



Junos[®] OS

OVSDB and VXLAN Feature Guide for QFX Series Switches (VMware NSX)

Release
15.1



Modified: 2016-12-12

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

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the 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.

Junos[®] OS OVSDDB and VXLAN Feature Guide for QFX Series Switches (VMware NSX)

15.1

Copyright © 2016, 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.html>. 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	Overview	
Chapter 1	OVSDB and VXLAN Overview	3
	Understanding the Junos OS Implementation of OVSDB and VXLAN in a VMware NSX for Multi-Hypervisor Environment	3
	Understanding VXLANs	6
	VXLAN Benefits	6
	How Does VXLAN Work?	7
	VXLAN Implementation Methods	8
	Using QFX5100 and QFX5110 Switches with VXLANs	8
	Changing the UDP Port on QFX5100 and QFX5110 Switches	9
	Controlling Transit Multicast Traffic on QFX5100 and QFX5110 Switches	9
	Using an MX Series Router, EX9200 Switch, or QFX10000 Switch as a VTEP	10
	Manual VXLANs Require PIM	10
	Load Balancing VXLAN Traffic	11
	Understanding the OVSDB Protocol Running on Juniper Networks Devices	12
	OVSDB Support on Juniper Networks Devices	12
	OVSDB Schema for Physical Devices	14
	Understanding How Layer 2 BUM and Layer 3 Routed Multicast Traffic Are Handled with OVSDB	16

Part 2	Configuring OVSDB and VXLAN	
Chapter 2	Configuring OVSDB-Managed VXLANs with an SDN Controller	21
	OVSDB and VXLAN Configuration Workflows for VMware NSX Environment	21
	OVSDB and VXLAN Configuration Workflow for QFX Series Switches	21
	OVSDB and VXLAN Configuration Workflow for MX Series Routers and EX9200 Switches	23
	Understanding How to Set Up OVSDB Connections on a Juniper Networks Device	24
	Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers	25
	Setting Up OVSDB on Juniper Networks Devices That Support the Dynamic Configuration of VXLANs	27
	Understanding Dynamically Configured VXLANs in an OVSDB Environment	28
	Performing Tasks Before and After the Dynamic Configuration of OVSDB-Managed VXLANs	28
	What the Juniper Networks Switch Actually Creates Dynamically	33
	Dynamic Association of a Trunk Interface Supporting Untagged Packets to a Dynamically Created VXLAN	33
	Dynamic Association of a Trunk Interface Supporting Tagged Packets to a Dynamically Created VXLAN	34
	VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints	35
	Creating a Gateway	36
	Creating a Gateway Service	36
	Creating a Logical Switch Port	37
	Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections in a VMware NSX Environment (Trunk Interfaces Supporting Untagged Packets)	38
	Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections in a VMware NSX Environment (Trunk Interfaces Supporting Tagged Packets)	46
	Verifying That a Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN Are Working Properly	54
Chapter 3	Configuring VXLANs Without an SDN Controller	57
	VXLAN Constraints on QFX Series Switches	57
	VXLAN Constraints on QFX5100 and QFX5110 Switches	57
	VXLAN Constraints on QFX10000 Switches	59
	Manually Configuring VXLANs on QFX Series Switches	60
	Configuring a Source IP Address	60
	Configuring PIM for VXLANs	60
	Configuring VXLANs	61
	Examples: Manually Configuring VXLANs on QFX Series Switches	61
	Example: Configuring a VXLAN Transit Switch	61
	Example: Configuring a VXLAN Layer 2 Gateway	63
	Verifying That a Local VXLAN VTEP Is Configured Correctly	69
	Verifying MAC Learning from a Remote VTEP	69

Part 3	Troubleshooting	
Chapter 4	Troubleshooting Tasks	73
	Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS	
	OVSDB-Managed VXLAN	73
	Verifying VXLAN Reachability	75
	Monitoring a Remote VTEP Interface	75
Part 4	Configuration Statements and Operational Commands	
Chapter 5	OVSDB Configuration Statements	79
	controller (OVSDB)	80
	inactivity-probe-duration	81
	interfaces (OVSDB)	82
	maximum-backoff-duration	83
	ovsdb	84
	ovsdb-managed	85
	port (OVSDB)	86
	protocol (OVSDB)	87
	traceoptions (OVSDB)	88
Chapter 6	VXLAN Configuration Statements	91
	decapsulate-accept-inner-vlan	91
	encapsulate-inner-vlan	92
	multicast-group	92
	ovsdb-managed	93
	unreachable-vtep-aging-timer	94
	vni	94
	vtep-source-interface	95
	vxlan	95
Chapter 7	OVSDB Operational Commands	97
	clear ovsdb commit failures	98
	show ovsdb commit failures	100
	show ovsdb controller	102
	show ovsdb interface	104
	show ovsdb logical-switch	106
	show ovsdb mac	109
	show ovsdb statistics interface	113
	show ovsdb virtual-tunnel-end-point	115
	show vpls mac-table	117
Chapter 8	VXLAN Operational Commands	123
	show bridge mac-table	124
	show vpls mac-table	129

List of Figures

Part 1	Overview	
Chapter 1	OVSDB and VXLAN Overview	3
	Figure 1: High-Level View of NSX for Multi-Hypervisor Architecture	4
	Figure 2: Integration of Juniper Networks Device into NSX for Multi-Hypervisor Environment	5
	Figure 3: VXLAN Packet Format	8
Part 2	Configuring OVSDB and VXLAN	
Chapter 2	Configuring OVSDB-Managed VXLANs with an SDN Controller	21
	Figure 4: VXLAN-OVSDB Layer 2 Gateway Topology	40
	Figure 5: VXLAN/OVSDB Layer 2 Gateway Topology	47
Chapter 3	Configuring VXLANs Without an SDN Controller	57
	Figure 6: QFX5100 Acting as a VXLAN Transit Switch	62
	Figure 7: QFX5100 Acting as a VTEP	64

List of Tables

	About the Documentation	xi
	Table 1: Notice Icons	xiii
	Table 2: Text and Syntax Conventions	xiii
Part 1	Overview	
Chapter 1	OVSDB and VXLAN Overview	3
	Table 3: NSX for Multi-Hypervisor Components and Related Products	3
	Table 4: OVSDB Support on Juniper Networks Devices	13
	Table 5: OVSDB Schema Tables	14
Part 2	Configuring OVSDB and VXLAN	
Chapter 2	Configuring OVSDB-Managed VXLANs with an SDN Controller	21
	Table 6: OVSDB and VXLAN Configuration Workflow for QFX Series Switches	22
	Table 7: OVSDB and VXLAN Configuration Workflow for MX Series Routers and EX9200 Switches	23
	Table 8: Workflow of Tasks and Events for the Dynamic Configuration of OVSDB-Managed VXLANs in an NSX Environment	29
	Table 9: Workflow of Tasks and Events for the Dynamic Configuration of OVSDB-Managed VXLANs in a Contrail Environment	31
	Table 10: Key Configurations to Create a Gateway in NSX Manager	36
	Table 11: Key Configurations to Create a Gateway Service in NSX Manager	37
	Table 12: Key Configurations to Create a Logical Switch Port in NSX Manager	37
	Table 13: NSX Manager and Junos OS Entities That Must Be Configured	40
	Table 14: Components of the Topology for Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections	41
	Table 15: NSX Manager and Junos OS Entities That Must Be Configured	48
	Table 16: NSX Manager Configurations and Dynamic Configurations by Juniper Networks Switch	49
	Table 17: Components for Two VXLAN Topologies Configured on a Juniper Networks Switch that Functions as a Hardware VTEP	49
Part 4	Configuration Statements and Operational Commands	
Chapter 7	OVSDB Operational Commands	97
	Table 18: show ovssdb commit failures Output Fields	101
	Table 19: show ovssdb controller Output Fields	102
	Table 20: show ovssdb interface Output Fields	104
	Table 21: show ovssdb logical-switch Output Fields	107

	Table 22: show ovssdb mac Output Fields	110
	Table 23: show ovssdb statistics interface Output Fields	113
	Table 24: show ovssdb virtual-tunnel-end-point Output Fields	115
	Table 25: show vpls mac-table Output fields	118
Chapter 8	VXLAN Operational Commands	123
	Table 26: show bridge mac-table Output Fields	125
	Table 27: show vpls mac-table Output fields	130

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:

- [QFX 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 xiii 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

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

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	<pre>user@host> show chassis alarms</pre> <p>No alarms currently active</p>
<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.	<p>Configure the machine's domain name:</p> <pre>[edit] root@# set system domain-name domain-name</pre>
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 metric>;
(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 (string1 string2 string3)
# (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.	<pre>[edit] routing-options { static { route default { nexthop address; retain; } } }</pre>
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
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.

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

Convention	Description	Examples
> (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: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- 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://tools.juniper.net/SerialNumberEntitlementSearch/>

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

Overview

- [OVSDB and VXLAN Overview on page 3](#)

CHAPTER 1

OVSDB and VXLAN Overview

- [Understanding the Junos OS Implementation of OVSDB and VXLAN in a VMware NSX for Multi-Hypervisor Environment on page 3](#)
- [Understanding VXLANs on page 6](#)
- [Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 12](#)
- [OVSDB Support on Juniper Networks Devices on page 12](#)
- [OVSDB Schema for Physical Devices on page 14](#)
- [Understanding How Layer 2 BUM and Layer 3 Routed Multicast Traffic Are Handled with OVSDB on page 16](#)

Understanding the Junos OS Implementation of OVSDB and VXLAN in a VMware NSX for Multi-Hypervisor Environment

Some Juniper Networks devices support Virtual Extensible LAN (VXLAN) and the Open vSwitch Database (OVSDB) management protocol. (See [“OVSDB Support on Juniper Networks Devices” on page 12](#).) Support for VXLAN and OVSDB enables the Juniper Networks devices in a physical network to be integrated into a virtual network.

The implementation of VXLAN and OVSDB on Juniper Networks devices is supported in a VMware NSX for Multi-Hypervisor environment for the data center. [Table 3 on page 3](#) outlines the components that compose this environment and products that are typically deployed for each component.

Table 3: NSX for Multi-Hypervisor Components and Related Products

Component	Products
Cloud management platform (CMP)	CloudStack
	OpenStack
	Custom CMP
Network virtualization platform	NSX for Multi-Hypervisor

Table 3: NSX for Multi-Hypervisor Components and Related Products (*continued*)

Component	Products
Hypervisor	Kernel-based Virtual Machine (KVM) Red Hat VMware ESXi Xen NOTE: Juniper Networks supports only KVM and ESXi.
Virtual switch	Open vSwitch (OVS) NSX vSwitch
SDN controller	NSX for Multi-Hypervisor controller
Overlay protocol	VXLAN
Media access control (MAC) learning protocol	OVSDB

Figure 1 on page 4 shows a high-level view of the NSX for Multi-Hypervisor platform architecture, while Figure 2 on page 5 provides a more detailed representation of the components in the virtual and physical networks.

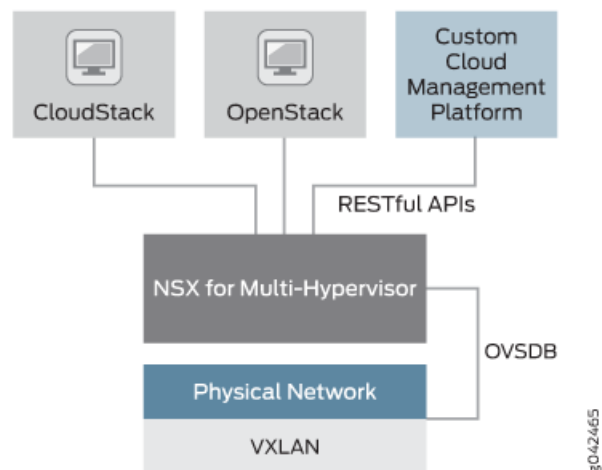
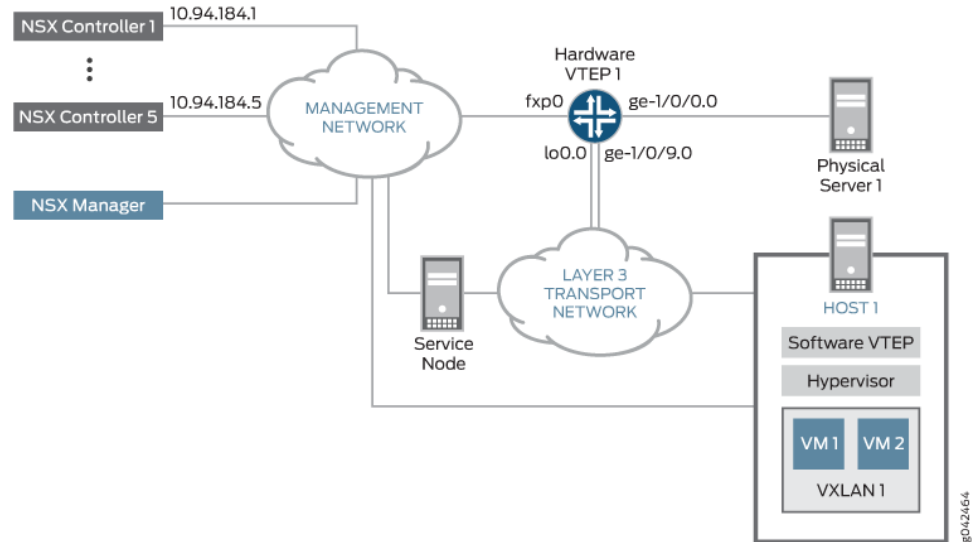
Figure 1: High-Level View of NSX for Multi-Hypervisor Architecture

Figure 2: Integration of Juniper Networks Device into NSX for Multi-Hypervisor Environment



In the data center topology shown in [Figure 2 on page 5](#), the physical and virtual servers need to communicate. To facilitate this communication, a Juniper Networks device that supports VXLAN is strategically deployed so that it serves as a *gateway*, which is also known as a hardware virtual tunnel endpoint (VTEP), at the edge of the physical network. Working in conjunction with the software VTEP, which is deployed at the edge of the virtual network, the hardware VTEP encapsulates packets from resources on Physical Server 1 with a VXLAN header, and after the packets traverse the Layer 3 transport network, the software VTEP removes the VXLAN header from the packets and forwards the packets to the appropriate virtual machines (VMs). In essence, the encapsulation and de-encapsulation of packets by the hardware and software VTEPs enable the components in the physical and virtual networks to coexist without one needing to understand the workings of the other.

The same Juniper Networks device that acts as a hardware VTEP in [Figure 2 on page 5](#) implements OVSDb, which enables this device to learn the MAC addresses of Physical Server 1 and other physical servers, and publish the addresses in the OVSDb schema, which was defined for physical devices. In the virtual network, one or more NSX controllers collect the MAC addresses of Host 1 and other virtual servers, and publish the addresses in the OVSDb schema. Using the OVSDb schema, components in the physical and virtual networks can exchange MAC addresses, as well as statistical information, enabling the components to learn about and reach each other in their respective networks.

Related Documentation

- [Understanding the OVSDb Protocol Running on Juniper Networks Devices on page 12](#)
- [OVSDb Schema for Physical Devices](#)

Understanding VXLANs

Virtual Extensible LAN protocol (VXLAN) technology allows networks to support more VLANs. According to the IEEE 802.1Q standard, traditional VLAN identifiers are 12 bits long—this naming limits networks to 4094 VLANs. The VXLAN protocol overcomes this limitation by using a longer logical network identifier that allows more VLANs and, therefore, more logical network isolation for large networks such as clouds that typically include many virtual machines.

- [VXLAN Benefits on page 6](#)
- [How Does VXLAN Work? on page 7](#)
- [VXLAN Implementation Methods on page 8](#)
- [Using QFX5100 and QFX5110 Switches with VXLANs on page 8](#)
- [Changing the UDP Port on QFX5100 and QFX5110 Switches on page 9](#)
- [Controlling Transit Multicast Traffic on QFX5100 and QFX5110 Switches on page 9](#)
- [Using an MX Series Router, EX9200 Switch, or QFX10000 Switch as a VTEP on page 10](#)
- [Manual VXLANs Require PIM on page 10](#)
- [Load Balancing VXLAN Traffic on page 11](#)

VXLAN Benefits

VXLAN technology allows you to segment your networks (as VLANs do), but it provides benefits that VLANs cannot. Here are the most important benefits of using VXLANs:

- You can theoretically create as many as 16 million VXLANs in an administrative domain (as opposed to 4094 VLANs on a Juniper Networks device).
 - MX Series routers and EX9200 switches support as many as 32,000 VXLANs, 32,000 multicast groups, and 8000 virtual tunnel endpoints (VTEPs). This means that VXLANs based on MX Series routers provide network segmentation at the scale required by cloud builders to support very large numbers of tenants.
 - QFX10000 switches support 4000 VXLANs and 2000 VTEPs.
 - QFX5100 and QFX5110 switches support 4000 VXLANs, 4000 multicast groups, and 2000 VTEPs.
- You can enable migration of virtual machines between servers that exist in separate Layer 2 domains by tunneling the traffic over Layer 3 networks. This functionality allows you to dynamically allocate resources within or between data centers without being constrained by Layer 2 boundaries or being forced to create large or geographically stretched Layer 2 domains.

Using VXLANs to create smaller Layer 2 domains that are connected over a Layer 3 network means that you do not need to use Spanning Tree Protocol (STP) to converge the topology but can use more robust routing protocols in the Layer 3 network instead. In the absence of STP, none of your links are blocked, which means you can get full value from all the ports that you purchase. Using routing protocols to connect your Layer 2 domains also allows you to load-balance the traffic to ensure that you get the best use

of your available bandwidth. Given the amount of east-west traffic that often flows within or between data centers, maximizing your network performance for that traffic is very important.

The video *Why Use an Overlay Network in a Data Center?* presents a brief overview of the advantages of using VXLANs.



Video: [Why Use an Overlay Network in a Data Center?](#)

How Does VXLAN Work?

VXLAN is often described as an overlay technology because it allows you to stretch Layer 2 connections over an intervening Layer 3 network by encapsulating (tunneling) Ethernet frames in a VXLAN packet that includes IP addresses. Devices that support VXLANs are called *virtual tunnel endpoints (VTEPs)*—they can be end hosts or network switches or routers. VTEPs encapsulate VXLAN traffic and de-encapsulate that traffic when it leaves the VXLAN tunnel. To encapsulate an Ethernet frame, VTEPs add a number of fields, including the following fields:

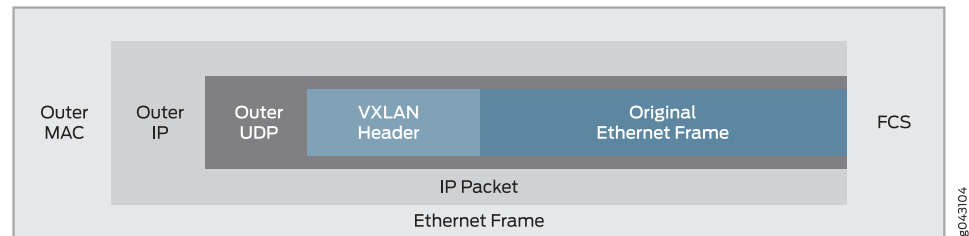
- Outer media access control (MAC) destination address (MAC address of the tunnel endpoint VTEP)
- Outer MAC source address (MAC address of the tunnel source VTEP)
- Outer IP destination address (IP address of the tunnel endpoint VTEP)
- Outer IP source address (IP address of the tunnel source VTEP)
- Outer UDP header
- A VXLAN header that includes a 24-bit field—called the *VXLAN network identifier (VNI)*—that is used to uniquely identify the VXLAN. The VNI is similar to a VLAN ID, but having 24 bits allows you to create many more VXLANs than VLANs.



NOTE: Because VXLAN adds 50 to 54 bytes of additional header information to the original Ethernet frame, you might want to increase the MTU of the underlying network. In this case, configure the MTU of the physical interfaces that participate in the VXLAN network, not the MTU of the logical VTEP source interface, which is ignored.

Figure 3 on page 8 shows the VXLAN packet format.

Figure 3: VXLAN Packet Format



VXLAN Implementation Methods

Junos OS supports implementing VXLANs in the following environments:

- **Manual VXLAN**—In this environment, a Juniper Networks device acts as a transit device for downstream devices acting as VTEPs, or a gateway that provides connectivity for downstream servers that host virtual machines (VMs), which communicate over a Layer 3 network. In this environment, software-defined networking (SDN) controllers are not deployed.



NOTE: QFX10000 switches do not support manual VXLANs.

- **OVSDB-VXLAN**—In this environment, SDN controllers use the Open vSwitch Database (OVSDB) management protocol to provide a means through which controllers (such as a VMware NSX or Juniper Networks Contrail controller) and Juniper Networks devices that support OVSDB can communicate.
- **EVPN-VXLAN**—In this environment, Ethernet VPN (EVPN) is a control plane technology that enables hosts (physical servers and VMs) to be placed anywhere in a network and remain connected to the same logical Layer 2 overlay network, and VXLAN creates the data plane for the Layer 2 overlay network.

