical-system-name</i>—(Optional) Display EVPN IP prefix information for the specified logical system or all logical systems.</p> <p>nexthop <i>ip-address</i>—(Optional) Display EVPN IP prefix information for the specified underlay next-hop IP address.</p> <p>prefix <i>ip-prefix</i>—(Optional) Display EVPN database information for the specified IP prefix.</p>
Required Privilege Level	view

- Related Documentation**
- [show evpn l3-context on page 336](#)
 - [show route table](#)

- List of Sample Output**
- [show evpn ip-prefix-database on page 331](#)
 - [show evpn ip-prefix-database extensive on page 332](#)

- Output Fields**
- [Table 18 on page 330](#) lists the output fields for the **show evpn ip-prefix-database** command. Output fields are listed in the approximate order in which they appear.

Table 18: show evpn ip-prefix-database Output Fields

Field Name	Field Description	Level of Output
L3 context	Name of virtual routing and forwarding (VRF) instance.	All levels
EVPN Exported Prefixes Prefix	List of exported IP prefixes.	All levels
EVPN route status	For exported prefixes only, status of route: Created .	level-of-output none
EVPN imported Prefixes	List of imported EVPN IP prefixes.	All levels
Etag	Ethernet tag	level-of-output none
Route distinguisher	IP address identifier for the IP prefix route.	All levels
VNI	Virtual network identifier for the Layer 3 virtual and routing forwarding (VRF) for the customer or tenant domain.	All levels
Router MAC	MAC address associated with the IP prefix.	All levels
Nexthop/Overlay GW/ESI	For imported IP prefixes, next-hop IP address.	level-of-output none
Change flags	Trace flags.	level-of-output extensive
Advertisement mode	For exported IP prefixes, type of next-hop address.	level-of-output extensive
Encapsulation	For exported IP prefixes, type of encapsulation	level-of-output extensive
Remote Advertisements	For imported IP prefixes, route distinguisher identifier, MAC address, virtual network identifier, and BGP next-hop address to remote destination.	level-of-output extensive

Sample Output

show evpn ip-prefix-database

```
user@host > show evpn ip-prefix-database
L3 context: VRF-100
```

IPv4->EVPN Exported Prefixes

Prefix	EVPN route status
100.1.0.0/22	Created
100.1.4.0/22	Created
100.1.8.0/22	Created
100.1.12.0/22	Created
100.1.16.0/22	Created

IPv6->EVPN Exported Prefixes

Prefix	EVPN route status
1234:100:1::/64	Created
1234:100:1:4::/64	Created
1234:100:1:8::/64	Created
1234:100:1:12::/64	Created
1234:100:1:16::/64	Created

EVPN->IPv4 Imported Prefixes

Prefix	Etag	IP route status
100.2.0.0/22	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0
100.2.4.0/22	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0
100.2.8.0/22	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0
100.2.12.0/22	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0
100.2.16.0/22	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0

EVPN->IPv6 Imported Prefixes

Prefix	Etag	IP route status
1234:100:2::/64	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0
1234:100:2:4::/64	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0
1234:100:2:8::/64	0	Created
Route distinguisher	VNI/Label	Router MAC
10.255.2.1:100	9101	00:05:86:e1:03:f0
10.255.2.2:100	9101	00:05:86:d0:a6:f0

1234:100:2:12::/64		0	Created
Route distinguisher	VNI/Label	Router MAC	Nexthop/Overlay GW/ESI
10.255.2.1:100	9101	00:05:86:e1:03:f0	10.255.2.1
10.255.2.2:100	9101	00:05:86:d0:a6:f0	10.255.2.2
1234:100:2:16::/64		0	Created
Route distinguisher	VNI/Label	Router MAC	Nexthop/Overlay GW/ESI
10.255.2.1:100	9101	00:05:86:e1:03:f0	10.255.2.1
10.255.2.2:100	9101	00:05:86:d0:a6:f0	10.255.2.2

show evpn ip-prefix-database extensive

```
user@host> show evpn ip-prefix-database extensive
L3 context: VRF-100
```