Using QFX5100 and QFX5110 Switches with VXLANs

You can configure the switches to perform all of the following roles:

- In an environment without an SDN controller, act as a transit Layer 3 switch for downstream hosts acting as VTEPs. In this configuration, you do not need to configure any VXLAN functionality on the switch. You do need to configure IGMP and PIM so that the switch can form the multicast trees for the VXLAN multicast groups. (See [Manual VXLANs Require PIM on page 10](#) for more information.)
- In an environment with or without an SDN controller, act as a Layer 2 gateway between virtualized and nonvirtualized networks in the same data center or between data centers. For example, you can use the switch to connect a network that uses VXLANs to one that uses VLANs.
- Act as a Layer 2 gateway between virtualized networks in the same or different data centers and allow virtual machines to move (VMotion) between those networks and

data centers. For example, if you want to allow VMotion between devices in two different networks, you can create the same VLAN in both networks and put both devices on that VLAN. The switches connected to these devices, acting as VTEPs, can map that VLAN to the same VXLAN, and the VXLAN traffic can then be routed between the two networks.



NOTE: The QFX Series switches described in this section cannot route traffic between different VXLANs. To connect devices in different VXLANs you need a VXLAN-capable Layer 3 gateway, such as a Juniper Networks MX Series router, EX9200 switch, or QFX10000 switch.

Because the additional headers add 50 to 54 bytes, you might need to increase the MTU on a VTEP to accommodate larger packets. For example, if the switch is using the default MTU value of 1514 bytes and you want to forward 1500-byte packets over the VXLAN, you need to increase the MTU to allow for the increased packet size caused by the additional headers.

Changing the UDP Port on QFX5100 and QFX5110 Switches

Starting with Junos OS Release 14.1X53-D25 on QFX5100 switches and Junos OS Release 15.1X53-D210 on QFX5110 switches, you can configure the UDP port used as the destination port for VXLAN traffic. To configure the VXLAN destination port to be something other than the default UDP port of 4789, enter the following statement:

```
set protocols l2-learning destination-udp-port port-number
```

The port you configure will be used for all VXLANs configured on the switch.



NOTE: If you make this change on one switch in a VXLAN, you must make the same change on all the devices that terminate the VXLANs configured on your switch. If you do not do so, traffic will be disrupted for all the VXLANs configured on your switch. When you change the UDP port, the previously learned remote VTEPs and remote MACs are lost and VXLAN traffic is disrupted until the switch relearns the remote VTEPs and remote MACs.

Controlling Transit Multicast Traffic on QFX5100 and QFX5110 Switches

When the switch acting as a VTEP receives a broadcast, unknown unicast, or multicast packet, it performs the following actions on the packet:

1. It de-encapsulates the packet and delivers it to the locally attached hosts.
2. It then adds the VXLAN encapsulation again and sends the packet to the other VTEPs in the VXLAN.

These actions are performed by the loopback interface used as the VXLAN tunnel address and can, therefore, negatively impact the bandwidth available to the VTEP. With Junos OS Release 14.1X53-D30 and later, if you know that there are no multicast receivers

attached to other VTEPs in the VXLAN that want traffic for a specific multicast group, you can reduce the processing load on the loopback interface by entering the following statement:

set protocols l2-learning disable-vxlan-multicast-transit vxlan-multicast-group *multicast-group*

In this case, no traffic will be forwarded for the specified group but all other multicast traffic will be forwarded. If you do not want to forward any multicast traffic to other VTEPs in the VXLAN, enter the following statement:

set protocols l2-learning disable-vxlan-multicast-transit vxlan-multicast-group all

Using an MX Series Router, EX9200 Switch, or QFX10000 Switch as a VTEP

You can configure an MX Series router, EX9200 switch, or QFX10000 switch to act as a VTEP and perform all of the following roles:

- Act as a Layer 2 gateway between virtualized and nonvirtualized networks in the same data center or between data centers. For example, you can use an MX Series router to connect a network that uses VXLANs to one that uses VLANs.
- Act as a Layer 2 gateway between virtualized networks in the same or different data centers and allow virtual machines to move (VMotion) between those networks and data centers.
- Act as a Layer 3 gateway to route traffic between different VXLANs in the same data center.
- Act as a Layer 3 gateway to route traffic between different VXLANs in different data centers over a WAN or the Internet using standard routing protocols or virtual private LAN service (VPLS) tunnels.



NOTE: If you want one of the devices described in this section to be a VXLAN Layer 3 gateway, you must configure integrated routing and bridging (IRB) interfaces to connect the VXLANs, just as you do if you want to route traffic between VLANs.

Manual VXLANs Require PIM

In an environment with a controller (such as a VMware NSX or Juniper Networks Contrail controller), you can provision VXLANs on a Juniper Networks device. A controller also provides a control plane that VTEPs use to advertise their reachability and learn about the reachability of other VTEPs. You can also manually create VXLANs on Juniper Networks devices instead of using a controller. If you use this approach, you must also configure Protocol Independent Multicast (PIM) on the VTEPs so that they can create VXLAN tunnels between themselves.

You must also configure each VTEP in a given VXLAN to be a member of the same multicast group. (If possible, you should assign a different multicast group address to each VXLAN, although this is not required. Multiple VXLANs can share the same multicast group.) The VTEPs can then forward ARP requests they receive from their connected hosts to the multicast group. The other VTEPs in the group de-encapsulate the VXLAN

information, and (assuming they are members of the same VXLAN) they forward the ARP request to their connected hosts. When the target host receives the ARP request, it responds with its MAC address, and its VTEP forwards this ARP reply back to the source VTEP. Through this process, the VTEPs learn the IP addresses of the other VTEPs in the VXLAN and the MAC addresses of the hosts connected to the other VTEPs.

The multicast groups and trees are also used to forward broadcast, unknown unicast, and multicast (BUM) traffic between VTEPs. This prevents BUM traffic from being unnecessarily flooded outside the VXLAN.



NOTE: Multicast traffic that is forwarded through a VXLAN tunnel is sent only to the remote VTEPs in the VXLAN. That is, the encapsulating VTEP does not copy and send copies of the packets according to the multicast tree—it only forwards the received multicast packets to the remote VTEPs. The remote VTEPs de-encapsulate the encapsulated multicast packets and forward them to the appropriate Layer 2 interfaces. The remote VTEPs also do not copy and send copies of the packets according to the multicast tree.

Load Balancing VXLAN Traffic

On QFX5100 and QFX5110 switches, the Layer 3 routes that form VXLAN tunnels use per-packet load balancing by default, which means that load balancing is implemented if there are ECMP paths to the remote VTEP. This is different from normal routing behavior in which per-packet load balancing is not used by default. (Normal routing uses per-prefix load balancing by default.)

The source port field in the UDP header is used to enable ECMP load balancing of the VXLAN traffic in the Layer 3 network. This field is set to a hash of the inner packet fields, which results in a variable that ECMP can use to distinguish between tunnels (flows). (None of the other fields that flow-based ECMP normally uses are suitable for use with VXLANs. All tunnels between the same two VTEPs have the same outer source and destination IP addresses, and the UDP destination port is set to port 4789 by definition. Therefore, none of these fields provide a sufficient way for ECMP to differentiate flows.)

Related Documentation

- [Examples: Manually Configuring VXLANs on QFX Series Switches on page 61](#)
- [delete](#)
- [OVSDb Support on Juniper Networks Devices on page 12](#)
- [mtu](#)

Understanding the OVSDB Protocol Running on Juniper Networks Devices

The Juniper Networks Junos OS implementation of the Open vSwitch Database (OVSDB) management protocol provides a means through which Juniper Networks devices that support OVSDB can communicate with software-defined networking (SDN) controllers. Juniper Networks devices exchange control and statistical information with the SDN controllers, thereby enabling virtual machine (VM) traffic from the entities in a virtualized network to be forwarded to entities in a physical network, and vice versa.

The Junos OS implementation of OVSDB includes an OVSDB server and an OVSDB client, both of which run on each Juniper Networks device that supports OVSDB.

The OVSDB server on a Juniper Networks device can communicate with an OVSDB client on an SDN controller. To establish a connection between a Juniper Networks device and an SDN controller, you must specify information about the SDN controller (IP address) and the connection (port over which the connection occurs and the communication protocol to be used) on each Juniper Networks device. After the configuration is successfully committed, the connection is established between the management port of the Juniper Networks device and the SDN controller port that you specify in the Junos OS configuration.

The OVSDB server stores and maintains an OVSDB database schema, which is defined for physical devices. This schema contains control and statistical information provided by the OVSDB client on the Juniper Networks devices and on SDN controllers. This information is stored in various tables in the schema. The OVSDB client monitors the schema for additions, deletions, and modifications to this information, and the information is used for various purposes, such as learning the media access control (MAC) addresses of virtual hosts and physical servers.

The schema provides a means through which the Juniper Networks devices and the SDN controllers can exchange information. For example, the Juniper Networks devices capture MAC routes to entities in the physical network and push this information to a table in the schema so that SDN controllers with connections to these Juniper Networks devices can access the MAC routes. Conversely, SDN controllers capture MAC routes to entities in the virtualized network and push this information to a table in the schema so that Juniper Networks devices with connections to the SDN controllers can access the MAC routes.

Some of the OVSDB table names include the words *local* or *remote*, for example, *unicast MACs local table* and *unicast MACs remote table*. Information in *local* tables is learned by a Juniper Networks device that functions as a hardware virtual tunnel endpoint (VTEP), while information in *remote* tables is learned from other software or hardware VTEPs.

OVSDB Support on Juniper Networks Devices

Table 4 on page 13 lists the Juniper Networks devices that support the Open vSwitch Database (OVSDB) management protocol and the Junos OS releases in which OVSDB is supported.

The OVSDB software is included in the jsdn package. For some Juniper Networks devices, the jsdn package is included in the Junos OS software (jinstall) package. If the jsdn

package is not included in the jinstall package for a particular device, a separate jsdn package must be installed on the device in addition to the jinstall package.

For each device and Junos OS release, the table outlines whether or not the jsdn package is included in the jinstall package. If the jsdn package is not included, the table also includes the name of the separate jsdn package.



NOTE: The separate jsdn package release number must be the same as the jinstall release number running on the device.

Table 4: OVSDb Support on Juniper Networks Devices

Juniper Networks Device	Junos OS Release	jsdn Package Included in jinstall Package?	Separate jsdn Package Name
EX9200 Line of Ethernet Switches	14.2R1 and later	No	jsdn-i386-release jsdn-x86-release*
MX80 3D Universal Edge Routers	14.1R2 and later	No	jsdn-powerpc-release
MX104 3D Universal Edge Routers	14.2R4 and later	No	jsdn-powerpc-release
MX240, MX480, and MX960 3D Universal Edge Routers	14.1R2 and later	No	jsdn-i386-release jsdn-x86-release*
MX2010 and MX2020 3D Universal Edge Routers	15.1R2 and later	No	jsdn-i386-release jsdn-x86-release
QFX5100 Switches	14.1X53-D10 and later	Yes	—
QFX5110 Switches	15.1X53-D210 and later	Yes	—
QFX10002 Switches	15.1X53-D10	No	jsdn-i386-release
	15.1X53-D20 and later	Yes	—
QFX10008 Switches	15.1X53-D30 and later	Yes	—

*This jsdn package is introduced in Junos OS Release 15.1R2.

Related Documentation

- *Installing OVSDb on Juniper Networks Devices*

OVSDB Schema for Physical Devices

An Open vSwitch Database (OVSDB) server runs on a Juniper Networks device that supports the OVSDB management protocol. When this device is connected to one or more SDN controllers, the connections provide a means through which the Juniper Networks device and the SDN controllers can communicate.

Juniper Networks devices that support OVSDB and SDN controllers exchange control and statistical data. This data is stored in the OVSDB database schema defined for physical devices. The schema resides in the OVSDB server. The schema includes several tables. Juniper Networks devices and SDN controllers, both of which have OVSDB clients, can add rows to the tables as well as monitor the tables for the addition, deletion, and modification of rows.

For example, the OVSDB client on a Juniper Networks device and an SDN controller can collect MAC routes learned by entities in the physical or virtualized networks, respectively, and publish the routes to the appropriate table in the schema. By using the MAC routes and other information provided in the table, Juniper Networks devices in the physical network and entities in the virtualized network can determine where to forward virtual machine (VM) traffic.

Some of the OVSDB table names include the words *local* or *remote*—for example, the *unicast MACs local table* and the *unicast MACs remote table*. Information in *local* tables is learned by a Juniper Networks device that functions as a hardware virtual tunnel endpoint (VTEP), whereas information in *remote* tables is learned by other software or hardware VTEPs.

[Table 5 on page 14](#) describes the tables in the schema, the physical or virtual entity that is the source of the data provided in the table, and the command that you can enter in the CLI of the Juniper Networks device to get similar information.

Table 5: OVSDB Schema Tables

Table Name	Description	Source of Information	Command
Global table	Includes the top-level configuration for the Juniper Networks device.	Juniper Networks device	—
Manager table	Includes information about each SDN controller that is connected to the Juniper Networks device.	Juniper Networks device	show ovsdb controller
Physical switch table	Includes information about a Juniper Networks device that functions as a hardware VTEP. This table includes information only for the device on which the table resides.	Juniper Networks device	—

Table 5: OVSDb Schema Tables (*continued*)

Table Name	Description	Source of Information	Command
Physical port table	Includes information about OVSDb-managed interfaces.	Juniper Networks device	show ovssdb interface
Logical switch table	Includes the following information: <ul style="list-style-type: none"> Logical switches, which you configured in a VMware NSX environment, or virtual networks, which you configured in a Contrail environment. The equivalent VXLANs, which were configured on the Juniper Networks device. 	<ul style="list-style-type: none"> SDN controller Juniper Networks device 	show ovssdb logical-switch
Logical binding statistics table	Includes statistics for OVSDb-managed interfaces.	Juniper Networks device	show ovssdb statistics interface
Physical locator table	Includes information about Juniper Networks devices configured as hardware VTEPs, software VTEPs, and service nodes in an NSX environment.	Juniper Networks device	show ovssdb virtual-tunnel-end-point
Physical locator set table	Includes a list of software VTEPs, service nodes, or top-of-rack service nodes (TSNs) for a logical switch.	Juniper Networks device	—
Unicast MACs remote table	Reachability information, including unicast MAC addresses, for entities in the virtualized network.	SDN controller	show ovssdb mac
Unicast MACs local table	Reachability information, including unicast MAC addresses, for entities in the physical network.	Juniper Networks device	show ovssdb mac
Multicast MACs remote table	Includes only one row. In this row, the MAC column includes the keyword unknown dst along with a list of software VTEPs, service nodes, or TSNs, which handle multicast traffic.	SDN controller	show ovssdb mac

Table 5: OVSDB Schema Tables (*continued*)

Table Name	Description	Source of Information	Command
Multicast MACs local table	<p>NOTE: Only QFX Series switches support this table.</p> <p>Includes one row for each logical switch. In this row, the MAC column includes the keyword unknown dst and a list of hardware VTEPs, which are identified by the IP address assigned to the hardware VTEP loopback interface (lo0). These hardware VTEPs can terminate or originate a VXLAN tunnel.</p>	Juniper Networks device	<code>show ovbdb mac</code>

- Related Documentation**
- [Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 12](#)
 - [Understanding How to Set Up OVSDB Connections Between Juniper Networks Devices and SDN Controllers on page 24](#)

Understanding How Layer 2 BUM and Layer 3 Routed Multicast Traffic Are Handled with OVSDB

The Juniper Networks Junos OS implementation of the Open vSwitch Database (OVSDB) management protocol provides a means through which software-defined networking (SDN) controllers and Juniper Networks devices that support OVSDB can communicate.

This topic explains how a Juniper Networks device with Virtual Extensible LAN (VXLAN) and OVSDB management protocol capabilities handles the following types of traffic:

- (This scenario applies to all Juniper Networks devices that support VXLAN and OVSDB.) Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic that originates in an OVSDB-managed VXLAN and is forwarded to interfaces within the same VXLAN.



NOTE: You must explicitly configure the replication of unknown unicast traffic in a Contrail environment.

- (This scenario applies only to MX Series routers and EX9200 switches that support VXLAN and OVSDB.) Layer 3 multicast traffic that is received by an integrated routing and bridging (IRB) in an OVSDB-managed VXLAN and is forwarded to interfaces in another OVSDB-managed VXLAN.

By default, Layer 2 BUM traffic that originates in an OVSDB-managed VXLAN is handled by one or more software virtual tunnel endpoints (VTEPs), service nodes, or top-of-rack service nodes (TSNs) in the same VXLAN. (This topic refers to the software VTEPs, service nodes, and TSNs collectively as *replicators*.) The table for remote multicast media

access control (MAC) addresses in the OVSDB schema for physical devices contains only one entry that has the keyword **unknown-dst** as the MAC string and a list of replicators.

Given the previously described table entry, Layer 2 BUM traffic received on an interface in the OVSDB-managed VXLAN is forwarded to one of the replicators. The replicator to which a BUM packet is forwarded is determined by the Juniper Networks device on which the OVSDB-managed VXLAN is configured. On receiving the BUM packet, the entity replicates the packet and forwards the replicas to all interfaces within the VXLAN.

Instead of using replicators, you can optionally enable ingress node replication to handle Layer 2 BUM traffic on Juniper Networks devices that support OVSDB.



NOTE: For VXLAN-OVSDB, ingress node replication is supported on all Juniper Networks devices that support OVSDB except the QFX Series switches.

With ingress node replication enabled, on receiving a Layer 2 BUM packet on an interface in an OVSDB-managed VXLAN, the Juniper Networks device replicates the packet and then forwards the replicas to all software VTEPs included in the unicast MACs remote table in the OVSDB schema. The software VTEPs then forward the replicas to all virtual machines (VMs), except service VMs or nodes, on the same host.



NOTE: When Juniper Networks devices replicate Layer 2 BUM packets to a large number of remote software VTEPs, the performance of the Juniper Networks devices can be impacted.

On IRB interfaces that forward Layer 3 multicast traffic from one OVSDB-managed VXLAN to another, ingress node replication is automatically implemented. With ingress node replication, the Juniper Networks device replicates a Layer 3 multicast packet and then the IRB interface forwards the replicas to all hardware and software VTEPs, but not to service nodes, in the other OVSDB-managed VXLAN. For the routing of Layer 3 multicast traffic from one OVSDB-managed VXLAN to another, ingress node replication is the only option and does not need to be configured.

PART 2

Configuring OVSDB and VXLAN

- [Configuring OVSDB-Managed VXLANs with an SDN Controller on page 21](#)
- [Configuring VXLANs Without an SDN Controller on page 57](#)

CHAPTER 2

Configuring OVSDb-Managed VXLANs with an SDN Controller

- [OVSDb and VXLAN Configuration Workflows for VMware NSX Environment on page 21](#)
- [Understanding How to Set Up OVSDb Connections on a Juniper Networks Device on page 24](#)
- [Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers on page 25](#)
- [Setting Up OVSDb on Juniper Networks Devices That Support the Dynamic Configuration of VXLANs on page 27](#)
- [Understanding Dynamically Configured VXLANs in an OVSDb Environment on page 28](#)
- [VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints on page 35](#)
- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDb Connections in a VMware NSX Environment \(Trunk Interfaces Supporting Untagged Packets\) on page 38](#)
- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDb Connections in a VMware NSX Environment \(Trunk Interfaces Supporting Tagged Packets\) on page 46](#)
- [Verifying That a Logical Switch and Corresponding Junos OS OVSDb-Managed VXLAN Are Working Properly on page 54](#)

OVSDb and VXLAN Configuration Workflows for VMware NSX Environment

The workflow that you use to configure Open vSwitch Database (OVSDb) and Virtual Extensible LAN (VXLAN) in a VMware NSX environment depends on the Juniper Networks device that you are configuring. This topic provides more information about the following workflows:

- [OVSDb and VXLAN Configuration Workflow for QFX Series Switches on page 21](#)
- [OVSDb and VXLAN Configuration Workflow for MX Series Routers and EX9200 Switches on page 23](#)

OVSDb and VXLAN Configuration Workflow for QFX Series Switches

[Table 6 on page 22](#) provides a high-level workflow of the tasks that you must perform to configure OVSDb and VXLAN on QFX Series switches. You must perform the tasks in [Table 6 on page 22](#) for each Juniper Networks switch that you plan to deploy in an OVSDb

environment. In general, the successful completion of a task in this workflow depends on the successful completion of the previous task, so it is important to adhere to the task sequence provided in [Table 6 on page 22](#).

Table 6: OVSDB and VXLAN Configuration Workflow for QFX Series Switches

Sequence	Task	For More Information
1	Create and install a Secure Sockets Layer (SSL) key and certificate.	“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers” on page 25.
2	Enter the set switch-options ovssdb-managed configuration mode command on the Juniper Networks switch.	—
3	Explicitly configure a connection to at least one VMware NSX controller.	“Setting Up the OVSDB Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs” on page 27.
4	Specify that each physical interface associated with a VXLAN is to be managed by OVSDB.	“Setting Up the OVSDB Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs” on page 27.
5	Configure a logical switch for each OVSDB-managed VXLAN that you plan to implement.	See the VMware documentation that accompanies NSX Manager or the NSX API.
6	<ul style="list-style-type: none"> For each Juniper Network switch on which OVSDB-managed VXLANs and interfaces are configured, create a gateway. For each OVSDB-managed interface that you configure, create a gateway service. For each logical interface that you plan to implement for a VXLAN, configure a logical switch port. 	<p>For general information about configuring gateways, gateway services, and logical switch ports, see the VMware documentation that accompanies NSX Manager or the NSX API.</p> <p>For key NSX Manager configuration details that help you configure gateways, gateway services, and logical switch ports so they function properly with their physical counterparts, see “VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints” on page 35.</p>
7	<p>Configure the loopback interface (lo0) on the Juniper Networks switch for VXLAN by entering the following configuration mode commands:</p> <ul style="list-style-type: none"> set interfaces lo0 unit 0 family inet address <i>ip-address</i> primary set switch-options vtep-source-Interface lo0.0 	—

After you successfully complete task 6 in [Table 6 on page 22](#), the Juniper Networks switch dynamically creates a VXLAN for each logical switch that you configured in task 5. The Juniper Networks switch also dynamically creates and associates interfaces with each VXLAN. The dynamically created interface configuration is based on the gateway service and logical switch ports that you configured in task 6. For more information, see [“Understanding Dynamically Configured VXLANs in an OVSDB Environment” on page 28.](#)

For OVSDb-VXLAN scenarios in which Juniper Networks switches are commonly deployed, see the following topics:

- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDb Connections in a VMware NSX Environment \(Trunk Interfaces Supporting Untagged Packets\)](#) on page 38
- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDb Connections Between Virtual and Physical Entities in a Data Center \(Using Trunk Interfaces\)](#) on page 46

OVSDb and VXLAN Configuration Workflow for MX Series Routers and EX9200 Switches

[Table 7 on page 23](#) provides a high-level workflow of the tasks that you must perform to configure OVSDb and VXLAN on MX Series routers and EX9200 switches. You must perform the tasks in [Table 7 on page 23](#) for each Juniper Networks device that you plan to deploy in an OVSDb environment. In general, the successful completion of a task in this workflow depends on the successful completion of the previous task, so it is important to adhere to the task sequence provided in [Table 7 on page 23](#).

Table 7: OVSDb and VXLAN Configuration Workflow for MX Series Routers and EX9200 Switches

Sequence	Task	For More Information
1	Install the jsdn software package.	<i>Installing OVSDb on Juniper Networks Devices.</i>
2	Create and install an SSL key and certificate.	“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers” on page 25.
3	Explicitly configure a connection to at least one NSX controller.	<i>Setting Up the OVSDb Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs.</i>
4	Specify that each physical interface associated with a VXLAN is to be managed by OVSDb.	<i>Setting Up the OVSDb Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs.</i>
5	Configure a logical switch for each OVSDb-managed VXLAN that you plan to implement.	See the VMware documentation that accompanies NSX Manager or the NSX API.
6	Configure OVSDb-managed VXLANs.	<i>Configuring OVSDb-Managed VXLANs.</i>
7	<p>For each Juniper Network device on which OVSDb-managed VXLANs and interfaces will be configured, create a gateway.</p> <p>For each OVSDb-managed interface that you configure, create a gateway service.</p> <p>For each logical interface that you plan to implement for a VXLAN, configure a logical switch port.</p>	<p>For general information about configuring gateways, gateway services, and logical switch ports, see the VMware documentation that accompanies NSX Manager or the NSX API.</p> <p>For key NSX Manager configuration details that help you configure gateways, gateway services, and logical switch ports, so that they function properly with their physical counterparts, see “VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints” on page 35.</p>

Table 7: OVSDB and VXLAN Configuration Workflow for MX Series Routers and EX9200 Switches (*continued*)

Sequence	Task	For More Information
8	<p>Configure the loopback interface (lo0) on the Juniper Networks device for VXLAN by entering the following configuration mode commands:</p> <ul style="list-style-type: none"> • set interfaces lo0 unit 0 family inet address <i>ip-address</i> primary • set switch-options vtep-source-Interface lo0.0 	–

For OVSDB-VXLAN scenarios in which these Juniper Networks devices are commonly deployed, see the following topics:

- *Example: Setting Up Inter-VXLAN Unicast Routing and OVSDB Connections in a Data Center*
- *Example: Setting Up Inter-VXLAN Unicast and Multicast Routing and OVSDB Connections in a Data Center*
- *Example: Configuring VXLAN to VPLS Stitching with OVSDB*

Related Documentation • [Verifying That a Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN Are Working Properly on page 54](#)

Understanding How to Set Up OVSDB Connections on a Juniper Networks Device

The Juniper Networks Junos OS implementation of the Open vSwitch Database (OVSDB) management protocol provides a means through which Juniper Networks devices that support OVSDB can communicate with software-defined networking (SDN) controllers. A Juniper Networks device exchanges control and statistical data with each SDN controller to which it is connected.

You can connect a Juniper Networks device to more than one SDN controller for redundancy.

In a VMware NSX environment, one cluster of NSX controllers typically includes three or five controllers. To implement the OVSDB management protocol on a Juniper Networks device, you must explicitly configure a connection to one SDN controller, using the Junos OS CLI. If the SDN controller to which you explicitly configure a connection is in a cluster, the controller pushes information about other controllers in the same cluster to the device, and the device establishes connections with the other controllers. However, you can also explicitly configure connections with the other controllers in the cluster, using the Junos OS CLI.

To implement the OVSDB management protocol on a Juniper Networks device in a Contrail environment, you must configure a connection to a Contrail controller, using the Junos OS CLI.

Connections to all SDN controllers are made on the management interface of the Juniper Networks device. To set up a connection between a Juniper Networks device and an SDN controller, you need to configure the following parameters on the Juniper Networks device:

- IP address of the SDN controller.
- The protocol that secures the connection. Secure Sockets Layer (SSL) is the supported protocol.



NOTE: The SSL connection requires a private key and certificates, which must be stored in the `/var/db/certs` directory of the Juniper Networks device. See [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers”](#) on page 25.

- Number of the port over which the connection is made. The port number of the default port is 6632.

Optionally, you can configure the following connection timers on the Juniper Networks device:

- Inactivity probe duration—The maximum amount of time, in milliseconds, that the connection can be inactive before an inactivity probe is sent. The default value is 0 milliseconds, which means that an inactivity probe is never sent.
- Maximum backoff duration—If an attempt to connect to an SDN controller fails, the maximum amount of time, in milliseconds, before the device can make the next attempt. The default value is 1000 milliseconds.

Related Documentation

- [Setting Up the OVSDb Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs](#)
- [Setting Up the OVSDb Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs](#) on page 27

Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers

To secure a connection between a Juniper Networks device that supports the Open vSwitch Database (OVSDb) management protocol and one or more software-defined networking (SDN) controllers, the following Secure Sockets Layer (SSL) files must be present in the `/var/db/certs` directory on the device:

- `vtep-privkey.pem`
- `vtep-cert.pem`
- `ca-cert.pem`

You must create the `vtep-privkey.pem` and `vtep-cert.pem` files for the device and then install the two files in the `/var/db/certs` directory on the device.

Upon initial connection between a Juniper Networks device with OVSDB implemented and an SDN controller, the **ca-cert.pem** file is automatically generated and then installed in the **/var/db/certs** directory on the device.



NOTE: The situation at your particular site determines the possible methods that you can use to create the **vtep-privkey.pem** and **vtep-cert.pem** files and install them in the Juniper Networks device. Instead of providing procedures for all possible situations, this topic provides a procedure for one common scenario.

The procedure provided in this topic uses the OpenFlow public key infrastructure (PKI) management utility **ovs-pki** on a Linux computer to initialize a PKI and create the **vtep-privkey.pem** and **vtep-cert.pem** files. (If you have an existing PKI on your Linux computer, you can skip the step to initialize a new one.) By default, the utility initializes the PKI and places these files in the **/usr/local/share/openvswitch/pki** directory of the Linux computer.

To create and install an SSL key and certificate on a Juniper Networks device:

1. Initialize a PKI if one does not already exist on your Linux computer.

```
# ovs-pki init
```

2. On the same Linux computer on which the PKI exists, create a new key and certificate for the Juniper Networks device.

```
# ovs-pki req+sign vtep
```

3. Copy only the **vtep-privkey.pem** and **vtep-cert.pem** files from the Linux computer to the **/var/db/certs** directory on the Juniper Networks device.

**Related
Documentation**

- [Understanding How to Set Up OVSDB Connections Between Juniper Networks Devices and SDN Controllers on page 24](#)
- [OVSDB and VXLAN Configuration Workflows for VMware NSX Environment on page 21](#)

Setting Up OVSDb on Juniper Networks Devices That Support the Dynamic Configuration of VXLANs

To implement the Open vSwitch Database (OVSDb) management protocol on a Juniper Networks device, you must configure a connection between the Juniper Networks device and a software-defined networking (SDN) controller using the Junos OS CLI.

All SDN controller connections are made on the management interface of the Juniper Networks device. This connection is secured by using the Secure Sockets Layer (SSL) protocol. The default port number for the connection is 6632.

You must also specify that each physical interface that is connected to a physical server is managed by OVSDb. By performing this configuration, you essentially disable the Juniper Networks device from learning about other Juniper Networks devices that function as hardware virtual tunnel endpoints (VTEPs) and the MAC addresses learned by the hardware VTEPs. Instead, this configuration enables OVSDb to learn about these elements.

Before setting up OVSDb on a Juniper Networks device, you must do the following:

- Create an SSL private key and certificate, if they do not already exist, and install them in the `/var/db/certs` directory of the Juniper Networks device. See [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers”](#) on page 25.

To set up OVSDb on a Juniper Networks device:

1. Specify the IP address of the SDN controller.


```
[edit protocols ovbdb]
user@host# set controller ip-address
```
2. Specify SSL as the protocol that secures the connection between the Juniper Networks device and the SDN controller.


```
[edit protocols ovbdb]
user@host# set controller ip-address protocol ssl
```
3. Set the number of the port over which the connection to the SDN controller is made.


```
[edit protocols ovbdb]
user@host# set controller ip-address protocol ssl port number
```
4. (Optional) Specify (in milliseconds) how long the connection can be inactive before an inactivity probe is sent.


```
[edit protocols ovbdb]
user@host# set controller ip-address inactivity-probe-duration milliseconds
```
5. (Optional) Specify (in milliseconds) how long the device must wait before it can try to connect to the SDN controller again if the previous attempt failed.


```
[edit protocols ovbdb]
user@host# set controller ip-address maximum-backoff-duration milliseconds
```
6. (Optional) Repeat Steps 1 through 5 to configure a connection to an additional SDN controller in the NSX environment.
7. Specify that each physical interface that is connected to a physical server is managed by OVSDb.

```
[edit protocols ovssdb]
user@host# set interfaces interface-name
```

When specifying the *interface-name*, you do not need to include a logical unit number.

8. Complete the remaining configuration tasks. The remaining tasks for an NSX environment are described in [“OVSDB and VXLAN Configuration Workflows for VMware NSX Environment” on page 21](#)).

Understanding Dynamically Configured VXLANs in an OVSDB Environment



NOTE: This topic applies only to QFX Series switches, which support the dynamic configuration of Open vSwitch Database (OVSDB)-managed Virtual Extensible LANs (VXLANs). Although the configuration of OVSDB-managed VXLANs is automated on these switches, there are tasks that you must perform before and after the dynamic configuration.

On all other Juniper Networks devices that support OVSDB and VXLAN, you must manually configure OVSDB-managed VXLANs using the Junos OS CLI. For more information about manually configuring OVSDB-managed VXLANs, see [Configuring OVSDB-Managed VXLANs](#).

The Juniper Networks Junos OS implementation of the OVSDB management protocol provides a means through which Juniper Networks devices that support OVSDB can communicate with software-defined networking (SDN) controllers. Support for OVSDB enables the devices in a physical network to be integrated into a virtualized network.

In a Junos OS environment, the concept of an OVSDB-managed Layer 2 broadcast domain in which data flows are limited to that domain is known as a *VXLAN*. The term used for the same concept in other OVSDB environments depends on the environment:

- In an NSX environment, the same concept is known as a *logical switch*.
- In a Contrail environment, the same concept is known as a *virtual network*.

Understanding the terminology used in the different environments will help you to better understand the workflow associated with the dynamic configuration of OVSDB-managed VXLANs, including tasks that you must perform before and after the dynamic configuration.

The following topics describe the dynamic configuration of OVSDB-managed VXLANs:

- [Performing Tasks Before and After the Dynamic Configuration of OVSDB-Managed VXLANs on page 28](#)
- [What the Juniper Networks Switch Actually Creates Dynamically on page 33](#)

Performing Tasks Before and After the Dynamic Configuration of OVSDB-Managed VXLANs

Although the configuration of OVSDB-managed VXLANs is automated, there are some tasks that you must perform before and after the dynamic configuration.

[Table 8 on page 29](#) includes a sequentially ordered workflow of tasks and events for the

dynamic configuration of OVSDb-managed VXLANs in an NSX environment, while [Table 9 on page 31](#) includes the equivalent information for a Contrail environment. Your familiarity with these workflows will ensure that the dynamic configuration of OVSDb-managed VXLANs is properly implemented.

In [Table 8 on page 29](#), the NSX controller and Juniper Networks switch handle the events described in workflow numbers 4, 6, and 7. You must perform the tasks described in workflow numbers 1, 2, 3, 5, and 8. If you perform a task in a different order than that outlined in [Table 8 on page 29](#), the dynamic configuration might not work or the dynamically configured OVSDb-managed VXLAN might not become functional.

Table 8: Workflow of Tasks and Events for the Dynamic Configuration of OVSDb-Managed VXLANs in an NSX Environment

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
1	Enable the Juniper Networks switch to dynamically configure an OVSDb-managed VXLAN.	You must manually enable this capability by entering the set switch-options ovbdb-managed configuration mode command on the switch.	—
2	On the Juniper Networks switch, configure each physical interface that is connected to a physical server so that the interface is managed by OVSDb.	For each physical interface, you must manually enter the set protocols ovbdb interfaces interface-name configuration mode command.	When entering the interface name, you do not need to include a logical unit number.
3	For each OVSDb-managed VXLAN that you want to implement, configure a logical switch.	You must manually configure the logical switch by using NSX Manager or the NSX API. See the documentation that accompanies NSX Manager or the NSX API.	A universally unique identifier (UUID) for the logical switch is dynamically generated.
4	Relevant information about the logical switch is pushed to the Juniper Networks switch.	The NSX controller pushes relevant information to the logical switch table in the OVSDb schema for physical devices. This schema resides in the Juniper Networks switch.	—

Table 8: Workflow of Tasks and Events for the Dynamic Configuration of OVSDB-Managed VXLANs in an NSX Environment (*continued*)

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
5	<p>Create the following entities:</p> <ul style="list-style-type: none"> For each Juniper Networks switch that you deploy as a hardware VTEP, you create a gateway. For each OVSDB-managed interface that you configured in workflow number 2, you create a gateway service. For each interface that you plan to implement for a VXLAN, configure a logical switch port. 	You must manually configure these entities by using NSX Manager or the NSX API. See the documentation that accompanies NSX Manager or the NSX API. Also see “VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints” on page 35.	–
6	Relevant information about the gateway service and logical switch port are pushed to the Juniper Networks switch.	The NSX controller pushes this information to the Juniper Networks switch.	–
7	A corresponding VXLAN is dynamically created. Based on the gateway service and logical switch port configured in NSX Manager or the NSX API, one or more interfaces are also created and associated with the VXLAN.	The Juniper Networks switch dynamically creates the VXLAN and interface configuration.	For the name of the VXLAN, the Juniper Networks switch uses the UUID of the logical switch.
8	(Recommended) Verify that the logical switch, corresponding VXLAN, and associated interfaces are configured properly and are operational.	You can enter the show ovssdb logical-switch operational mode command on the Juniper Networks switch. In the output, check the Flags field for the logical switches that you configured as described in workflow number 3 to ensure that it displays Created by both.	If the output of the show ovssdb logical-switch operational mode command does not include the Created by both state, see “Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN” on page 73.

In [Table 9 on page 31](#), the Contrail controller and Juniper Networks switch handle the events described in workflow numbers 5, 8, and 9. You must perform all other tasks described in the table. If you perform a task in a different order than that outlined in [Table 9 on page 31](#), the dynamic configuration might not work or the dynamically configured OVSDB-managed VXLAN might not become functional.



NOTE: Although you can perform the Contrail configurations outlined in [Table 9 on page 31](#) in the Contrail Web user interface or in the Contrail REST API, [Table 9 on page 31](#) only describes how to perform tasks in the Contrail Web user interface.

Table 9: Workflow of Tasks and Events for the Dynamic Configuration of OVSDb-Managed VXLANs in a Contrail Environment

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
1	On the Juniper Networks switch, configure a unique hostname for the switch.	You must manually enter the set system host-name <i>host-name</i> configuration mode command on the switch.	If implementing a virtual chassis, be aware that all members of the virtual chassis must have the same hostname.
2	Enable the Juniper Networks switch to dynamically configure an OVSDb-managed VXLAN.	You must manually enable this capability by entering the set switch-options <i>ovsdb-managed</i> configuration mode command on the switch.	—
3	On the Juniper Networks switch, configure each physical interface that is connected to a physical server so that the interface is managed by OVSDb.	For each physical interface, you must manually enter the set protocols ovsdb interfaces <i>interface-name</i> configuration mode command.	When entering the interface name, you do not need to include a logical unit number.
4	For each OVSDb-managed VXLAN that you want to implement, configure a virtual network in the Contrail Web user interface.	You must manually configure the virtual network by navigating to Configure > Networking > Networks. See Creating a Virtual Network .	See <i>Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs</i> .
5	Relevant information about the virtual network is pushed to the Juniper Networks switch.	The Contrail controller pushes relevant information to the logical switch table in the OVSDb schema for physical devices. This schema resides in the Juniper Networks switch.	—

Table 9: Workflow of Tasks and Events for the Dynamic Configuration of OVSDB-Managed VXLANs in a Contrail Environment (*continued*)

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
6	For each interface that you plan to implement for a VXLAN, configure a logical interface.	<p>In the Contrail Web user interface, you must manually configure the logical interface by navigating to Configure > Physical Devices > Interfaces.</p> <p>For information about configuring a logical interface, see Using TOR Switches and OVSDB to Extend the Contrail Cluster to Other Instances.</p>	See <i>Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs</i> .
7	For each Juniper Networks switch that you deploy as a hardware VTEP, you create a physical router.	<p>In the Contrail Web user interface, you must manually configure the physical router by navigating to Configure > Physical Devices > Physical Routers.</p> <p>For information about configuring a physical router, see Using TOR Switches and OVSDB to Extend the Contrail Cluster to Other Instances.</p>	See <i>Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs</i> .
8	Relevant information about the logical interfaces is pushed to the Juniper Networks switch.	The Contrail controller pushes this information to the Juniper Networks switch.	—
9	A corresponding VXLAN is dynamically created. Based on the logical interface configured in the Contrail Web user interface, one or more interfaces are also created and associated with the VXLAN.	The Juniper Networks switch dynamically creates the VXLAN and interface configurations.	For the name of the VXLAN, the Juniper Networks switch uses the prefix “Contrail-” and the UUID of the virtual network.
10	(Recommended) Verify that the virtual network, corresponding VXLAN, and interfaces are configured properly and are operational.	You can enter the show ovssdb logical-switch operational mode command on the Juniper Networks switch. In the output, check the Flags field for the virtual network that you configured as described in workflow number 4 to ensure that it displays Created by both.	If the output of the show ovssdb logical-switch operational mode command does not include the Created by both state, see “Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN” on page 73.

What the Juniper Networks Switch Actually Creates Dynamically

When a Juniper Networks switch creates a VXLAN, it sets up a configuration similar to the following sample:

```
set vlans 28805c1d-0122-495d-85df-19abd647d772 vxlan vni 100
```

Note the following meanings for this sample configuration:

- The name of the VXLAN is 28805c1d-0122-495d-85df-19abd647d772. The UUID of the logical switch, which was configured in NSX Manager or in the NSX API, is 28805c1d-0122-495d-85df-19abd647d772. For a VXLAN created in a Contrail environment, the name would be preceded by “Contrail-”.
- For the virtual network identifier (VNI), the Juniper Networks switch uses either the VNI specified in the logical switch configuration (NSX) or the VXLAN identifier specified in the virtual network configuration (Contrail). In this example, VNI 100 is used. If the Juniper Networks switch detects that VNI 100 is a duplicate of a VNI from a VXLAN configured by manually using the **set vlans *vlan-name* vxlan vni (1–16777214)** command in the Junos OS CLI, the switch deletes the manually configured VXLAN. Or, if the Juniper Networks switch detects that VNI 100 is specified in the dynamically configured VXLAN, but for some reason, the VNI is no longer in the equivalent logical switch or virtual network configuration, the Juniper Networks switch deletes VNI 100 from the VXLAN.

If you need to modify or delete an OVSDb-managed VXLAN that was dynamically configured by the Juniper Networks switch, you must modify or delete either the corresponding logical switch configuration (NSX), or the corresponding virtual network configuration (Contrail). After you modify or delete the configuration, the SDN controller pushes the update to the Juniper Networks switch, and the switch modifies or deletes its configuration accordingly.