IPv4->EVPN Exported Prefixes

```
Prefix: 100.1.0.0/22
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0
```

```
Prefix: 100.1.4.0/22
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0
```

```
Prefix: 100.1.8.0/22
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0
```

```
Prefix: 100.1.12.0/22
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0
```

```
Prefix: 100.1.16.0/22
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0
```

IPv6->EVPN Exported Prefixes

```
Prefix: 1234:100:1::/64
EVPN route status: Created
Change flags: 0x0
```

```

Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0

Prefix: 1234:100:1:4::/64
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0

Prefix: 1234:100:1:8::/64
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0

Prefix: 1234:100:1:12::/64
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0

Prefix: 1234:100:1:16::/64
EVPN route status: Created
Change flags: 0x0
Advertisement mode: Direct nexthop
Encapsulation: VXLAN
VNI: 9100
Router MAC: 00:05:86:28:90:f0

EVPN->IPv4 Imported Prefixes

Prefix: 100.2.0.0/22, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
  Route Distinguisher: 10.255.2.1:100
  VNI: 9101
  Router MAC: 00:05:86:e1:03:f0
  BGP nexthop address: 10.255.2.1
  Route Distinguisher: 10.255.2.2:100
  VNI: 9101
  Router MAC: 00:05:86:d0:a6:f0
  BGP nexthop address: 10.255.2.2

Prefix: 100.2.4.0/22, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
  Route Distinguisher: 10.255.2.1:100
  VNI: 9101
  Router MAC: 00:05:86:e1:03:f0
  BGP nexthop address: 10.255.2.1
  Route Distinguisher: 10.255.2.2:100

```

VNI: 9101
Router MAC: 00:05:86:d0:a6:f0
BGP nexthop address: 10.255.2.2

Prefix: 100.2.8.0/22, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
Route Distinguisher: 10.255.2.1:100
VNI: 9101
Router MAC: 00:05:86:e1:03:f0
BGP nexthop address: 10.255.2.1
Route Distinguisher: 10.255.2.2:100
VNI: 9101
Router MAC: 00:05:86:d0:a6:f0
BGP nexthop address: 10.255.2.2

Prefix: 100.2.12.0/22, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
Route Distinguisher: 10.255.2.1:100
VNI: 9101
Router MAC: 00:05:86:e1:03:f0
BGP nexthop address: 10.255.2.1
Route Distinguisher: 10.255.2.2:100
VNI: 9101
Router MAC: 00:05:86:d0:a6:f0
BGP nexthop address: 10.255.2.2

Prefix: 100.2.16.0/22, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
Route Distinguisher: 10.255.2.1:100
VNI: 9101
Router MAC: 00:05:86:e1:03:f0
BGP nexthop address: 10.255.2.1
Route Distinguisher: 10.255.2.2:100
VNI: 9101
Router MAC: 00:05:86:d0:a6:f0
BGP nexthop address: 10.255.2.2

EVPN->IPv6 Imported Prefixes

Prefix: 1234:100:2::/64, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
Route Distinguisher: 10.255.2.1:100
VNI: 9101
Router MAC: 00:05:86:e1:03:f0
BGP nexthop address: 10.255.2.1
Route Distinguisher: 10.255.2.2:100
VNI: 9101
Router MAC: 00:05:86:d0:a6:f0
BGP nexthop address: 10.255.2.2