Depending on either the gateway service and logical switch ports configuration (NSX), or the logical interface configuration (Contrail), the Juniper Networks switch dynamically creates and associates one or more interfaces with the VXLAN. The configuration generated by the switch depends on whether an interface must support untagged or tagged packets. The following sections provide information about the configuration that the switch dynamically generates for each interface:

- [Dynamic Association of a Trunk Interface Supporting Untagged Packets to a Dynamically Created VXLAN on page 33](#)
- [Dynamic Association of a Trunk Interface Supporting Tagged Packets to a Dynamically Created VXLAN on page 34](#)

Dynamic Association of a Trunk Interface Supporting Untagged Packets to a Dynamically Created VXLAN

To determine the type of interface to create and associate with an OVSDb-managed VXLAN, the Juniper Networks switch uses the VLAN ID that you specified when configuring either the logical switch port (NSX), or the logical interface (Contrail). If you specified **0** as the VLAN ID, the switch dynamically configures a trunk interface that can handle

untagged packets. (If you specified a valid VLAN ID other than 0, the switch creates a trunk interface that handles tagged packets.)

After the SDN controller pushes either the NSX or Contrail configurations to the Juniper Networks switch, the switch dynamically creates a configuration similar to the following:

```
set interfaces ge-1/0/0 flexible-vlan-tagging
set interfaces ge-1/0/0 native-vlan-id 4094
set interfaces ge-1/0/0 encapsulation extended-vlan-bridge
set interfaces ge-1/0/0 unit 0 vlan-id 4094
set vlans 28805c1d-0122-495d-85df-19abd647d772 interface ge-1/0/0.0
```

This sample configuration sets up physical interface ge-1/0/0 as a trunk interface. It also configures a native VLAN with an ID of 4094 and specifies that logical interface ge-1/0/0.0 is a member of the native VLAN. As a result, logical interface ge-1/0/0.0 handles incoming untagged packets.



NOTE: We reserve VLAN ID 4094 for native VLANs in an OVSDB environment. As a result, when you create either a logical switch port (NSX) or a logical interface (Contrail), if you specify VLAN ID 4094, the Juniper Networks switch does not dynamically configure a corresponding interface. Also, a system log error message is generated.

Instead of dynamically configuring physical interface ge-1/0/0 as an access interface, which typically handles untagged packets, the Juniper Networks switch configures it as a trunk interface. The intent of this configuration is to support the division of physical interface ge-1/0/0 into multiple logical interfaces, some of which are associated with VXLANs that handle untagged packets and some of which are associated with VXLANs that handle tagged packets.

The sample configuration also creates logical interface ge-1/0/0.0 and associates this interface with VXLAN 28805c1d-0122-495d-85df-19abd647d772.

Dynamic Association of a Trunk Interface Supporting Tagged Packets to a Dynamically Created VXLAN

In a network that is divided into multiple VXLANs, each VXLAN has a VLAN ID associated with it. Packets associated with a particular VXLAN include the corresponding tag. In this situation, the interface that connects the Juniper Networks switch to a physical server in an OVSDB environment is a trunk interface that handles only tagged packets.

To determine the type of interface to create and associate with an OVSDB-managed VXLAN, the Juniper Networks switch uses the VLAN ID that you specified when configuring either the logical switch port (NSX), or the logical interface (Contrail). If you specified a valid VLAN ID other than 0 in either configuration, the switch creates a trunk interface that can handle tagged packets. (If you specified 0 as the VLAN ID, the switch creates a trunk interface that handles untagged packets.)

After the SDN controller pushes the NSX or Contrail configuration to the Juniper Networks switch, the switch dynamically creates a configuration similar to the following:

```
set interfaces ge-1/0/0 flexible-vlan-tagging
set interfaces ge-1/0/0 encapsulation extended-vlan-bridge
set interfaces ge-1/0/0 unit 10 vlan-id 10
set vlans 28805c1d-0122-495d-85df-19abd647d772 interfaces ge-1/0/0.10
```

The sample configuration sets up physical interface ge-1/0/0 as a trunk interface. It also configures a VLAN with an ID of 10 and specifies that interface ge-1/0/0.10 is a member of the VLAN. With the configuration of VLAN 10, logical interface ge-1/0/0.10 accepts incoming packets with a VLAN tag of 10 and adds a tag of 100 to each packet. Adding a tag of 100 identifies the packets as received by the VXLAN 28805c1d-0122-495d-85df-19abd647d772, which has a VNI of 100. This configuration also associates the trunk interface with VXLAN 28805c1d-0122-495d-85df-19abd647d772.

- Related Documentation**
- [Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 12](#)
 - [show ovssdb logical-switch on page 106](#)

VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints

When implementing the Open vSwitch Database (OVSDB) management protocol and Virtual Extensible LANs (VXLANs) on a Juniper Networks device, you must perform the following tasks in VMware NSX Manager or in the NSX API:

- For each Juniper Networks device on which OVSDB-managed VXLANs and physical interfaces are configured, you must create an NSX-equivalent entity, which is known as a *gateway*.
- For each OVSDB-managed physical interface that you configure on a Juniper Networks device, you must configure a gateway service—for example, a VTEP Layer 2 gateway service.
- For each logical interface that you want to implement for a VXLAN, you must configure a logical switch port.

The configurations described in this topic enable connectivity between physical servers in the physical network and virtual machines (VMs) in the virtual network.

This topic provides a high-level summary of the tasks that you must perform to create a gateway, gateway service, and logical switch ports. Although you can create these virtual entities either in NSX Manager or in the NSX API, this topic only describes how to perform the tasks in NSX Manager. Also, this topic does not include a complete procedure for each task. Rather, it includes key NSX Manager configuration details for ensuring the correct configuration of the virtual entities so that they function properly with the physical entities.

For complete information about performing the tasks described in this topic, see the documentation that accompanies NSX Manager.

This topic describes the following tasks:

- [Creating a Gateway on page 36](#)
- [Creating a Gateway Service on page 36](#)
- [Creating a Logical Switch Port on page 37](#)

Creating a Gateway

In NSX Manager, you must create a gateway for each Juniper Networks device on which OVSDB-managed VXLANs and physical interfaces are configured. [Table 10 on page 36](#) provides a summary of key configuration fields in NSX Manager and how to configure them when creating a gateway.

Table 10: Key Configurations to Create a Gateway in NSX Manager

NSX Manager Configuration Page or Dialog Box	NSX Manager Configuration Field	How to Configure
Type	Transport Node Type	Select Gateway .
Properties	VTEP Enabled	Select VTEP Enabled .
Credential	Type	Select Management Address .
Credential	Management Address	Specify the management IP address of the Juniper Networks device.
Connections/Create Transport Connector	Transport Type	Select VXLAN .
Connections/Create Transport Connector	Transport Zone UUID	Select the UUID of an existing transport zone, or create a new transport zone.
Connections/Create Transport Connector	IP Address	Specify the IP address of the loopback interface (lo0) of the Juniper Networks device.

Creating a Gateway Service

In NSX Manager, you must create a gateway service for each OVSDB-managed physical interface that you configure on a Juniper Networks device. Creating a gateway service essentially does the following for each OVSDB-managed physical interface:

- Specifies a gateway service—for example, a VTEP Layer 2 gateway service.
- Binds the interface to a gateway that you created in [“Creating a Gateway” on page 36](#).

Before you start this task, you must complete the following configurations:

- A gateway for the Juniper Networks device on which the OVSDB-managed physical interfaces are configured. See [“Creating a Gateway” on page 36](#).

- The OVSDb-managed physical interfaces on the Juniper Networks device. For information about configuring OVSDb-managed physical interfaces on Juniper Networks devices, see [“Setting Up the OVSDb Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs” on page 27](#) or [Setting Up the OVSDb Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs](#).

[Table 11 on page 37](#) provides a summary of key configuration fields in NSX Manager and how to configure them when creating a gateway service.

Table 11: Key Configurations to Create a Gateway Service in NSX Manager

NSX Manager Configuration Page or Dialog Box	NSX Manager Configuration Field	How to Configure
Type	Gateway Service Type	Select VTEP L2 Gateway Service .
Transport Nodes/Edit Gateway	Transport Node	Select the gateway that you created for the Juniper Networks device.
Transport Nodes/Edit Gateway	Port ID	Select an OVSDb-managed physical interface configured on the Juniper Networks device.

Creating a Logical Switch Port

In NSX Manager, you must create a logical switch port for each logical interface that you plan to implement for a VXLAN. Creating the logical switch port essentially does the following for each logical interface:

- Binds the logical switch port to a logical switch that you created in NSX Manager or in the NSX API.
- Binds the logical interface to a gateway service that you configured in [“Creating a Gateway Service” on page 36](#).

Before you start this task, you must complete the following configurations:

- A logical switch with which this logical port is associated. For information about configuring a logical switch, see the VMware documentation that accompanies NSX Manager or the NSX API.
- A gateway service that specifies the OVSDb-managed physical interface with which the logical interface is associated. See [“Creating a Gateway Service” on page 36](#).

[Table 12 on page 37](#) provides a summary of key configuration fields in NSX Manager and how to configure them when creating a logical switch port.

Table 12: Key Configurations to Create a Logical Switch Port in NSX Manager

NSX Manager Configuration Page or Dialog Box	NSX Manager Configuration Field	How to Configure
Logical Switch	Logical Switch UUID	Select the UUID of a logical switch.

Table 12: Key Configurations to Create a Logical Switch Port in NSX Manager (*continued*)

NSX Manager Configuration Page or Dialog Box	NSX Manager Configuration Field	How to Configure
Attachment	Attachment Type	Select VTEP L2 Gateway .
Attachment	VTEP L2 Gateway Service UUID	Select the UUID of a gateway service.
Attachment	VLAN	<p>Select 0 to specify that the port handles untagged packets.</p> <p>Select 1 through 4000 to specify that the port handles tagged packets.</p> <p>NOTE: VLAN ID 4094 is reserved for a native VLAN in an OVSDB environment. Specifying this VLAN ID results in an error message. Do not specify this VLAN ID or any VLAN ID not in the accepted range.</p>

Related Documentation • [OVSDB and VXLAN Configuration Workflows for VMware NSX Environment on page 21](#)

Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections in a VMware NSX Environment (Trunk Interfaces Supporting Untagged Packets)

In a physical network, a Juniper Networks device that supports Virtual Extensible LAN (VXLAN) can function as a hardware virtual tunnel endpoint (VTEP). In this role, the Juniper Networks device encapsulates Layer 2 Ethernet frames received from software applications that run directly on a physical server in VXLAN packets. The VXLAN packets are tunneled over a Layer 3 transport network. Upon receipt of the VXLAN packets, software VTEPs in the virtual network de-encapsulate the packets and forward the packets to virtual machines (VMs).

In this VXLAN environment, you can also include VMware NSX controllers and implement the Open vSwitch Database (OVSDB) management protocol on the Juniper Networks device that functions as a hardware VTEP. The Junos OS implementation of OVSDB provides a means through which VMware NSX controllers and Juniper Networks devices can exchange MAC addresses of entities in the physical and virtual networks. This exchange of MAC addresses enables the Juniper Networks device that functions as a hardware VTEP to forward traffic to software VTEPs in the virtual network and software VTEPs in the virtual network to forward traffic to the Juniper Networks device in the physical network.

This example explains how to configure a QFX Series switch as a hardware VTEP, which serves as a Layer 2 gateway, and set up this device with an OVSDB connection to an NSX controller.

In this example, only one VXLAN is deployed. Given this scenario, the packets exchanged between an application running on a physical server and a VM in the VXLAN are untagged. As a result, the QFX Series switch automatically configures a logical trunk interface for

the connection between the physical server and the switch, as well as a native VLAN. The native VLAN enables the trunk interface to handle the untagged packets.

- [Requirements on page 39](#)
- [Overview and Topology on page 39](#)
- [Non-OVSDB and Non-VXLAN Configuration on page 42](#)
- [OVSDB and VXLAN Configuration on page 43](#)
- [Verification on page 44](#)

Requirements

This example includes the following hardware and software components:

- A physical server on which software applications directly run.
- A QFX10002 switch running Junos OS software 15.1X53-D30 or later.
- On the QFX Series switch, physical interface ge-1/0/0 provides a connection to physical server 1.
- A cluster of five NSX controllers. (In this example, you explicitly configure a connection with one NSX controller.)
- NSX Manager.
- A service node that handles the replication and forwarding of Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic within the VXLAN used in this example.
- A host that includes VMs managed by a hypervisor, which includes a software VTEP.

Before you begin:

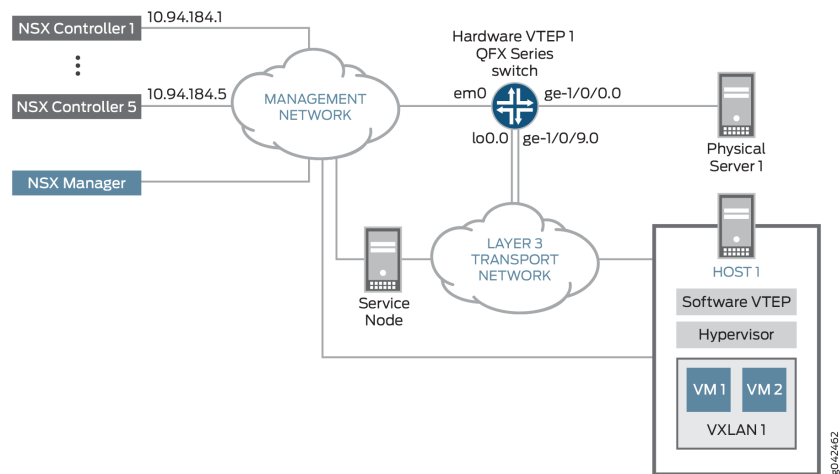
- Create an SSL private key and certificate, and install them in the `/var/db/certs` directory of the QFX Series switch. See [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers” on page 25](#).
- Using NSX Manager, specify the IP address of the service node.

For information about using NSX Manager, see the documentation that accompanies these VMware products.

Overview and Topology

[Figure 4 on page 40](#) shows a topology in which a software application running directly on physical server 1 in the physical network needs to communicate with virtual machine VM 1 in VXLAN 1 and vice versa.

Figure 4: VXLAN-OVSDB Layer 2 Gateway Topology



To establish communication between the software application on physical server 1 and VM 1 in VXLAN 1, a connection with an NSX controller is explicitly configured on the management interface of the QFX Series switch by using the Junos OS CLI.

Also, some entities in the VXLAN-OVSDB topology must be configured in both NSX Manager and on the QFX Series switch. [Table 13 on page 40](#) provides a summary of the entities that must be configured and where they must be configured.

Table 13: NSX Manager and Junos OS Entities That Must Be Configured

Entities	What Must Be Configured in NSX Manager	What Must Be Configured on a QFX Series Switch
VXLAN 1	Logical switch for VXLAN 1	VXLAN 1 NOTE: The QFX Series switch automatically configures this VXLAN.
Physical interface (ge-1/0/0) between physical server 1 and QFX Series switch	A gateway service. For gateway service type, select VTEP L2 Gateway service.	OVSDB management. Specify that interface ge-1/0/0 is managed by OVSDB.
One logical interface (ge-1/0/0.0) associated with VXLAN 1	One logical switch port for VXLAN 1. For this port, specify VLAN number 0. NOTE: A VLAN number of 0 indicates that the port must handle untagged packets.	One logical interface (ge-1/0/0.0) for VXLAN 1. NOTE: The QFX Series switch automatically configures this logical interface.
QFX Series switch (hardware VTEP 1)	Gateway	—

In NSX Manager, a logical switch for VXLAN 1 is configured. In this configuration, a VXLAN network identifier (VNI) of 100 is specified. Also, the universally unique identifier (UUID) that NSX Manager assigns to the logical switch is 28805c1d-0122-495d-85df-19abd647d772. Based on this configuration, the QFX Series

switch automatically creates the following configuration for a Junos OS-equivalent VXLAN:

```
set vlans 28805c1d-0122-495d-85df-19abd647d772 vxlan vni 100
```

Based on the gateway service and logical switch port configuration (VLAN number 0) in NSX Manager, the QFX Series switch automatically creates the following configuration for a Junos OS-equivalent interface:

```
set interfaces ge-1/0/0 flexible-vlan-tagging
set interfaces ge-1/0/0 native-vlan-id 4094
set interfaces ge-1/0/0 encapsulation extended-vlan-bridge
set interfaces ge-1/0/0 unit 0 vlan-id 4094
set vlans 28805c1d-0122-495d-85df-19abd647d772 interface ge-1/0/0.0
```

This configuration sets physical interface ge-1/0/0 as a trunk interface. It also configures a native VLAN with an ID of 4094. The configuration creates logical interface ge-1/0/0.0 and specifies that it is a member of the native VLAN. As a result, logical interface ge-1/0/0.0 handles incoming untagged packets.

The configuration also associates logical interface ge-1/0/0.0 with VXLAN 28805c1d-0122-495d-85df-19abd647d772.

[Table 14 on page 41](#) provides a summary of the VXLAN-OVSDB topology components that are configured on the QFX Series switch and the configuration settings for each component.

Table 14: Components of the Topology for Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections

Component	Setting
NSX controller	IP address: 10.94.184.1
OVSDB-managed physical interface	Interface name: ge-1/0/0 Native VLAN ID: 4094
Logical interface	<p>NOTE: The QFX Series switch automatically creates this logical interface configuration, which is based on the gateway service configuration and logical switch port configuration in NSX Manager. Therefore, no manual configuration is required.</p> <p>Interface name: ge-1/0/0.0</p> <p>Interface type: trunk</p> <p>Member of native VLAN 4094</p> <p>Associated with VXLAN 28805c1d-0122-495d-85df-19abd647d772</p>

Table 14: Components of the Topology for Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections (*continued*)

Component	Setting
OVSDB-managed VXLAN	<p>NOTE: The QFX Series switch automatically creates this VXLAN configuration, which is based on the logical switch configuration in NSX Manager. Therefore, no manual configuration is required.</p> <p>For VXLAN 1:</p> <p>VXLAN name: 28805c1d-0122-495d-85df-19abd647d772</p> <p>VNI: 100</p>
OVSDB tracing operations	<p>Filename: /var/log/ovsdb</p> <p>File size: 10 MB</p> <p>Flag: All</p>
Hardware VTEP source identifier	<p>Source interface: loopback (lo0.0)</p> <p>Source IP address: 10.17.17.17/32</p>
Handling of Layer 2 BUM traffic in VXLAN 28805c1d-0122-495d-85df-19abd647d772	<p>Service node</p> <p>NOTE: By default, one or more service nodes handle Layer 2 BUM traffic within a VXLAN; therefore, no manual configuration is required.</p>

Non-OVSDB and Non-VXLAN 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 configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-1/0/9 unit 0 family inet address 10.40.40.1/24
set routing-options static route 10.19.19.19/32 next-hop 10.40.40.2
set routing-options router-id 10.17.17.17
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-1/0/9.0
```

Step-by-Step Procedure To configure the Layer 3 network over which the packets exchanged between the physical server and VMs are tunneled:

1. Configure the Layer 3 interface.


```
[edit interfaces]
user@switch# set ge-1/0/9 unit 0 family inet address 10.40.40.1/24
```
2. Set the routing options.


```
[edit routing-options]
user@switch# set static route 10.19.19.19/32 next-hop 10.40.40.2
user@switch# set router-id 10.17.17.17
```

3. Configure the routing protocol.

```
[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface ge-1/0/9.0
```

OVSDb and VXLAN 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 configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set switch-options ovbdb-managed
set protocols ovbdb controller 10.94.184.1
set protocols ovbdb interfaces ge-1/0/0
set protocols ovbdb traceoptions file ovbdb
set protocols ovbdb traceoptions file size 10m
set protocols ovbdb traceoptions flag all
set interfaces lo0 unit 0 family inet address 10.17.17.17/32 primary
set interfaces lo0 unit 0 family inet address 10.17.17.17/32 preferred
set switch-options vtep-source-interface lo0.0
```

Step-by-Step Procedure To configure the QFX Series switch as a hardware VTEP with an OVSDb connection to an NSX controller:

1. Enable the QFX Series switch to automatically configure OVSDb-managed VXLANs and associated interfaces.

```
[edit switch-options]
user@switch# ovbdb-managed
```

2. Explicitly configure a connection with an NSX controller.

```
[edit protocols]
user@switch# set ovbdb controller 10.94.184.1
```

3. Specify that the interface between hardware VTEP 1 and physical server 1 is managed by OVSDb.

```
[edit protocols]
user@switch# set ovbdb interfaces ge-1/0/0
```

4. Set up OVSDb tracing operations.

```
[edit protocols]
user@switch# set ovbdb traceoptions file ovbdb
user@switch# set ovbdb traceoptions file size 10m
user@switch# set ovbdb traceoptions flag all
```

5. Specify an IP address for the loopback interface. This IP address serves as the source IP address in the outer header of any VXLAN-encapsulated packet.

```
[edit interfaces]
user@switch# set lo0 unit 0 family inet address 10.17.17.17/32 primary
user@switch# set lo0 unit 0 family inet address 10.17.17.17/32 preferred
```

6. Set the loopback interface as the interface that identifies hardware VTEP 1.

[edit switch-options]

user@switch# **set vtep-source-interface lo0.0**

7. In NSX Manager, configure a logical switch for VXLAN 1. See the VMware documentation that accompanies NSX Manager.
8. In NSX Manager, configure a gateway for the QFX Series switch, and configure a gateway service and logical switch port for the logical interface (ge-1/0/0.0). See [“VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints”](#) on page 35.

Verification

Confirm that the configuration is working properly:

- [Verifying the Logical Switch Configuration on page 44](#)
- [Verifying the MAC Address of VM 1 on page 44](#)
- [Verifying the NSX Controller Connection on page 45](#)
- [Verifying the OVSDB-Managed Interface on page 45](#)

Verifying the Logical Switch Configuration

Purpose Verify that the configuration of the logical switch with the UUID of 28805c1d-0122-495d-85df-19abd647d772 is present in the OVSDB schema for physical devices and that the Flags field of the **show ovssdb logical switch** output displays **Created by both**.

Action From operational mode, enter the **show ovssdb logical-switch** command.

```
user@switch> show ovssdb logical-switch
Logical switch information:
Logical Switch Name: 28805c1d-0122-495d-85df-19abd647d772
Flags: Created by both
VNI: 100
Num of Remote MAC: 1
Num of Local MAC: 0
```

Meaning The output verifies that the configuration for the logical switch is present. The **Created by both** state indicates that the logical switch was configured in NSX Manager, and that the QFX Series switch automatically created the corresponding VXLAN. In this state, the logical switch and the VXLAN are operational.

If the state of the logical switch is something other than **Created by both**, see [“Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN”](#) on page 73.

Verifying the MAC Address of VM 1

Purpose Verify that the MAC address of VM 1 is present in the OVSDB schema.

Action From operational mode, enter the **show ovssdb mac remote** command.

```
user@switch> show ovssdb mac remote
Logical Switch Name: 28805c1d-0122-495d-85df-19abd647d772
  Mac          IP          Encapsulation  Vtep
  Address      Address      Address      Address
a8:59:5e:f6:38:90  0.0.0.0      Vxlan over Ipv4  10.17.17.17
```

Meaning The output shows that the MAC address for VM 1 is present and is associated with the logical switch with the UUID of 28805c1d-0122-495d-85df-19abd647d772. Given that the MAC address is present, VM 1 is reachable through the QFX Series switch, which functions as a hardware VTEP.

Verifying the NSX Controller Connection

Purpose Verify that the connection with the NSX controller is up.

Action From operational mode, enter the **show ovssdb controller** command to verify that the controller connection state is **up**.

```
user@switch> show ovssdb controller
VTEP controller information:
Controller IP address: 10.94.184.1
Controller protocol: ssl
Controller port: 6632
Controller connection: up
Controller seconds-since-connect: 542325
Controller seconds-since-disconnect: 542346
Controller connection status: active
```

Meaning The output shows that the connection state of the NSX controller is up, in addition to other information about the controller. The **up** state of the NSX controller indicates that OVSDb is enabled on the QFX Series switch.

Verifying the OVSDb-Managed Interface

Purpose Verify that interface ge-1/0/0.0 is managed by OVSDb.

Action From operational mode, enter the **show ovssdb interface** command to verify that interface ge-1/0/0.0 is managed by OVSDb.

```
user@switch> show ovssdb interface
Interface  VLAN ID  Bridge-domain
ge-1/0/0   0        28805c1d-0122-495d-85df-19abd647d772
```

Meaning The output shows that interface ge-1/0/0 is managed by OVSDb. It also indicates that the interface is associated with VXLAN 28805c1d-0122-495d-85df-19abd647d772, which has a VLAN ID of 0.

Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections in a VMware NSX Environment (Trunk Interfaces Supporting Tagged Packets)

In a physical network, a Juniper Networks device that supports Virtual Extensible LAN (VXLAN) can function as a hardware virtual tunnel endpoint (VTEP). In this role, the Juniper Networks device encapsulates Layer 2 Ethernet frames received from software applications that run directly on a physical server in VXLAN packets. The VXLAN packets are tunneled over a Layer 3 transport network. Upon receipt of the VXLAN packets, software VTEPs in the virtual network de-encapsulate the packets and forward the packets to virtual machines (VMs).

In this VXLAN environment, you can also include VMware NSX controllers and implement the Open vSwitch Database (OVSDB) management protocol on the Juniper Networks device that functions as a hardware VTEP. The Junos OS implementation of OVSDB provides a means through which VMware NSX controllers and Juniper Networks devices can exchange MAC addresses of entities in the physical and virtual networks. This exchange of MAC addresses enables the Juniper Networks device that functions as a hardware VTEP to forward traffic to software VTEPs in the virtual network and software VTEPs in the virtual network to forward traffic to the Juniper Networks device in the physical network.

This example explains how to configure a Juniper Networks device that supports VXLAN as a hardware VTEP. (The VTEP serves as a Layer 2 gateway.) This example also explains how to configure this device with an OVSDB connection to an NSX controller.

In this example, an application running directly on a physical server needs to communicate with a VM in a VXLAN, while another application on the physical server needs to communicate with VMs in another VXLAN. Therefore, the packets exchanged between the applications running on the physical server and the respective VMs with which they must communicate are tagged. As a result, a trunk interface is used for the connection between the physical server and the Juniper Networks device.

- [Requirements on page 46](#)
- [Overview and Topology on page 47](#)
- [Non-OVSDB and Non-VXLAN Configuration on page 50](#)
- [OVSDB and VXLAN Configuration on page 51](#)
- [Verification on page 52](#)

Requirements

This example includes the following hardware and software components:

- A physical server on which software applications directly run.
- A Juniper Networks switch that supports VXLAN and OVSDB. This switch can be a QFX10002 switch running Junos OS Release 15.1X53-D10 and later.
- On the Juniper Networks switch, physical interface ge-1/0/0 provides a connection to physical server 1.

- A cluster of five NSX controllers. (In this example, you explicitly configure a connection with one NSX controller.)
- NSX Manager.
- A service node that handles the replication and forwarding of Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic within the VXLANs.
- Two hosts that include VMs. Each host is managed by a hypervisor, and each hypervisor includes a software VTEP.

Before you begin the configuration, you must perform the following tasks:

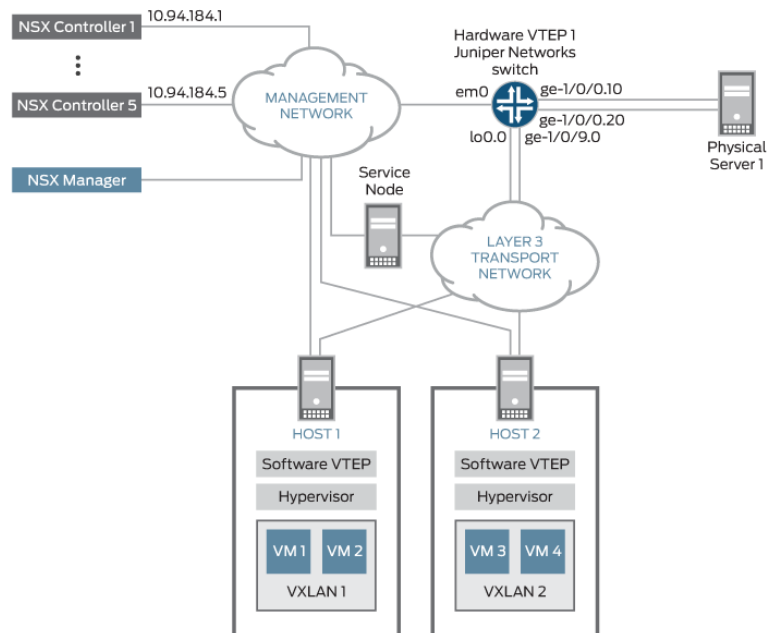
- Create an SSL private key and certificate, and install them in the `/var/db/certs` directory of the Juniper Networks switch. See [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers”](#) on page 25.
- Using NSX Manager, specify the IP address of the service node.

For information about using NSX Manager, see the documentation that accompanies NSX Manager.

Overview and Topology

Figure 4 on page 40 shows a topology in which a software application running directly on physical server 1 in the physical network needs to communicate with virtual machine VM 1 in VXLAN 1 and vice versa, and another software application on physical server 1 needs to communicate with virtual machines VM 3 and VM 4 in VXLAN 2 and vice versa.

Figure 5: VXLAN/OVSDB Layer 2 Gateway Topology



85043237

To establish communication between the software applications on physical server 1 and the VMs in VXLANs 1 and 2, some entities in the VXLAN-OVSDB topology must be configured in both NSX Manager and on the Juniper Networks switch. [Table 15 on page 48](#) provides a summary of the entities that must be configured and where they must be configured.



NOTE: The term used for an entity configured in NSX Manager can differ from the term used for essentially the same entity configured on the Junos Network switch. To prevent confusion, [Table 15 on page 48](#) shows the NSX Manager and Junos OS entities side-by-side.

Table 15: NSX Manager and Junos OS Entities That Must Be Configured

Entities	What Must Be Configured In NSX Manager	What Must Be Configured on Juniper Networks Switch
VXLAN 1	Logical switch for VXLAN 1	VXLAN 1
VXLAN 2	Logical switch for VXLAN 2	VXLAN 2
		NOTE: The Juniper Networks switch dynamically configures these VXLANs.
Interface (ge-1/0/0) between physical server 1 and Juniper Networks switch	A gateway service. For gateway service type, select VTEP L2 gateway service.	OVSDB management. Specify that interface ge-1/0/0 is managed by OVSDB.
One logical interface associated with VXLAN 1	One logical switch port for VXLAN 1. For this port, specify VLAN number 10.	One logical interface (ge-1/0/0.10) for VXLAN 1
One logical interface associated with VXLAN 2	One logical switch port for VXLAN 2. For this port, specify VLAN number 20.	One logical interface (ge-1/0/0.20) for VXLAN 1
	NOTE: A VLAN number from 1 through 4000 indicates that the port is a trunk port.	NOTE: The Juniper Networks switch dynamically configures these logical interfaces.
Juniper Networks switch (hardware VTEP 1)	Gateway	—

Based on the configuration of the entities in NSX Manager as described in [Table 15 on page 48](#), the Juniper Networks switch dynamically creates VXLANs 1 and 2 and their associated logical interfaces. [Table 16 on page 49](#) provides the relevant NSX Manager configuration and the resulting VXLANs and associated logical interfaces that the Juniper Networks switch dynamically configures.

Table 16: NSX Manager Configurations and Dynamic Configurations by Juniper Networks Switch

NSX Manager Configuration: Logical Switch and Logical Switch Port	VXLANs and Associated Logical Interfaces Dynamically Configured By Juniper Networks Switch
Logical switch configuration: UUID: 28805c1d-0122-495d-85df-19abd647d772 VNI: 100 Logical switch port configuration: VLAN ID: 10	For VXLAN 1: set vlans 28805c1d-0122-495d-85df-19abd647d772 vxlan vni 100 For associated logical interface ge-1/0/0.10: set interfaces ge-1/0/0 flexible-vlan-tagging set interfaces ge-1/0/0 encapsulation extended-vlan-bridge set interfaces ge-1/0/0 unit 10 vlan-id 10 set vlans 28805c1d-0122-495d-85df-19abd647d772 interfaces ge-1/0/0.10
Logical switch configuration: UUID: 9acc24b3-7b0a-4c2e-b572-3370c3e1acff VNI: 200 Logical switch port configuration: VLAN ID: 20	For VXLAN 2: set vlans 9acc24b3-7b0a-4c2e-b572-3370c3e1acff vxlan vni 200 For associated logical interface ge-1/0/0.20: set interfaces ge-1/0/0 flexible-vlan-tagging set interfaces ge-1/0/0 encapsulation extended-vlan-bridge set interfaces ge-1/0/0 unit 20 vlan-id 20 set vlans 9acc24b3-7b0a-4c2e-b572-3370c3e1acff interfaces ge-1/0/0.20

For VXLANs 1 and 2, the Juniper Networks switch uses the UUIDs and VNI values that were provided for the corresponding logical switches.

In the logical switch port configurations in NSX Manager, VLAN ID values 10 and 20 and logical switch mappings are specified. As a result, the Juniper Networks switch creates logical interfaces ge-1/0/0.10 and ge-1/0/0.20, respectively. Both of these logical interfaces function as trunk interfaces. The Juniper Networks switch also maps the logical interfaces ge-1/0/0.10 and ge-1/0/0.20 to their respective VXLANs.

Based on the configurations generated by the Juniper Networks switch, the interface ge-1/0/0.10 accepts packets with a VLAN tag of 10 from VXLAN 1, and interface ge-1/0/0.20 accepts packets with a VLAN tag of 20 from VXLAN 2. On receiving packets from VXLAN 1, a VLAN tag of 100 is added to the packets, and a VLAN tag of 200 is added to packets from VXLAN 2. These tags are added to the respective packet streams to map the VLAN ID in a particular VXLAN to the corresponding VNI.

[Table 14 on page 41](#) provides a summary of the components that are configured on the Juniper Networks switch. Unless noted, all configurations are performed manually in the Junos OS CLI.

Table 17: Components for Two VXLAN Topologies Configured on a Juniper Networks Switch that Functions as a Hardware VTEP

Components	Settings
NSX controller	IP address: 10.94.184.1

Table 17: Components for Two VXLAN Topologies Configured on a Juniper Networks Switch that Functions as a Hardware VTEP (*continued*)

Components	Settings
OVSDB-managed interface	Interface name: ge-1/0/0
VXLAN 1 and associated logical interface	<p>NOTE: The Juniper Networks switch dynamically configures the VXLAN and associated logical interface, which are based on the logical switch and logical switch port configurations in NSX Manager. Therefore, no manual configuration is required.</p> <p>VXLAN name: 28805c1d-0122-495d-85df-19abd647d772</p> <p>VNI: 100</p> <p>Logical interface name: ge-1/0/0.10</p> <p>VLAN ID: 10</p> <p>Interface type: trunk</p>
VXLAN 2 and associated logical interface	<p>NOTE: The Juniper Networks switch dynamically configures the VXLAN and associated interface, which are based on the logical switch and logical switch port configurations in NSX Manager. Therefore, no manual configuration is required.</p> <p>VXLAN name: VXLAN 9acc24b3-7b0a-4c2e-b572-3370c3e1acff</p> <p>VNI: 200</p> <p>Logical interface name: ge-1/0/0.20</p> <p>VLAN ID: 20</p> <p>Interface type: trunk</p>
OVSDB tracing operations	<p>Filename: /var/log/ovsdb</p> <p>File size: 10 MB</p> <p>Flag: All</p>
Hardware VTEP source identifier	<p>Source interface: loopback (lo0.0)</p> <p>Source IP address: 10.17.17.17/32</p>
Handling of Layer 2 BUM traffic within VXLAN 28805c1d-0122-495d-85df-19abd647d772 and within VXLAN 9acc24b3-7b0a-4c2e-b572-3370c3e1acff	<p>Service node</p> <p>NOTE: By default, one or more service nodes handle Layer 2 BUM traffic in a VXLAN; therefore, no configuration is required.</p>

Non-OVSDB and Non-VXLAN 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 configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```

set interfaces ge-1/0/9 unit 0 family inet address 10.40.40.1/24
set routing-options static route 10.19.19.19/32 next-hop 10.40.40.2
set routing-options router-id 10.17.17.17
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-1/0/9.0

```

Step-by-Step Procedure To configure the Layer 3 network over which the packets exchanged between physical server 1 and VM1 are tunneled:

1. Configure the Layer 3 interface.

```

[edit interfaces]
user@switch# set ge-1/0/9 unit 0 family inet address 10.40.40.1/24

```

2. Set the routing options.

```

[edit routing-options]
user@switch# set static route 10.19.19.19/32 next-hop 10.40.40.2
user@switch# set router-id 10.17.17.17

```

3. Configure the routing protocol.

```

[edit protocols]
user@switch# set ospf area 0.0.0.0 interface lo0.0
user@switch# set ospf area 0.0.0.0 interface ge-1/0/9.0

```

OVSDb and VXLAN 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 configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```

set switch-options ovssdb-managed
set protocols ovssdb controller 10.94.184.1
set protocols ovssdb interfaces ge-1/0/0
set protocols ovssdb traceoptions file ovssdb
set protocols ovssdb traceoptions file size 10m
set protocols ovssdb traceoptions flag all
set interfaces lo0 unit 0 family inet address 10.17.17.17/32 primary
set interfaces lo0 unit 0 family inet address 10.17.17.17/32 preferred
set switch-options vtep-source-interface lo0.0

```

Step-by-Step Procedure To configure the Juniper Networks switch as hardware VTEP 1 and with an OVSDb connection to an NSX controller:

1. Enable the Juniper Networks switch to dynamically configure OVSDb-managed VXLANs and associated interfaces.

```

[edit switch-options]
user@switch# set ovssdb-managed

```

2. Explicitly configure a connection with an NSX controller.

```

[edit protocols]
user@switch# set ovssdb controller 10.94.184.1

```

3. Specify that interface ge-1/0/0 is managed by OVSDB.

```
[edit protocols]
user@switch# set ovsdb interfaces ge-1/0/0
```

4. Set up OVSDB tracing operations.

```
[edit protocols]
user@switch# set ovsdb traceoptions file ovsdb
user@switch# set ovsdb traceoptions file size 10m
user@switch# set ovsdb traceoptions flag all
```

5. Specify an IP address for the loopback interface. This IP address serves as the source IP address in the outer header of any VXLAN-encapsulated packets.

```
[edit interfaces]
user@switch# set lo0 unit 0 family inet address 10.17.17.17/32 primary
user@switch# set lo0 unit 0 family inet address 10.17.17.17/32 preferred
```

6. Set the loopback interface as the interface that identifies hardware VTEP 1.

```
[edit switch-options]
user@switch# set vtep-source-interface lo0.0
```

7. In NSX Manager, configure a logical switch for VXLAN 1 and a logical switch for VXLAN 2. See the documentation that accompanies NSX Manager.
8. In NSX Manager, configure a gateway for the Juniper Networks switch, a gateway service for OVSDB-managed interface ge-1/0/0, and a logical switch port for logical interface ge-1/0/0.10, which is associated with VXLAN 1, and a logical switch port for logical interface ge-1/0/0.20, which is associated with VXLAN 2.

See “[VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints](#)” on page 35.

Verification

Confirm that the configuration is working properly.

- [Verifying the Logical Switch Configuration on page 52](#)
- [Verifying the MAC Addresses of VM 1, VM 3, and VM 4 on page 53](#)
- [Verifying the NSX Controller Connection on page 53](#)
- [Verifying the OVSDB-Managed Interface on page 54](#)

Verifying the Logical Switch Configuration

Purpose Verify that the configuration of logical switches with the UUIDs of 28805c1d-0122-495d-85df-19abd647d772 and 9acc24b3-7b0a-4c2e-b572-3370c3e1acff are present in the OVSDB schema for physical devices and that the Flags field of the **show ovsdb logical-switch** output is Created by both.

Action From operational mode, enter the **show ovssdb logical-switch** command.

```
user@switch> show ovssdb logical-switch
Logical switch information:
Logical Switch Name: 28805c1d-0122-495d-85df-19abd647d772
Flags: Created by both
VNI: 100
Num of Remote MAC: 1
Num of Local MAC: 0
Logical Switch Name: 9acc24b3-7b0a-4c2e-b572-3370c3e1acff
Flags: Created by both
VNI: 200
Num of Remote MAC: 2
Num of Local MAC: 0
```

Meaning The output verifies that the configuration for the logical switches is present. The **Created by both** state indicates that the logical switches were configured in NSX Manager, and that the Juniper Networks switch dynamically configured the corresponding VXLANs. In this state, the logical switches and VXLANs are operational.

If the state of the logical switches is something other than **Created by both**, see [“Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN” on page 73](#).

Verifying the MAC Addresses of VM 1, VM 3, and VM 4

Purpose Verify that the MAC addresses of VM1, VM3, and VM 4 are present in the OVSDB schema.

Action From operational mode, enter the **show ovssdb mac remote** command.

```
user@switch> show ovssdb mac remote
Logical Switch Name: 28805c1d-0122-495d-85df-19abd647d772
  Mac      IP      Encapsulation      Vtep
  Address  Address
  a8:59:5e:f6:38:90    0.0.0.0      Vxlan over Ipv4    10.17.17.17
Logical Switch Name: 9acc24b3-7b0a-4c2e-b572-3370c3e1acff
  Mac      IP      Encapsulation      Vtep
  Address  Address
  00:23:9c:5e:a7:f0    0.0.0.0      Vxlan over Ipv4    10.17.17.17
  00:23:9c:5e:a7:f0    0.0.0.0      Vxlan over Ipv4    10.17.17.17
```

Meaning The output shows that the MAC addresses for VM 1, VM 3, and VM 4 are present and are associated with their respective logical switches. Given that the MAC addresses are present, VM1, VM 3, and VM 4 are reachable through the Juniper Networks switch, which functions as a hardware VTEP.

Verifying the NSX Controller Connection

Purpose Verify that the connection with the NSX controller is up.

Action From operational mode, enter the **show ovssdb controller** command to verify that the controller connection state is **up**.

```
user@switch> show ovssdb controller
VTEP controller information:
Controller IP address: 10.94.184.1
Controller protocol: ssl
Controller port: 6632
Controller connection: up
Controller seconds-since-connect: 542325
Controller seconds-since-disconnect: 542346
Controller connection status: active
```

Meaning The output shows that the connection state of the NSX controller is **up**, in addition to other information about the controller. By virtue of this connection being up, OVSDB is enabled on the Juniper Networks switch.

Verifying the OVSDB-Managed Interface

Purpose Verify that interface ge-1/0/0 is managed by OVSDB.