Prefix: 1234:100:2:4::/64, Ethernet tag: 0
IP route status: Created
Change flags: 0x0

```
Remote advertisements:
  Route Distinguisher: 10.255.2.1:100
    VNI: 9101
    Router MAC: 00:05:86:e1:03:f0
    BGP nexthop address: 10.255.2.1
  Route Distinguisher: 10.255.2.2:100
    VNI: 9101
    Router MAC: 00:05:86:d0:a6:f0
    BGP nexthop address: 10.255.2.2

Prefix: 1234:100:2:8::/64, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
  Route Distinguisher: 10.255.2.1:100
    VNI: 9101
    Router MAC: 00:05:86:e1:03:f0
    BGP nexthop address: 10.255.2.1
  Route Distinguisher: 10.255.2.2:100
    VNI: 9101
    Router MAC: 00:05:86:d0:a6:f0
    BGP nexthop address: 10.255.2.2

Prefix: 1234:100:2:12::/64, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
  Route Distinguisher: 10.255.2.1:100
    VNI: 9101
    Router MAC: 00:05:86:e1:03:f0
    BGP nexthop address: 10.255.2.1
  Route Distinguisher: 10.255.2.2:100
    VNI: 9101
    Router MAC: 00:05:86:d0:a6:f0
    BGP nexthop address: 10.255.2.2

Prefix: 1234:100:2:16::/64, Ethernet tag: 0
IP route status: Created
Change flags: 0x0
Remote advertisements:
  Route Distinguisher: 10.255.2.1:100
    VNI: 9101
    Router MAC: 00:05:86:e1:03:f0
    BGP nexthop address: 10.255.2.1
  Route Distinguisher: 10.255.2.2:100
    VNI: 9101
    Router MAC: 00:05:86:d0:a6:f0
    BGP nexthop address: 10.255.2.2
```

show evpn l3-context

Syntax `show evpn l3-context`
`<brief | extensive>`
`<logical-system logical-system-name>`
`<routing-instance-name>`

Release Information Statement introduced in Junos OS Release 15.1X53-D30 for QFX10002 switches.
Statement introduced in Junos OS Release 15.1X53-D60 for QFX10008 and QFX10016 switches in Junos OS Release 15.1X53-D60.
Statement introduced in Junos OS Release 17.3R1 for EX9200 switches.

Description Display EVPN database information for Layer 3 virtual routing and forwarding (VRF) instances.

Options **none**—Display standard EVPN database information for all Layer 3 VRF instances.

brief | extensive—(Optional) Display the specified level of output.

logical-system <logical-system-name>—(Optional) Display Layer 3 information for the specified logical system or all logical systems.

routing-instance-name—(Optional) Display EVPN database information for the specified Layer 3 VRF instance.

Additional Information

Required Privilege Level view

Related Documentation • [show evpn ip-prefix-database on page 329](#)

List of Sample Output [show evpn l3-context on page 337](#)
[show evpn l3-context extensive on page 337](#)

Output Fields [Table 19 on page 336](#) lists the output fields for the **show evpn l3-context** command. Output fields are listed in the approximate order in which they appear.

Table 19: show evpn l3-context Output Fields

Field Name	Field Description	Level of Output
L3 context	Name of VRF instance.	All levels
Type	Status of VRF instance	All levels
Fwd	Type of forwarding route	level-of-output none

Table 19: show evpn l3-context Output Fields (*continued*)

Field Name	Field Description	Level of Output
Encap	Type of encapsulation.	level-of-output none
VNI	Virtual network identifier for VRF instance.	All levels
Router MAC/GW intf	For a pure type-5 route, the router MAC provides next-hop reachability information and the MAC address of the sending device For a standard type-5 route, the GW interface provides the next-hop route.	All levels
Advertisement mode	Type of forwarding route.	level-of-output Extensive
IPv4 source VTEP address	IPv4 source address for the Virtual Extensible LAN (VXLAN) tunnel.	level-of-output Extensive
IPv6 source VTEP address	IPv6 source address for the Virtual Extensible LAN (VXLAN) tunnel.	level-of-output Extensive
IP->EVPN export policy	Name of export routing policy applied to forwarded IP routes.	level-of-output extensive
Flags	Enabled trace flags.	level-of-output extensive
Change flags	Changed trace flags.	level-of-output extensive
Composite nexthop	Status of next-hop route: Enabled or Disabled .	level-of-output extensive
Route distinguisher	Route distinguisher identifier of the VRF instance	level-of-output extensive
Reference count		level-of-output extensive