Action From operational mode, enter the **show ovssdb interface** command, and verify that logical interfaces ge-1/0/0.10 and ge-1/0/0.20 are managed by OVSDB.

```
user@switch> show ovssdb interface
Interface  VLAN  ID  Bridge-domain
ge-1/0/0   10    28805c1d-0122-495d-85df-19abd647d772
ge-1/0/0   20    9acc24b3-7b0a-4c2e-b572-3370c3e1acff
```

Meaning The output shows that logical interfaces **ge-1/0/0.10** and **ge-1/0/0.20** are managed by OVSDB. It also indicates that interface **ge-1/0/0.10** is associated with VXLAN **28805c1d-0122-495d-85df-19abd647d772** and interface **ge-1/0/0.20** is associated with VXLAN **9acc24b3-7b0a-4c2e-b572-3370c3e1acff**.

Verifying That a Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN Are Working Properly

Purpose Verify the following:

- A logical switch, which is configured in an NSX environment, or a virtual network, which is configured in a Contrail environment, is learning MAC addresses in their respective environments.
- The corresponding OVSDB-managed Virtual Extensible LAN (VXLAN), which is configured on a Juniper Networks device, is learning MAC addresses in the Junos OS environment.
- The logical switch or virtual network and OVSDB-managed VXLAN are exchanging the MAC addresses learned in their respective environments so that virtual and physical servers can communicate.

Action To verify that a logical switch or virtual network and its corresponding OVSDB-managed VXLAN are learning and exchanging MAC addresses in their respective environments, enter the **show ovssdb logical-switch** operational mode command.

```
user@device> show ovssdb logical-switch
Logical switch information:
Logical Switch Name: 28805c1d-0122-495d-85df-19abd647d772
Flags: Created by both
VNI: 100
Num of Remote MAC: 1
Num of Local MAC: 0
```



NOTE: In the Open vSwitch Database (OVSDB) schema for physical devices, the logical switch table stores information about the Layer 2 broadcast domain that you configured in a VMware NSX or Contrail environment. In the NSX environment, the Layer 2 broadcast domain is known as a *logical switch*, while in the Contrail environment, the domain is known as a *virtual network*.

In the context of the **show ovssdb logical-switch** command, the term *logical switch* refers to the logical switch or virtual network that was configured in the NSX or Contrail environments, respectively, and the corresponding configuration that was pushed to the OVSDB schema.

Meaning The output in the Flags field (**Created by both**) indicates that the logical switch or virtual network and its corresponding OVSDB-managed VXLAN are both properly configured. In this state, the logical switch or virtual network and the VXLAN are learning and exchanging MAC addresses in their respective environments.

If the output in the Flags field displays a state other than **Created by both**, see [“Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN” on page 73](#).

Related Documentation

- [show ovssdb logical-switch on page 106](#)

CHAPTER 3

Configuring VXLANs Without an SDN Controller

- [VXLAN Constraints on QFX Series Switches on page 57](#)
- [Manually Configuring VXLANs on QFX Series Switches on page 60](#)
- [Examples: Manually Configuring VXLANs on QFX Series Switches on page 61](#)
- [Verifying That a Local VXLAN VTEP Is Configured Correctly on page 69](#)
- [Verifying MAC Learning from a Remote VTEP on page 69](#)

VXLAN Constraints on QFX Series Switches

When configuring VXLANs on QFX Series switches, be aware of the constraints described in the following sections. In these sections, “Layer 3 side” refers to a network-facing interface that performs VXLAN encapsulation and de-encapsulation, and “Layer 2 side” refers to a server-facing interface that is a member of a VLAN that is mapped to a VXLAN.

- [VXLAN Constraints on QFX5100 and QFX5110 Switches on page 57](#)
- [VXLAN Constraints on QFX10000 Switches on page 59](#)

VXLAN Constraints on QFX5100 and QFX5110 Switches

- (QFX5100 switches only) You can use VXLANs on a Virtual Chassis or Virtual Chassis Fabric if all of the members are supported QFX5100 switches. You cannot use VXLANs if any of the members is not a supported QFX5100 switch.
- VXLAN configuration is supported only in the default routing instance.
- These QFX Series switches cannot route traffic between different VXLANs.
- A physical interface cannot be a member of a VLAN and a VXLAN. That is, an interface that performs VXLAN encapsulation and de-encapsulation cannot also be a member of a VLAN. For example, if a VLAN that is mapped to a VXLAN is a member of trunk port xe-0/0/0, any other VLAN that is a member of xe-0/0/0 must also be assigned to a VXLAN.
- Multichassis link aggregation groups (MC-LAGs) are not supported with VXLAN.



NOTE: In an EVPN-VXLAN environment, multihoming active-active mode is used instead of MC-LAG for redundant connectivity between hosts and leaf devices.

- IP fragmentation and defragmentation are not supported on the Layer 3 side.
- The following features are not supported on the Layer 2 side:
 - STP (any variant).
 - IGMP snooping.
 - Storm control is supported on the Layer 2 side with the following constraints:
 - The ability to shut down a Layer 2 interface or temporarily disable the interface when the storm control level is exceeded is not supported.
 - The storm control feature enables you to control the amount of broadcast, unknown unicast, and multicast (BUM) traffic received on specified Layer 2 interfaces that are associated with VXLANs. Conversely, if a QFX Series switch that functions as a VTEP is the source of a broadcast storm, these switches cannot control the amount of BUM traffic sent on its Layer 2 interfaces.
- Access port security features are not supported with VXLAN. For example, the following features are not supported:
 - DHCP snooping.
 - Dynamic ARP inspection.
 - MAC limiting and MAC move limiting.



NOTE: An exception to this constraint is that MAC limiting is supported on OVSDB-managed interfaces configured on QFX5100 switches in an OVSDB-VXLAN environment with Contrail controllers. For more information, see *Features Supported on OVSDB-Managed Interfaces*.

- Ingress node replication is not supported in the following cases:
 - When PIM is used for the control plane.
 - When an SDN controller is used for the control plane.

Ingress node replication is supported for EVPN-VXLAN.
- PIM-BIDIR and PIM-SSM are not supported with VXLANs.
- Class of service (CoS) features are not supported with VXLANs.



NOTE: An exception to CoS constraint is that CoS features are supported on OVSDB-managed interfaces in an OVSDB-VXLAN environment with Contrail controllers. For more information, see *Features Supported on OVSDB-Managed Interfaces*.

- If you configure a port-mirroring instance to mirror traffic exiting from an interface that performs VXLAN encapsulation, the source and destination MAC addresses of the mirrored packets are invalid. The original VXLAN traffic is not affected.

VXLAN Constraints on QFX10000 Switches

- Multichassis link aggregation groups (MC-LAGs) are not supported with VXLAN.



NOTE: In an EVPN-VXLAN environment, multihoming active-active mode is used instead of MC-LAG for redundant connectivity between hosts and leaf devices.

- IP fragmentation is not supported on the Layer 3 side.
- The following features are not supported on the Layer 2 side:
 - STP (any variant).
 - IGMP snooping.
- The storm control feature enables you to control the amount of broadcast, unknown unicast, and multicast (BUM) traffic received on specified Layer 2 interfaces that are associated with VXLANs. Conversely, if a Juniper Networks QFX Series switch that functions as a VTEP is the source of a broadcast storm, these switches cannot control the amount of BUM traffic sent on its Layer 2 interfaces
- Access port security features are not supported with VXLAN. For example, the following features are not supported:
 - DHCP snooping.
 - Dynamic ARP inspection.
 - MAC limiting and MAC move limiting.
- Ingress node replication is not supported when an SDN controller is used for the control plane. Ingress node replication is supported for EVPN-VXLAN.
- CoS features are not supported with VXLANs.



NOTE: An exception to the CoS constraint is that CoS features are supported on OVSDB-managed interfaces in an OVSDB-VXLAN environment with Contrail controllers. For more information, see *Features Supported on OVSDB-Managed Interfaces*.

- Related Documentation**
- [Understanding VXLANs on page 6](#)
 - [Examples: Manually Configuring VXLANs on QFX Series Switches on page 61](#)
 - [Manually Configuring VXLANs on QFX Series Switches on page 60](#)

Manually Configuring VXLANs on QFX Series Switches

You can configure QFX5100 and QFX5110 switches to act as a VTEP. (If the switch is acting as a transit Layer 3 switch for downstream VTEPs, you do not need to perform the steps in this topic as no special configuration is needed.)

- [Configuring a Source IP Address on page 60](#)
- [Configuring PIM for VXLANs on page 60](#)
- [Configuring VXLANs on page 61](#)

Configuring a Source IP Address

On a switch that will act as a VTEP, you must configure an IP address that will be used as the source address in the outer IP header of the VXLAN packet. This is the VXLAN tunnel source address.

1. Create a reachable IPv4 address on the loopback interface.

```
[edit]
user@switch# set interfaces lo0.0 unit 0 family inet address ip-address
```

2. Configure the address to be used as the tunnel source address.

```
[edit]
user@switch# set switch-options vtep-interface-source lo0.0
```

Configuring PIM for VXLANs

If you are not using an SDN controller to create a VXLAN control plane, you must enable PIM on the switch so that the VTEP can use multicast groups to establish reachability with other VTEPs and to forward BUM traffic.

1. Enable PIM on the interface that connects to the Layer 3 network. This is the interface that performs the VXLAN encapsulation and de-encapsulation.

```
[edit]
user@switch# set protocols pim interface interface-name
```

2. Configure the address of a PIM rendezvous point.

```
[edit]
user@switch# set protocols pim rp static address ip-address
```

Configuring VXLANs

You configure VXLANs under the **vlan** stanza (which is why a QFX5100 switch supports 4000 VXLANs). You must also configure the server-facing interfaces to be VLAN members.

1. Create a VLAN to VXLAN mapping and assign a multicast group address to the VXLAN. All members of a VXLAN must use the same multicast group address.

```
[edit]
user@switch# set vlans name vlan-id ID vxlan vni ID multicast-group multicast-group-address
```

2. (Optional) Configure the switch to retain the original VLAN tag (in the inner Ethernet packet) after VXLAN encapsulation. By default, the original tag is dropped when the packet is encapsulated.

```
[edit]
user@switch# set vlans name vxlan encapsulate-inner-vlan
```

3. (Optional) Configure the switch to de-encapsulate and accept original VLAN tags in VXLAN packets. By default, the original tag is dropped when the packet is encapsulated.

```
[edit]
user@switch# set protocols l2-learning decapsulate-accept-inner-vlan
```

4. Configure server-facing interfaces to support multiple VLANs.

```
[edit]
user@switch# set interfaces interface unit unit family ethernet-switching interface-mode trunk
```

```
[edit]
user@switch# set interfaces interface unit unit family ethernet-switching vlan members all
```

You must create a VLAN to VXLAN mapping for each VLAN that will need Layer 2 connectivity over the Layer 3 network.

- Related Documentation**
- [Understanding VXLANs on page 6](#)
 - [Examples: Manually Configuring VXLANs on QFX Series Switches on page 61](#)

Examples: Manually Configuring VXLANs on QFX Series Switches

The following examples show use cases for manually configuring VXLANs on QFX5100 and QFX5110 switches.

- [Example: Configuring a VXLAN Transit Switch on page 61](#)
- [Example: Configuring a VXLAN Layer 2 Gateway on page 63](#)

Example: Configuring a VXLAN Transit Switch

If a QFX Series switch acts as a transit switch for downstream devices acting as VTEPs, you do not need to configure any VXLAN information on the QFX Series switch. You do

need to configure PIM on the switch so that it can form the multicast tree required so that the VTEPs can establish reachability with each other.

- [Requirements on page 62](#)
- [Overview on page 62](#)
- [Configuring PIM on the Transit Switches on page 63](#)

Requirements

This example uses the following hardware and software components:

- Two QFX5100 switches
- Junos OS Release 14.1X53-D10

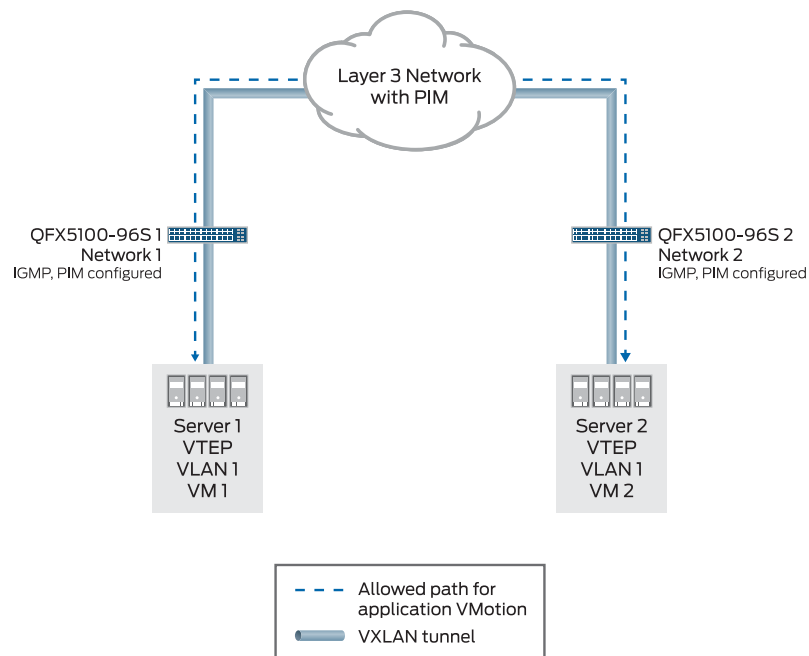
Overview

This example shows a simple use case in which QFX Series switches are connected to downstream servers acting as VTEPs. The QFX Series switches need to forward VXLAN packets between VM1 on Server 1 and VM 2 on Server 2. Because this configuration allows Layer 2 connectivity between the VMs through the VXLAN tunnels, applications can VMotion between the VMs.

Topology

[Figure 6 on page 62](#) shows QFX 5100 switches configured to forward VXLAN packets for downstream VTEPs.

Figure 6: QFX5100 Acting as a VXLAN Transit Switch



8043109

Configuring PIM on the Transit Switches

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.

```
set protocols pim interface all
set protocols pim rp static address ip-address
```

Step-by-Step Procedure If you are not using an SDN controller to create a VXLAN control plane, you must enable PIM on each switch so that the VTEP can use multicast groups to advertise its existence and to learn about other VTEPs. (Configuring PIM automatically enables IGMP.) You do not need to perform any VXLAN-specific configuration. Note that you also do not need to configure VLAN 1 on either switch.

1. Enable PIM.

```
[edit]
user@switch# set protocols pim interface all
```
2. Configure the address of a PIM rendezvous point.

```
[edit]
user@switch# set protocols pim rp static address ip-address
```

Example: Configuring a VXLAN Layer 2 Gateway

If a QFX Series switch is connected to a downstream server that hosts a VM that needs Layer 2 connectivity with another VM that is reachable only through a Layer 3 network, you must configure the switch to act as a VTEP—that is, a Layer 2 gateway for downstream Layer 2 devices. You also need to configure PIM on the switch so that it can form the multicast tree required for reachability with other VTEPs and to allow BUM traffic to be forwarded between the VTEPs.

- [Requirements on page 63](#)
- [Overview on page 63](#)
- [Configuring the Switches on page 64](#)
- [Verification on page 67](#)

Requirements

This example uses the following hardware and software components:

- Two QFX5100 switches
- Junos OS Release 14.1X53-D10

Overview

This example shows a use case in which QFX Series switches act as VTEPs that allow Layer 2 connectivity between VM 1 on Server 1 and VM 2 on Server 2 so that VMotion can occur between the VMs. The servers in this example can be in the same or different data centers—the only constraint is that there must be Layer 3 connectivity between the QFX

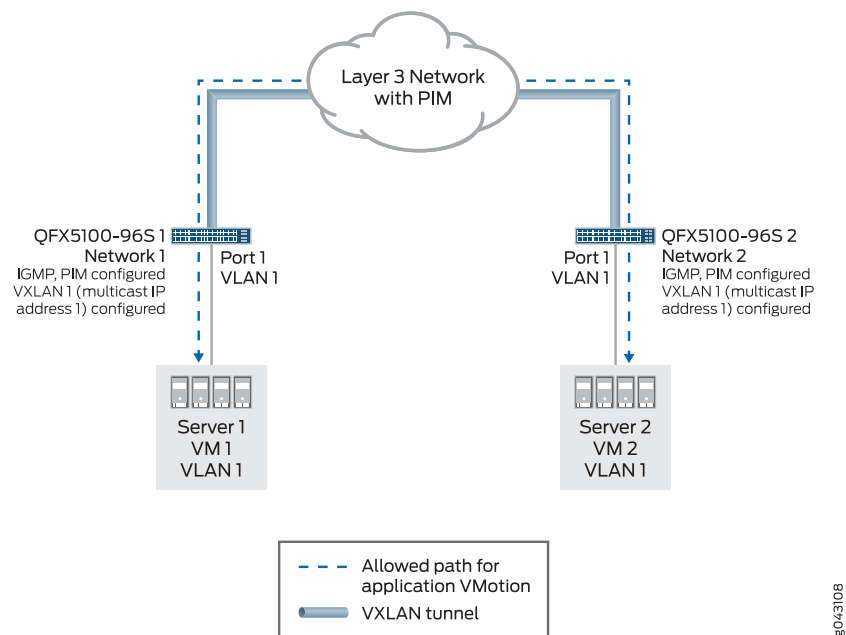
Series switches. This allows your network to be very agile in response to demand for server usage or changes in bandwidth requirements.

Note that because the same VLAN exists in both Layer 2 domains and both switches encapsulate the VLAN traffic into the same VXLAN, you do not need a gateway for the VXLAN traffic in the Layer 3 network. The Layer 3 VXLAN packets are routed normally and no de-encapsulation or re-encapsulation is required.

Topology

Figure 7 on page 64 shows QFX5100 switches configured to act as VTEPs.

Figure 7: QFX5100 Acting as a VTEP



Configuring the Switches

CLI Quick Configuration

To quickly configure the QFX5100-96S 1 in 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.

```
set interfaces lo0 unit 0 family inet address 10.1.1.1
set switch-options vtep-source-interface lo0.0
set protocols pim interface lo0.0
set protocols pim interface xe-0/0/0.0
set protocols pim rp static address 10.2.2.2
set vlans VLAN1 vlan-id 100 vxlan vni 100 multicast-group 224.2.2.2
set vlans VLAN1 vxlan encapsulate-inner-vlan
set vlans VLAN1 vxlan unreachable-vtep-aging-timer 600
set protocols l2-learning decapsulate-accept-inner-vlan
set interfaces xe-0/0/0 unit 0 family inet address 10.2.2.100/24
```



```
set interfaces xe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members all
```

The configuration for QFX5100-96S 2 is identical except for changes to a few of the addresses:

```
set interfaces lo0 unit 0 family inet address 10.1.1.2
set switch-options vtep-source-interface lo0.0
set protocols pim interface lo0.0
set protocols pim interface xe-0/0/0.0
set protocols pim rp static address 10.2.2.2
set vlans VLAN1 vlan-id 100 vxlan vni 100 multicast-group 224.2.2.2
set vlans VLAN1 vxlan encapsulate-inner-vlan
set vlans VLAN1 vxlan unreachable-vtep-aging-timer 600
set protocols l2-learning decapsulate-accept-inner-vlan
set interfaces xe-0/0/0 unit 0 family inet address 10.2.2.200/24
set interfaces xe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members all
```



NOTE: You must configure the same multicast group address for VLAN1 on both switches.

Step-by-Step Procedure

Perform the following procedure on both switches to set up the example configuration.

1. Create a reachable IPv4 address on the loopback interface.

```
[edit]
user@switch# set interfaces lo0 unit 0 family inet address 10.1.1.1
For switch QFX5100-96S 2, use address 10.1.1.2.
```

2. Configure the loopback interface—and therefore, its associated address—to be used as the tunnel source address.

```
[edit]
user@switch# set switch-options vtep-source-interface lo0.0
```

3. Enable PIM on the loopback interface.

```
[edit]
user@switch# set protocols pim interface lo0.0
```

4. Enable PIM on the interface that connects to the Layer 3 network.

```
[edit]
user@switch# set protocols pim interface xe-0/0/0.0
```

5. Configure the address of a PIM rendezvous point.

```
[edit]
user@switch# set protocols pim rp static address 10.2.2.2
```

6. Create a VLAN, map it to a VXLAN, and assign a multicast group address to the VXLAN. All members of a VXLAN must use the same multicast group address.

```
[edit]
user@switch# set vlans VLAN1 vlan-id 100 vxlan vni 100 multicast-group 224.2.2.2
```

In this example, the **vlan-id** and **vni** are both set to **100**. This is done only for simplicity and clarity. You do not need to set the **vlan-id** and **vni** to the same value.