Sample Output

show evpn l3-context

```

user@DC1_SPINE1_RE> show evpn l3-context
L3 context          Type  Fwd  Encap  VNI/Label  Router MAC/GW intf
VRF-100             Cfg   Sym  VXLAN  9100       00:05:86:28:90:f0
VRF-200             Cfg   Sym  VXLAN  9200       00:05:86:28:90:f0

```

show evpn l3-context extensive

```

user@DC1_SPINE1_RE> show evpn l3-context extensive

```

L3 context: VRF-100

Type: Configured

Advertisement mode: Direct nexthop, Router MAC: 00:05:86:28:90:f0

Encapsulation: VXLAN, VNI: 9100

IPv4 source VTEP address: 10.255.1.1

IPv6 source VTEP address: abcd::128:102:242:145

IP->EVPN export policy: EVPN-TYPE5-EXPORT-VRF-100

Flags: 0x19 <Configured IRB-MAC New>

Change flags: 0xe <Export-Policy Fwd-Mode Encap>

Composite nexthop support: Enabled

Route Distinguisher: 10.255.1.1:100

Reference count: 21

L3 context: VRF-200

Type: Configured

Advertisement mode: Direct nexthop, Router MAC: 00:05:86:28:90:f0

Encapsulation: VXLAN, VNI: 9200

IPv4 source VTEP address: 10.255.1.1

IPv6 source VTEP address: abcd::128:102:242:145

IP->EVPN export policy: EVPN-TYPE5-EXPORT-VRF-200

Flags: 0x19 <Configured IRB-MAC New>

Change flags: 0xe <Export-Policy Fwd-Mode Encap>

Composite nexthop support: Enabled

Route Distinguisher: 10.255.1.1:200

Reference count: 21

show evpn mac-table

Syntax	<pre>show evpn mac-table <age> <address> <brief count detail extensive summary> <instance <i>instance-name</i>> <interface <i>interface-name</i>> <isid <isid>> <logical-system <logical-system-name>> <vlan-id <i>vlan-id</i>></pre>
Release Information	Command introduced in Junos OS Release 14.2 for EX Series switches.
Description	Show Ethernet VPN (EVPN) MAC table information.
Options	<p>none—Display brief information about the EVPN MAC table.</p> <p>age— (Optional) Display age of a single mac-address.</p> <p>address—(Optional) Display MAC table information for the specified MAC address.</p> <p>brief count detail extensive summary—(Optional) Display the specified level of output.</p> <p>instance <i>instance-name</i>—(Optional) Display MAC table information for a specific routing instance.</p> <p>isid <isid>—(Optional) Display MAC table information for the specified ISID or all ISIDs.</p> <p>logical-system <logical-system-name>—(Optional) Display MAC table information for the specified logical system or all logical systems.</p> <p>vlan-id <i>vlan-id</i>—(Optional) Display MAC table information for the specified VLAN.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 106

show evpn nd-table

Syntax	<pre>show evpn nd-table <address> <brief count detail extensive> <instance instance-name></pre>
Release Information	<p>Command introduced in Junos OS Release 16.2.</p> <p>Command introduced in Junos OS Release 17.1 for MX series Routers.</p> <p>Command introduced in Junos OS Release 17.3R1 for EX series switches.</p>
Description	Show Ethernet VPN (EVPN) Network Discovery Protocol (NDP) entries associated with learned MAC addresses.
Options	<p>none—Display brief information about the EVPN NDP table.</p> <p>address—(Optional) Display NDP information for the specified MAC address.</p> <p>brief count detail extensive—(Optional) Display the specified level of output.</p> <p>instance <instance-name>—(Optional) Display NDP information for the specified routing instance .</p>
Required Privilege Level	view
List of Sample Output	<p>show evpn nd-table on page 341</p> <p>show evpn nd-table 00:05:86:a0:dc:f0 (MAC address) on page 341</p> <p>show evpn nd-table brief on page 341</p> <p>show evpn nd-table detail on page 341</p> <p>show evpn nd-table count on page 341</p> <p>show evpn nd-table extensive on page 342</p> <p>show evpn nd-table instance evpn1 on page 342</p>
Output Fields	<p>Table 20 on page 340 lists the output fields for the show evpn nd-table command. Output fields are listed in the approximate order in which they appear.</p>