7. (Optional) Configure the switch to retain the original VLAN tag (in the inner Ethernet packet) after VXLAN encapsulation. By default, the original tag is dropped when the packet is encapsulated.

```
[edit]
```

```
user@switch# set vlans VLAN1 vxlan encapsulate-inner-vlan
```

8. (Optional) Configure the system to age out the address for the remote VTEP (the other QFX5100 switch) if all the MAC addresses learned from that VTEP age out. The address for the remote VTEP expires the configured number of seconds after the last learned MAC address expires.

```
[edit]
```

```
user@switch# set vlans VLAN1 vxlan unreachable-vtep-aging-timer 600
```

(Optional) Configure the switch to de-encapsulate and accept original VLAN tags in VXLAN packets. By default, a preserved VLAN tag is dropped when the packet is de-encapsulated.

```
[edit]
```

```
user@switch# set protocols l2-learning decapsulate-accept-inner-vlan
```

9. Configure the interface that connects to the Layer 3 network.

```
[edit]
```

```
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.2.2.100/24
```

For switch QFX5100-96S 2, use address 10.2.2.200.

10. Configure the server-facing interface to support multiple VLANs.

```
[edit]
```

```
user@switch# set interfaces xe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
```

```
[edit]
```

```
user@switch# set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members all
```



NOTE: Because this example shows only one VLAN, this step is not required for the example. In a real-world configuration, however, it would be required in order to support multiple VMs connected to multiple VLANs. In this case you would also need to configure additional VLAN to VXLAN mappings.

Results

From configuration mode, confirm your configuration by entering the following commands on QFX5100-96S 1. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@switch# show switch-options
```

```
  vtep-source-interface lo0.0;
```

```
user@switch# show vlans
```

```
VLAN1 {
  vlan-id 100;
  vxlan {
    vni 100;
```

```

        multicast-group 224.2.2.2;
        encapsulate-inner-vlan;
    }
}
user@switch# show interfaces
xe-0/0/0 {
  unit 0 {
    family inet {
      address 10.2.2.100/24;
    }
  }
}
xe-0/0/1 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members all;
      }
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.1.1.1/32;
    }
  }
}
user@switch# show protocols pim
rp {
  static {
    address 10.2.2.2;
  }
}
interface xe-0/0/0.0

```

Verification

Confirm that the configuration is working properly.

- [Verifying VXLAN Reachability on page 67](#)
- [Verifying That the Local VTEP Is Configured Correctly on page 68](#)
- [Verifying MAC Learning from the Remote VTEP on page 68](#)
- [Monitor the Remote Interface on page 68](#)

Verifying VXLAN Reachability

Purpose On QFX5100-96S 1, verify that there is connectivity with the remote VTEP (QFX5100-96S 2).

Action user@switch> show ethernet-switching vxlan-tunnel-end-point remote

Logical System Name	Id	SVTEP-IP	IFL	L3-Idx
<default>	0	10.1.1.2	1o0.0	0
RVTEP-IP	IFL-Idx	NH-Id		
10.1.1.2	559	1728		
VNID	MC-Group-IP			
100	224.2.2.2			

Meaning The VTEP on QFX5100-96S 2 is reachable because its IP address (the address assigned to the loopback interface) appears in the output. The output also shows that the VXLAN (VNI 100) and corresponding multicast group are configured correctly on the remote VTEP.

Verifying That the Local VTEP Is Configured Correctly

Purpose On QFX5100-96S 1, verify that the tunnel endpoint is correct.

Action user@switch> show ethernet-switching vxlan-tunnel-end-point source

Logical System Name	Id	SVTEP-IP	IFL	L3-Idx
<default>	0	10.1.1.1	1o0.0	0
L2-RTT	Bridge Domain		VNID	MC-Group-IP
default-switch	VLAN1+100		100	224.2.2.2

Meaning The VTEP on QFX5100-96S 1 shows the correct tunnel source IP address (assigned to the loopback interface), VLAN, and multicast group for the VXLAN.

Verifying MAC Learning from the Remote VTEP

Purpose On QFX5100-96S 1, verify that it is learning MAC addresses from the remote VTEP.

Action user@switch> show ethernet-switching table

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)

Ethernet switching table : 2 entries, 2 learned

Routing instance : default-switch

Vlan name	MAC address	MAC flags	Age	Logical interface
VLAN1	00:00:00:ff:ff:ff	D	-	vtep.12345
VLAN1	00:10:94:00:00:02	D	-	xe-0/0/0.0

Meaning The output shows the MAC addresses learned from the remote VTEP (in addition to those learned on the normal Layer 2 interfaces). It also shows the logical name of the remote VTEP interface (vtep.12345 in the above output).

Monitor the Remote Interface

Purpose On QFX5100-96S 1, monitor traffic details for the remote VTEP interface.

Action user@switch> show interface vtep.12345 detail

```

M   Flags: Up SNMP-Traps Encapsulation: ENET2
      VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 10.1.1.2, L2 Routing
Instance: default-switch, L3 Routing Instance: default
      Traffic statistics:
        Input bytes :          228851738624
        Output bytes :              0
        Input packets:          714162415
        Output packets:              0
      Local statistics:
        Input bytes :              0
        Output bytes :              0
        Input packets:              0
        Output packets:              0
      Transit statistics:
        Input bytes :          228851738624          0 bps
        Output bytes :              0          0 bps
        Input packets:          714162415          0 pps
        Output packets:              0          0 pps
      Protocol eth-switch, MTU: 1600, Generation: 277, Route table: 5

```

Meaning The output shows traffic details for the remote VTEP interface. To get this information, you must supply the logical name of the remote VTEP interface (vtep.12345 in the above output), which you can learn by using the **show ethernet-switching table** command.

Related Documentation

- [Understanding VXLANs on page 6](#)
- [VXLAN Constraints on QFX Series Switches on page 57](#)

Verifying That a Local VXLAN VTEP Is Configured Correctly

Purpose Verify that a local VTEP is correct.

Action user@switch> show ethernet-switching vxlan-tunnel-end-point source

Logical System Name	Id	SVTEP-IP	IFL	L3-Idx
<default>	0	10.1.1.1	lo0.0	0
L2-RTT	Bridge Domain		VNID	MC-Group-IP
default-switch	VLAN1+100		100	232.1.1.1

Meaning The output shows the correct tunnel source IP address (loopback address), VLAN, and multicast group for the VXLAN.

Related Documentation

- [Understanding VXLANs on page 6](#)

Verifying MAC Learning from a Remote VTEP

Purpose Verify that a local VTEP is learning MAC addresses from a remote VTEP.

Action user@switch> show ethernet-switching table

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)

Ethernet switching table : 2 entries, 2 learned

Routing instance : default-switch

Vlan name	MAC address	MAC flags	Age	Logical interface
VLAN1	00:00:00:ff:ff:ff	D	-	vtep.12345
VLAN1	00:10:94:00:00:02	D	-	xe-0/0/0.0

Meaning The output shows the MAC addresses learned from the remote VTEP (in addition to those learned on the normal Layer 2 interfaces). It also shows the logical name of the remote VTEP interface (**vtep.12345** in the above output).

Related Documentation

- [Understanding VXLANs on page 6](#)

PART 3

Troubleshooting

- [Troubleshooting Tasks on page 73](#)

CHAPTER 4

Troubleshooting Tasks

- [Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDb-Managed VXLAN on page 73](#)
- [Verifying VXLAN Reachability on page 75](#)
- [Monitoring a Remote VTEP Interface on page 75](#)

Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDb-Managed VXLAN

Problem **Description:** The **Flags** field in the **show ovssdb logical-switch** operational mode command output is one of the following:

- **Created by Controller**
- **Created by L2ALD**
- **Tunnel key mismatch**

Cause

- If the **Flags** field displays **Created by Controller**, a logical switch is configured in the NSX environment or a virtual network is configured in the Contrail environment. However, an equivalent VXLAN is not configured or is improperly configured on the Juniper Networks device.
- If the **Flags** field displays **Created by L2ALD**, a VXLAN is configured on the Juniper Networks device. However, an equivalent logical switch is not configured in the NSX environment or an equivalent virtual network is not configured in the Contrail environment.
- If the **Flags** field displays **Tunnel key mismatch**, the VXLAN network identifier (VNI) specified in the logical switch configuration or the VXLAN identifier specified in the virtual network configuration do not match the VNI in the equivalent VXLAN configuration.

Solution If the **Flags** field displays **Created by Controller**, take the following action:

- On a QFX Series switch, verify that the **set switch-options ovssdb-managed** configuration command was issued in the Junos OS CLI. Issuing this command and committing the configuration enable the Juniper Networks device to dynamically create OVSDb-managed VXLANs.

Another possible cause is that the L2ALD daemon has become nonfunctional. If this is the case, wait for a few seconds, reissue the **show ovssdb logical-switch** operational mode command, and recheck the setting of the Flags field.

Another possible cause is that the Juniper Networks device dynamically configured the VXLAN and its associated logical interface, but there is an error in the configuration of these entities themselves or in an entity that was committed in the same transaction. If there is an issue with one or more of the configurations in a transaction, all configurations in the transaction, even the ones that are correctly configured, remain uncommitted and in a queue until you troubleshoot and resolve the configuration issues. As a result, the Juniper Networks device was unable to commit all configurations in the transaction. For this situation, enter the **show ovssdb commit failures** operational mode command. In the output that is displayed, determine which configurations are erroneous. Issues that can cause commitment errors include but are not limited to the detection of the same VXLAN name or VXLAN network identifier (VNI) in a dynamically configured VXLAN and in a VXLAN that was previously configured using the Junos OS CLI. After resolving the errors, enter the **clear ovssdb commit failures** command to remove the transaction from the queue and then retry committing all configurations in the transaction.

- On all other Juniper Networks devices that support VXLAN and OVSDB, determine whether a VXLAN equivalent to the logical switch configuration or virtual network configuration exists on the device. If the VXLAN is not configured, configure it using the procedure in *Configuring OVSDB-Managed VXLANs*. If a VXLAN is configured, check the VXLAN name to make sure that it is the same as the universally unique identifier (UUID) of the logical switch (NSX) or virtual network (Contrail) configuration. Also, check the VNI to make sure that the value is the same as the value in the logical switch (NSX) or virtual network (Contrail) configuration.

If the Flags field displays **Created by L2ALD**, take the following action:

- On a QFX Series switch, two issues exist. First, despite the fact that the Juniper Networks device dynamically creates OVSDB-managed VXLANs, this VXLAN was configured by using the Junos OS CLI. Second, a corresponding logical switch (NSX) or virtual network (Contrail) was not configured. To resolve both issues, configure a logical switch in the NSX environment or a virtual network in the Contrail environment. After the software-defined networking (SDN) controller pushes relevant logical switch or virtual network information to the Juniper Networks device, the device dynamically creates a corresponding VXLAN and deletes the VXLAN configured using the Junos OS CLI.
- On all other Juniper Networks devices that support VXLAN and OVSDB, determine whether an equivalent logical switch is configured in the NSX environment or a virtual network is configured in the Contrail environment. If a logical switch or virtual network is not configured, configure one, keeping in mind that a UUID is automatically generated for the logical switch or virtual network and that this UUID must be used as the name of the VXLAN. That is, the VXLAN name must be reconfigured with the logical switch or virtual network UUID.

Another possibility is that the logical switch or virtual network configuration might exist, but the UUID of the entity might not match the VXLAN name. In the NSX or

Contrail environment, check for a logical switch or virtual network, respectively, that has the same configuration as the VXLAN but has a different UUID.

If the Flags field displays **Tunnel key mismatch**, take the following action:

- For a QFX Series switch, check the configuration of the VNI in the NSX environment or the VXLAN identifier in the Contrail environment to see whether it was changed after the Juniper Networks device dynamically created the equivalent VXLAN. If it was changed, update the VNI on the QFX Series switch using the Junos OS CLI.
- On all other Juniper Networks devices that support VXLAN and OVSDb, check the value of the VNI in the NSX environment or the VXLAN identifier in the Contrail environment and the Junos OS CLI. Change the incorrect value.

Related Documentation

- [Understanding Dynamically Configured VXLANs in an OVSDb Environment on page 28](#)
- [Understanding How to Manually Configure OVSDb-Managed VXLANs](#)
- [show ovssdb logical-switch on page 106](#)
- [show ovssdb commit failures on page 100](#)
- [clear ovssdb commit failures on page 98](#)

Verifying VXLAN Reachability

Purpose On the local VTEP, verify that there is connectivity with the remote VTEP.

Action `user@switch> show ethernet-switching vxlan-tunnel-end-point remote`

Logical System Name	Id	SVTEP-IP	IFL	L3-Idx
<default>	0	10.1.1.2	100.0	0
RVTEP-IP	IFL-Idx	NH-Id		
10.1.1.2	559	1728		
VNID	MC-Group-IP			
100	232.1.1.1			

Meaning The remote VTEP is reachable because its IP address appears in the output. The output also shows that the VXLAN (VNI 100) and corresponding multicast group are configured correctly on the remote VTEP.

Related Documentation

- [Understanding VXLANs on page 6](#)
- [Manually Configuring VXLANs on QFX Series Switches on page 60](#)
- [Examples: Manually Configuring VXLANs on QFX Series Switches on page 61](#)

Monitoring a Remote VTEP Interface

Purpose Monitor traffic details for a remote VTEP interface.

Action user@switch> show interface *logical-name* detail

```

M   Flags: Up SNMP-Traps Encapsulation: ENET2
      VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 10.1.1.2, L2 Routing
Instance: default-switch, L3 Routing Instance: default
  Traffic statistics:
    Input bytes :          228851738624
    Output bytes :              0
    Input packets:          714162415
    Output packets:           0
  Local statistics:
    Input bytes :              0
    Output bytes :              0
    Input packets:              0
    Output packets:             0
  Transit statistics:
    Input bytes :          228851738624          0 bps
    Output bytes :              0              0 bps
    Input packets:          714162415          0 pps
    Output packets:           0              0 pps
  Protocol eth-switch, MTU: 1600, Generation: 277, Route table: 5

```

Meaning The output shows traffic details for the remote VTEP interface. To get this information, you must supply the logical name of the remote VTEP interface (vtep.12345 in the above output), which you can learn by using the **show ethernet-switching table** command.

- Related Documentation**
- [Understanding VXLANs on page 6](#)
 - [Manually Configuring VXLANs on QFX Series Switches on page 60](#)
 - [Examples: Manually Configuring VXLANs on QFX Series Switches on page 61](#)

PART 4

Configuration Statements and Operational Commands

- [OVSDB Configuration Statements on page 79](#)
- [VXLAN Configuration Statements on page 91](#)
- [OVSDB Operational Commands on page 97](#)
- [VXLAN Operational Commands on page 123](#)

CHAPTER 5

OVSDB Configuration Statements

- [controller \(OVSDB\) on page 80](#)
- [inactivity-probe-duration on page 81](#)
- [interfaces \(OVSDB\) on page 82](#)
- [maximum-backoff-duration on page 83](#)
- [ovsdb on page 84](#)
- [ovsdb-managed on page 85](#)
- [port \(OVSDB\) on page 86](#)
- [protocol \(OVSDB\) on page 87](#)
- [traceoptions \(OVSDB\) on page 88](#)

controller (OVSDB)

Syntax	<pre> controller <i>ip-address</i> { <i>inactivity-probe-duration</i> <i>milliseconds</i>; <i>maximum-backoff-duration</i> <i>milliseconds</i>; protocol <i>protocol</i> { port <i>number</i>; } } </pre>
Hierarchy Level	[edit protocols ovsdb]
Release Information	<p>Statement introduced in Junos OS Release 14.1R2.</p> <p>Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.</p> <p>Statement introduced in Junos OS Release 14.2 for EX Series switches.</p>
Description	<p>Configure a connection between a Juniper Networks device running the Open vSwitch Database (OVSDB) management protocol and a software-defined networking (SDN) controller. You can connect a Juniper Networks device to more than one SDN controller for redundancy.</p> <p>In a VMware NSX environment, one cluster of NSX controllers typically includes three or five controllers. To implement the OVSDB management protocol on a Juniper Networks device, you must explicitly configure a connection to one NSX controller, using the Junos OS CLI. If the NSX controller to which you explicitly configure a connection is in a cluster, the controller pushes information about other controllers in the same cluster to the device, and the device establishes connections with the other controllers. However, you can also explicitly configure connections with the other controllers in the cluster, using the Junos OS CLI.</p> <p>To implement the OVSDB management protocol on a Juniper Networks device in a Contrail environment, you must configure a connection to a Contrail controller, using the Junos OS CLI.</p> <p>Connections to all SDN controllers are made on the management interface of the Juniper Networks device.</p>
Options	<p><i>ip-address</i>—IPv4 address of the SDN controller.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Setting Up the OVSDB Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27 • Setting Up the OVSDB Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs

- [Understanding How to Set Up OVSDb Connections Between Juniper Networks Devices and SDN Controllers on page 24](#)

inactivity-probe-duration

Syntax	<code>inactivity-probe-duration <i>milliseconds</i>;</code>
Hierarchy Level	[edit protocols ovsdb controller]
Release Information	Statement introduced in Junos OS Release 14.1R2. Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Statement introduced in Junos OS Release 14.2 for EX Series switches.
Description	Configure the maximum amount of time, in milliseconds, that the connection between a Juniper Networks device that supports the Open vSwitch Database (OVSDb) management protocol and a software-defined networking (SDN) controller can be inactive before an inactivity probe is sent.
Options	<i>milliseconds</i> —Number of milliseconds that the connection can be inactive before an inactivity probe is sent. Range: 0 through 4,294,967,295 Default: 0. This value indicates that an inactivity probe is never sent.
Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Setting Up the OVSDb Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27 • Setting Up the OVSDb Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs • Understanding How to Set Up OVSDb Connections Between Juniper Networks Devices and SDN Controllers on page 24

interfaces (OVSDB)

Syntax	<code>interfaces <i>interface-name</i>;</code>
Hierarchy Level	[edit protocols ovsdb]
Release Information	Statement introduced in Junos OS Release 14.1R2. Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Statement introduced in Junos OS Release 14.2 for EX Series switches.
Description	Specify the physical interfaces on a Juniper Networks device that you want the Open vSwitch Database (OVSDB) management protocol to manage. Typically, the only interfaces that need to be managed by OVSDB are interfaces that are connected to physical servers.
Options	<i>interface-name</i> —Name of the interface.
Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• <i>Setting Up the OVSDB Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs</i>• Setting Up the OVSDB Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27

maximum-backoff-duration

Syntax	<code>maximum-backoff-duration <i>milliseconds</i>;</code>
Hierarchy Level	[edit protocols ovsdb controller]
Release Information	Statement introduced in Junos OS Release 14.1R2. Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Statement introduced in Junos OS Release 14.2 for EX Series switches.
Description	Specify (in milliseconds) how long a Juniper Networks device that supports the Open vSwitch Database (OVSDb) management protocol waits before it tries again to connect with a software-defined networking (SDN) controller after a previous attempt has failed.
Options	<i>milliseconds</i> —Number of milliseconds a Juniper Networks device waits before it tries again to connect with an SDN controller. Range: 1000 through 4,294,967,295 Default: 1000
Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Setting Up the OVSDb Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27 • Setting Up the OVSDb Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs • Understanding How to Set Up OVSDb Connections Between Juniper Networks Devices and SDN Controllers on page 24

ovsdb

```
Syntax  ovsdb {
        controller ip-address {
            inactivity-probe-duration milliseconds;
            maximum-backoff-duration milliseconds;
            protocol protocol {
                port number;
            }
        }
        interfaces interface-name;
        traceoptions {
            file <filename> <files number> <match regular-expression> <no-world-readable |
              world-readable> <size size>;
            flag flag;
            no-remote-trace;
        }
    }
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 14.1R2.
Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.
Statement introduced in Junos OS Release 14.2 for EX Series switches.

Description Configure support for the Open vSwitch Database (OVSDB) management protocol on a Juniper Networks device.

The remaining statements are explained separately.