Table 20: show evpn nd-table Output Fields

Field Name	Field Description	Level of Output
INET address	The INET address related to the INET entries that are added to the NDP table.	All levels
MAC address	MAC addresses learned through NDP.	brief, detail, extensive, instance, mac-address,,

Table 20: show evpn nd-table Output Fields (*continued*)

Field Name	Field Description	Level of Output
Logical Interface	Logical interface associated with the routing instance in which the NDP INET address is learned.	brief, instance, mac-address,,
Routing instance	Routing instance in which the NDP INET address is learned.	all levels
Bridging domain	Bridging domain in which the NDP INET address is learned.	all levels
Learning interface	Interface on which the NDP INET address is learned.	detail, extensive
Count	Indicates the number of NDP INET addresses learned in a routing instance in a bridge domain.	count

Sample Output

show evpn nd-table

```

user@host> show evpn nd-table
INET          MAC          Logical    Routing    Bridging
address       address      interface  instance   domain
8001::2       00:05:86:a0:dc:f0  irb.0      evpn_1     __evpn_1__

```

show evpn nd-table 00:05:86:a0:dc:f0 (MAC address)

```

user@host> show evpn nd-table 00:05:86:a0:dc:f0
INET          MAC          Logical    Routing    Bridging
address       address      interface  instance   domain
8001::2       00:05:86:a0:dc:f0  irb.0      evpn1      __evpn1__

```

show evpn nd-table brief

```

user@host> show evpn nd-table brief
INET          MAC          Logical    Routing    Bridging
address       address      interface  instance   domain
8001::2       00:05:86:a0:dc:f0  irb.0      evpn1      __evpn1__

```

show evpn nd-table detail

```

user@host> show evpn nd-table detail

INET address: 8001::2
MAC address: 00:05:86:a0:dc:f0
Routing instance: evpn1
Bridging domain: __evpn1__
Learning interface: irb.0

```

show evpn nd-table count

```

user@switch> show evpn nd-table count
1 NDP INET addresses learned in routing instance evpn1 bridge domain __evpn1__

```

show evpn and-table extensive

```
user@host> show evpn nd-table extensive
```

```
INET address: 8001::2
MAC address: 00:05:86:a0:dc:f0
Routing instance: evpn1
Bridging domain: __evpn1__
Learning interface: irb.0
```

show evpn ndp-table instance evpn1

```
user@host> show evpn arp-table instance evpn1
```

INET address	MAC address	Logical interface	Routing instance	Bridging domain
8001::2	00:05:86:a0:dc:f0	irb.0	evpn1	__evpn1__

show evpn peer-gateway-macs

Syntax	show evpn peer-gateway-macs <address> <instance <i>instance-name</i> >
Release Information	Command introduced in Junos OS Release 14.2 for EX Series switches.
Description	Show Ethernet VPN (EVPN) peer gateway MAC information.
Options	none —Display brief information about the EVPN peer gateway MAC. address —(Optional) Display peer gateway information for the specified MAC address. instance <i>instance-name</i> —(Optional) Display peer gateway MAC information for the specified routing instance.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• Example: Configuring an EVPN with IRB Solution on EX9200 Switches on page 106

show vlans evpn nd-table

Syntax show vlans evpn nd-table
 <brief | detail | extensive>
 <count>
 <instance-name>
 <mac-address>
 <vlan-name>

Release Information Command introduced in Junos OS Release 17.3R1 for EX Series Switches.

Description Display information about INET entries associated with MAC addresses learned through Network Discovery Protocol (NDP).

Options **none**—Display information for all INET entries.

brief | detail | extensive—(Optional) Display the specified level of output.

count—(Optional) Display the number of INET addresses learned in a routing instance.

instance—(Optional) Display information for a specified instance.

mac-address—(Optional) Display information for a specified MAC address.

vlan-name (all)—(Optional) Display information for a specified VLAN or for all VLANs.

Required Privilege Level view

List of Sample Output [show vlans evpn nd-table brief on page 345](#)
[show vlans evpn nd-table detail on page 345](#)
[show vlans evpn nd-table extensive on page 345](#)
[show vlans evpn nd-table count on page 345](#)
[show vlans evpn nd-table instance evpn1 on page 345](#)
[show vlans evpn nd-table instance 00:05:86:90:bd:f0 \(MAC address\) on page 346](#)
[show vlans evpn nd-table vlan-name vlan10 on page 346](#)

Output Fields [Table 21 on page 344](#) lists the output fields for the **show vlans evpn nd-table** command. Output fields are listed in the approximate order in which they appear.

Table 21: show vlans evpn nd-table Output Fields

Field Name	Field Description	Level of Output
INET address	The INET address related to the INET entries that are added to the NDP table.	All levels
MAC address	MAC addresses learned through NDP.	brief, detail, extensive, instance, mac-address, vlan-name,,

Table 21: show vlans evpn nd-table Output Fields (*continued*)

Field Name	Field Description	Level of Output
Logical Interface	Logical interface associated with the routing instance in which the NDP INET address is learned.	brief, instance, mac-addressvlan-name,,
Routing instance	Routing instance in which the NDP INET address is learned.	all levels
Bridging domain	Bridging domain in which the NDP INET address is learned.	all levels
Learning interface	Interface on which the NDP INET address is learned.	detail, extensive
Count	Indicates the number of NDP INET addresses learned in a routing instance in a bridge domain.	count

Sample Output

show vlans evpn nd-table brief

```

user@switch> show vlans evpn nd-table brief
INET          MAC          Logical      Routing      Bridging
address       address      interface    instance     domain
8002::2       00:05:86:a0:d5:00  irb.0        evpn1        vlan10

```

show vlans evpn nd-table detail

```

user@switch> show vlans evpn nd-table detail
INET address: 8002::2
MAC address: 00:05:86:a0:d5:00
Routing instance: evpn1
Bridging domain: vlan10
Learning interface: irb.0

```

show vlans evpn nd-table extensive

```

user@switch> show vlans evpn nd-table extensive
INET address: 8002::2
MAC address: 00:05:86:a0:d5:00
Routing instance: evpn1
Bridging domain: vlan10
Learning interface: irb.0

```

show vlans evpn nd-table count

```

user@switch> show vlans evpn nd-table count
1 ND INET addresses learned in routing instance evpn1 bridge domain vlan10

```

show vlans evpn nd-table instance evpn1

```

user@switch> show vlans evpn nd-table instance evpn1
INET          MAC          Logical      Routing      Bridging
address       address      interface    instance     domain
8002::2       00:05:86:a0:d5:00
                evpn1        vlan10
irb.0

```

show vlans evpn nd-table instance 00:05:86:90:bd:f0 (MAC address)

```
user@switch> show vlans evpn nd-table
INET          MAC          Logical   Routing   Bridging
address       address      interface instance  domain
8002::2
               00:05:86:a0:d5:00
  irb.0        evpn1      __evpn1__
```

show vlans evpn nd-table vlan-name vlan10

```
user@switch> show vlans evpn nd-table vlan-name vlan10
INET          MAC          Logical   Routing   Bridging
address       address      interface instance  domain
8002::2        00:05:86:a0:d5:00
  irb.0        evpn1      vlan10
```