Default The OVSDB management protocol is disabled on Juniper Networks devices.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- [Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 12](#)


ovsdb-managed

Syntax	ovsdb-managed;
Hierarchy Level	<p>[edit bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> switch-options], [edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> vxlan], [edit switch-options], [edit vlans <i>vlan-name</i> vxlan]</p>
Release Information	<p>Statement introduced in Junos OS Release 14.1R2.</p> <p>Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.</p> <p>Statement introduced in Junos OS Release 14.2 for EX Series switches.</p>
Description	<p>Disable a Juniper Networks device from learning about other Juniper Networks devices that function as hardware virtual tunnel endpoints (VTEPs) in a specified Virtual Extensible LAN (VXLAN) and the media access control (MAC) addresses learned by the hardware VTEPs. Instead, the Juniper Networks device uses the Open vSwitch Database (OVSDb) management protocol to learn about the hardware VTEPs in the VXLAN and the MAC addresses learned by the hardware VTEPs.</p> <p>The specified VXLAN must have a VXLAN network identifier (VNI) configured, using the vni statement in the [edit bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instance <i>routing-instance-name</i> vxlan], or [edit vlans <i>vlan-name</i> vxlan] hierarchy.</p> <p>Also, for OVSDb-managed VXLANs, the multicast scheme described in “Understanding How Layer 2 BUM and Layer 3 Routed Multicast Traffic Are Handled with OVSDb” on page 16 is used. Therefore, specifying the multicast-group statement in the [edit bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> vxlan], or [edit vlans <i>vlan-name</i> vxlan] hierarchy has no effect.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Understanding Dynamically Configured VXLANs in an OVSDb Environment on page 28 • Configuring OVSDb-Managed VXLANs

port (OVSDB)

Syntax	<code>port <i>number</i>;</code>
Hierarchy Level	[edit protocols ovsdb controller protocol]
Release Information	Statement introduced in Junos OS Release 14.1R2. Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Statement introduced in Junos OS Release 14.2 for EX Series switches.
Description	Specify the software-defined networking (SDN) controller port to which a Juniper Networks device that supports the Open vSwitch Database (OVSDB) management protocol connects.
Options	<i>number</i> —Number of the SDN controller port. Range: 1024 through 65,535 Default: 6632
Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Setting Up the OVSDB Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27 • Setting Up the OVSDB Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs • Understanding How to Set Up OVSDB Connections Between Juniper Networks Devices and SDN Controllers on page 24

protocol (OVSDb)

Syntax	<code>protocol protocol { port number; }</code>
Hierarchy Level	[edit protocols <code>ovsdb controller</code>]
Release Information	Statement introduced in Junos OS Release 14.1R2. Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Statement introduced in Junos OS Release 14.2 for EX Series switches.
Description	<p>Configure the security protocol that protects the connection between a Juniper Networks device that supports the Open vSwitch Database (OVSDb) management protocol and a software-defined networking (SDN) controller.</p> <p>The Secure Sockets Layer (SSL) connection requires a private key and certificates, which must be stored in the <code>/var/db/certs</code> directory of the Juniper Networks device. See “Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers” on page 25.</p>
Options	<i>protocol</i> —Establish a secure connection to the SDN controller, using SSL or TCP.
<div>  NOTE: SSL is the only supported connection protocol. </div>	
Default: <code>ssl</code>	
The remaining statement is explained separately.	
Required Privilege Level	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> • Setting Up the OVSDb Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27 • Setting Up the OVSDb Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs • Understanding How to Set Up OVSDb Connections Between Juniper Networks Devices and SDN Controllers on page 24

traceoptions (OVSDB)

Syntax	<pre> traceoptions { file <filename> <files number> <match regular-expression> <no-world-readable world-readable> <size size>; flag flag; no-remote-trace; } </pre>
Hierarchy Level	[edit protocols ovsdb]
Release Information	<p>Statement introduced in Junos OS Release 14.1R2.</p> <p>Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.</p> <p>Statement introduced in Junos OS Release 14.2 for EX Series switches.</p>
Description	Define tracing operations for the Open vSwitch Database (OVSDB) management protocol, which is supported on Juniper Networks devices.
Default	If you do not include this statement, OVSDB-specific tracing operations are not performed.
Options	<p>file <i>filename</i>—Name of file in which the system places the output of the tracing operations. By default, the system places all files in the /var/log directory.</p> <p>Default: /var/log/vgd</p> <p>files <i>number</i>—(Optional) Maximum number of trace files. When a trace file reaches the size specified by the size option, the filename is appended with 0 and compressed. For example, a trace file named trace-file.gz would be renamed trace-file.0.gz. When trace-file.0.gz reaches the specified size, it is renamed trace-file.1.gz and its contents are compressed to trace-file.0.gz. 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 and a filename.</p> <p>Range: 2 through 1000 files</p> <p>Default: 10 files</p> <p>flag <i>flag</i>—Tracing operation to perform. You can include one or more of the following flags:</p> <ul style="list-style-type: none"> all—All OVSDB events. configuration—OVSDB configuration events. core—OVSDB core events. function—OVSDB function events. interface—OVSDB interface events. l2-client—OVSDB Layer 2 client events.

netconf-client—(QFX Series switches only) Events for the dynamic configuration of Virtual Extensible LANs (VXLANs).

ovs-client—OVSDDB client events.

match *regular-expression*—(Optional) Only log lines that match the regular expression.

no-remote-trace—(Optional) Disable tracing and logging operations that track normal operations, error conditions, and packets that are generated by or passed through the Juniper Networks device.

no-world-readable—Restrict access to the trace files to the owner.

Default: no-world-readable

size *size*—(Optional) Maximum size of each trace file in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). If you do not specify a unit, the default is bytes. If you specify a maximum file size, you also must specify a maximum number of trace files by using the **files** option and a filename by using the **file** option.

Syntax: *size* to specify bytes, *sizek* to specify KB, *sizem* to specify MB, or *sizeg* to specify GB.

Range: 10,240 through 1,073,741,824 bytes

Default: 128 KB

world-readable—Enable any user to access the trace files.

Required Privilege	admin—To view this statement in the configuration.
Level	admin-control—To add this statement to the configuration.

CHAPTER 6

VXLAN Configuration Statements

- [decapsulate-accept-inner-vlan on page 91](#)
- [encapsulate-inner-vlan on page 92](#)
- [multicast-group on page 92](#)
- [ovsdb-managed on page 93](#)
- [unreachable-vtep-aging-timer on page 94](#)
- [vni on page 94](#)
- [vtep-source-interface on page 95](#)
- [vxlan on page 95](#)

[decapsulate-accept-inner-vlan](#)

Syntax	<code>decapsulate-accept-inner-vlan</code>
Hierarchy Level	<code>[edit protocols l2-learning]</code>
Release Information	Statement modified in Junos OS Release 14.1X53 for the QFX Series.
Description	Configure the switch to de-encapsulate and accept original VLAN tags in Virtual Extensible LAN (VXLAN) packets.
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 VXLANs on page 6• encapsulate-inner-vlan on page 92

encapsulate-inner-vlan

Syntax	encapsulate-inner-vlan
Hierarchy Level	[edit vlans <i>vlan-name</i> vxlan]
Release Information	Statement introduced in Junos OS Release 14.1X53-D10.
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	<ul style="list-style-type: none">• Understanding VXLANs on page 6• Manually Configuring VXLANs on QFX Series Switches on page 60• Examples: Manually Configuring VXLANs on QFX Series Switches on page 61• decapsulate-accept-inner-vlan on page 91

multicast-group

Syntax	multicast-group <i>address</i>
Hierarchy Level	[edit vlans <i>vlan-name</i> vxlan]
Release Information	Statement introduced in Junos OS Release 14.1X53-D10.
Description	Assign a multicast group address to a Virtual Extensible LAN (VXLAN). All members of a VXLAN must use the same multicast group address.
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 VXLANs on page 6• Manually Configuring VXLANs on QFX Series Switches on page 60• Examples: Manually Configuring VXLANs on QFX Series Switches on page 61

ovsdb-managed

Syntax	ovsdb-managed;
Hierarchy Level	<p>[edit bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> switch-options], [edit routing-instances <i>routing-instance-name</i> vlans <i>vlan-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> vxlan], [edit switch-options], [edit vlans <i>vlan-name</i> vxlan]</p>
Release Information	<p>Statement introduced in Junos OS Release 14.1R2.</p> <p>Statement introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.</p> <p>Statement introduced in Junos OS Release 14.2 for EX Series switches.</p>
Description	<p>Disable a Juniper Networks device from learning about other Juniper Networks devices that function as hardware virtual tunnel endpoints (VTEPs) in a specified Virtual Extensible LAN (VXLAN) and the media access control (MAC) addresses learned by the hardware VTEPs. Instead, the Juniper Networks device uses the Open vSwitch Database (OVSDB) management protocol to learn about the hardware VTEPs in the VXLAN and the MAC addresses learned by the hardware VTEPs.</p> <p>The specified VXLAN must have a VXLAN network identifier (VNI) configured, using the vni statement in the [edit bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instance <i>routing-instance-name</i> vxlan], or [edit vlans <i>vlan-name</i> vxlan] hierarchy.</p> <p>Also, for OVSDB-managed VXLANs, the multicast scheme described in “Understanding How Layer 2 BUM and Layer 3 Routed Multicast Traffic Are Handled with OVSDB” on page 16 is used. Therefore, specifying the multicast-group statement in the [edit bridge-domains <i>bridge-domain-name</i> vxlan], [edit routing-instances <i>routing-instance-name</i> vxlan], or [edit vlans <i>vlan-name</i> vxlan] hierarchy has no effect.</p>
Required Privilege Level	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Understanding Dynamically Configured VXLANs in an OVSDB Environment on page 28 • Configuring OVSDB-Managed VXLANs

unreachable-vtep-aging-timer

Syntax	unreachable-vtep-aging-timer [300–1800]
Hierarchy Level	[edit vlans <i>vlan-name</i> vxlان]
Release Information	Statement introduced in Junos OS Release 14.1X53-D10.
Description	Configure the system to age out the address for the remote virtual tunnel endpoint (VTEP) if all the MAC addresses learned from that VTEP age out. The address for the remote VTEP expires the configured number of seconds after the last learned media access control (MAC) address expires.
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 VXLANs on page 6• Manually Configuring VXLANs on QFX Series Switches on page 60• Examples: Manually Configuring VXLANs on QFX Series Switches on page 61

vni

Syntax	vni [1–16777214]
Hierarchy Level	[edit vlans <i>vlan-name</i> vxlان]
Release Information	Statement introduced in Junos OS Release 14.1X53-D10.
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">• Understanding VXLANs on page 6• Manually Configuring VXLANs on QFX Series Switches on page 60• Examples: Manually Configuring VXLANs on QFX Series Switches on page 61

vtep-source-interface

Syntax	<code>vtep-source-interface <i>logical-interface</i>;</code>
Hierarchy Level	[edit switch-options]
Release Information	Statement introduced in Junos OS Release 14.1X53-D10.
Description	Configure a source interface for a Virtual Extensible LAN (VXLAN) tunnel. You must provide the name of a logical interface configured on the 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 VXLANs on page 6 • Manually Configuring VXLANs on QFX Series Switches on page 60 • Examples: Manually Configuring VXLANs on QFX Series Switches on page 61

vxlan

Syntax	<pre> vxlan { encapsulate-inner-vlan ingress-node-replication multicast-group ovsdb-managed unreachable-vtep-aging-timer vni } </pre>
Hierarchy Level	[edit vlans]
Release Information	Statement introduced in Junos OS Release 14.1X53-D10. ingress-node-replication option added for QFX Series switches in Junos OS Release 14.1X53-D30.
Description	Configure support for Virtual Extensible LANs (VXLANs) on a Juniper Networks device.
Options	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"> • Understanding VXLANs on page 6 • Manually Configuring VXLANs on QFX Series Switches on page 60 • Examples: Manually Configuring VXLANs on QFX Series Switches on page 61

CHAPTER 7

OVSDb Operational Commands

- `clear ovldb commit failures`
- `show ovldb commit failures`
- `show ovldb controller`
- `show ovldb interface`
- `show ovldb logical-switch`
- `show ovldb mac`
- `show ovldb statistics interface`
- `show ovldb virtual-tunnel-end-point`
- `show vpls mac-table`

clear ovssdb commit failures

Syntax clear ovssdb commit failures
<transaction-id>

Release Information Command introduced in Junos OS Release 14.1X53-D26 for QFX Series switches.

Description Remove a transaction from a queue maintained by a Juniper Networks switch that supports the Open vSwitch Database (OVSSDB) management protocol and Virtual Extensible LANs (VXLANs). The transaction includes OVSSDB-managed VXLANs and associated logical interfaces that the Juniper Networks switch dynamically configured and tried to commit but was unable to because of an issue with one or more of the configurations. In addition to removing the transaction, entering the **clear ovssdb commit failures** command causes the Juniper Networks switch to automatically retry committing all configurations in the transaction.

If there is an issue with one or more of the configurations in a transaction, this causes all configurations in the transaction, even the ones that are correctly configured, to remain uncommitted and in the queue until you troubleshoot and resolve the configuration issue(s).

You can display an erroneous transaction by entering the **show ovssdb commit failures** command. In the output that appears, you must determine which configuration(s) are erroneous and therefore prevent the Juniper Networks switch from committing the configurations in the transaction.

Issues that can cause commitment errors include but are not limited to the detection of the same VXLAN name or VXLAN network identifier (VNI) in a dynamically configured VXLAN and in a VXLAN that was previously configured using the Junos OS CLI.

To monitor for issues with dynamically configured OVSSDB-managed VXLANs and their associated interfaces, we recommend checking for system log messages and traceoptions files for OVSSDB.

After resolving the error(s), enter the **clear ovssdb commit failures** command to remove the transaction from the queue and retry committing all configurations in the transaction.



NOTE: While an erroneous transaction exists in the queue, the Juniper Networks switch cannot commit the dynamic configurations of additional VXLANs and their associated logical interfaces. The commitment of these VXLANs and logical interfaces remain in a pending state until all VXLAN and logical interface configurations in the erroneous transaction are resolved and successfully committed.

Options none—Remove the transaction that currently appears in the **show ovssdb commit failures** command output, and retry committing all configurations in the transaction.

transaction-id—Remove the transaction with the specified numerical ID, and retry committing the configurations in the transaction.

Required Privilege Level clear

Related Documentation • [show ovldb commit failures on page 100](#)

List of Sample Output [clear ovldb commit failures on page 99](#)
 [clear ovldb commit failures \(Specific Transaction\) on page 99](#)

Sample Output

[clear ovldb commit failures](#)

```
user@host> clear ovldb commit failures
```

[clear ovldb commit failures \(Specific Transaction\)](#)

```
user@host> clear ovldb commit failures 1
```

show ovssdb commit failures

Syntax `show ovssdb commit failures`
`<transaction-id>`

Release Information Command introduced in Junos OS Release 14.1X53-D26 for QFX Series switches.

Description Display configurations of Open vSwitch Database (OVSSDB)-managed Virtual Extensible LANs (VXLANs) and associated logical interfaces that the Juniper Networks switch dynamically configured but was unable to commit.

For each OVSSDB-managed VXLAN and associated logical interface that you plan to implement in a Junos OS environment, you must configure equivalent entities in NSX Manager or in the NSX API for an NSX environment, or in the Contrail Web user interface for a Contrail environment. The software-defined networking (SDN) controller pushes these configurations to the connected Juniper Networks switch by way of the OVSSDB schema for physical devices. After the Juniper Networks switch receives these configurations, it dynamically configures a Junos OS-equivalent VXLAN and associated logical interface, and attempts to commit the configurations.

During the commitment of the dynamic configurations, If there is an issue with one or more of the configurations, all configurations in the transaction, even the ones that are correctly configured, remain uncommitted and are saved in a queue. All configurations in the transaction remain uncommitted and in the queue until you troubleshoot and resolve the configuration issues. After you resolve the configuration issues, you must use the [clear ovssdb commit failures](#) command to remove the transaction from the queue and retry committing the configurations.



NOTE: While an erroneous transaction exists in the queue, the Juniper Networks switch cannot commit the dynamic configurations of additional VXLANs and their associated logical interfaces. The commitment of these VXLANs and logical interfaces remain in a pending state until all VXLAN and logical interface configurations in the erroneous transaction are resolved and successfully committed.

Issues that can cause commitment errors include but are not limited to the detection of the same VXLAN name or VXLAN network identifier (VNI) in a dynamically configured VXLAN and in a VXLAN that was previously configured using the Junos OS CLI.

To monitor for issues with dynamically configured OVSSDB-managed VXLANs and their associated interfaces, we recommend checking for system log messages and traceoptions files for OVSSDB.

Options **none**—Display information about an erroneous transaction.

transaction-id—Display information about the transaction with the specified numerical ID.

Required Privilege Level admin

Related Documentation

- [Understanding Dynamically Configured VXLANs in an OVSDb Environment on page 28](#)
- [traceoptions \(OVSDb\) on page 88](#)

List of Sample Output [show ovssdb commit failures on page 101](#)
[show ovssdb commit failure \(Specific Transaction\) on page 101](#)

Output Fields [Table 18 on page 101](#) lists the output fields for the **show ovssdb commit failures** command. Output fields are listed in the approximate order in which they appear.

Table 18: show ovssdb commit failures Output Fields

Field Names	Field Descriptions
Txn ID	ID assigned to a transaction by the Juniper Networks switch.
Logical-switch	Name of the VXLAN that the Juniper Networks switch dynamically configured but was unable to commit the configuration of.
Port	Name of an OVSDb-managed physical interface that is associated with the VXLAN.
VLAN ID	ID that is assigned to the VXLAN.

Sample Output

show ovssdb commit failures

```

user@host> show ovssdb commit failures
Txn ID      Logical-switch      Port      VLAN ID
1           28805c1d-0122-495d-85df-19abd647d772  xe-0/0/5:0  1016
1
1           9acc24b3-7b0a-4c2e-b572-3370c3e1acff  xe-0/0/5:0  1017
1
...
```

show ovssdb commit failure (Specific Transaction)

```

user@host> show ovssdb commit failures 1
Txn ID      Logical-switch      Port      VLAN ID
1           28805c1d-0122-495d-85df-19abd647d772  xe-0/0/5:0  1016
1
1           9acc24b3-7b0a-4c2e-b572-3370c3e1acff  xe-0/0/5:0  1017
1
...
```

show ovssdb controller

Syntax	<code>show ovssdb controller</code> <code><address ip-address></code>
Release Information	Command introduced in Junos OS Release 14.1R2. Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Command introduced in Junos OS Release 14.2 for EX Series switches.
Description	Display information and connection status for software-defined networking (SDN) controllers to which the Juniper Networks device is connected.
Options	none —Display information about all SDN controllers to which the Juniper Networks device is connected. address ip-address —Display information about the SDN controller at the specified IP address.
Required Privilege Level	admin
Related Documentation	<ul style="list-style-type: none"> • Setting Up the OVSSDB Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs • Setting Up the OVSSDB Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27 • Understanding How to Set Up OVSSDB Connections Between Juniper Networks Devices and SDN Controllers on page 24
List of Sample Output	show ovssdb controller on page 103 show ovssdb controller address on page 103
Output Fields	Table 19 on page 102 lists the output fields for the show ovssdb controller command. Output fields are listed in the approximate order in which they appear.

Table 19: show ovssdb controller Output Fields

Field Name	Field Description
Controller IP address	IP address of the SDN controller to which the Juniper Networks device is connected.
Controller protocol	Protocol used by the Juniper Networks device to initiate the connection.
Controller port	Port to which the Juniper Networks device is connected.
Controller connection	State of the connection with the SDN controller.
Controller seconds-since-connect	Number of seconds since the connection with the SDN controller was established.

Table 19: show ovsdb controller Output Fields (*continued*)

Field Name	Field Description
Controller seconds-since-disconnect	Number of seconds since the connection with the SDN controller was dropped.
Controller connection status	Status of the connection with the SDN controller.

Sample Output

show ovsdb controller

```

user@host> show ovsdb controller
VTEP controller information:
Controller IP address: 10.168.66.189
Controller protocol: ssl
Controller port: 6632
Controller connection: up
Controller seconds-since-connect: 56290
Controller seconds-since-disconnect: 0
Controller connection status: active

Controller IP address: 10.168.181.54
Controller protocol: ssl
Controller port: 6632
Controller connection: up
Controller seconds-since-connect: 56292
Controller seconds-since-disconnect: 0
Controller connection status: active

Controller IP address: 10.168.182.45
Controller protocol: ssl
Controller port: 6632
Controller connection: up
Controller seconds-since-connect: 56292
Controller seconds-since-disconnect: 0
Controller connection status: active

```

show ovsdb controller address

```

user@host> show ovsdb controller address 10.168.182.45
VTEP controller information:
Controller IP address: 192.168.182.45
Controller protocol: ssl
Controller port: 6632
Controller connection: up
Controller seconds-since-connect: 56347
Controller seconds-since-disconnect: 0
Controller connection status: active

```

show ovssdb interface

Syntax	<code>show ovssdb interface</code> <code><interface-name></code>
Release Information	Command introduced in Junos OS Release 14.1R2. Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Command introduced in Junos OS Release 14.2 for EX Series switches.
Description	Display information about Open vSwitch Database (OVSSDB)-managed interfaces configured by using the <code>interfaces interface-name</code> statement in the <code>[edit protocols ovssdb]</code> hierarchy.
Options	none —Display information about all OVSSDB-managed interfaces. interface-name —Display information about the specified OVSSDB-managed interface.
Required Privilege Level	admin
Related Documentation	<ul style="list-style-type: none"> Setting Up the OVSSDB Protocol on Juniper Networks Devices that Support Manual Configuration of VXLANs Setting Up the OVSSDB Protocol on Juniper Networks Devices that Support the Dynamic Configuration of VXLANs on page 27
List of Sample Output	show ovssdb interface on page 104 show ovssdb (Specific Interface) on page 105
Output Fields	Table 20 on page 104 lists the output fields for the <code>show ovssdb interface</code> command. Output fields are listed in the approximate order in which they appear.

Table 20: show ovssdb interface Output Fields

Field Name	Field Description
Interface	Name of interface.
VLAN ID	ID of Virtual Extensible LAN (VXLAN) with which the interface is associated. NOTE: This field is not supported by MX Series routers or EX9200 switches.
Bridge domain or VLAN	Bridge domain or VLAN under which the VXLAN is created. NOTE: This field is not supported by MX Series routers or EX9200 switches.

Sample Output

show ovssdb interface

```
user@host> show ovssdb interface
```


Interface	VLAN ID	Bridge-domain
ge-7/0/9.0		
ge-7/0/9.1		
irb.11		
irb.12		
irb.2		
irb.3		
xe-10/3/0.0		
xe-10/3/0.1		

show ovbdb (Specific Interface)

```
user@host> show ovbdb interface ge-7/0/9.0
```

Interface	VLAN ID	Bridge-domain
ge-7/0/9.0		

show ovssdb logical-switch

Syntax `show ovssdb logical-switch`
 `<logical-switch-name>`

Release Information Command introduced in Junos OS Release 14.1R2.
 Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.
 Command introduced in Junos OS Release 14.2 for EX Series switches.

Description



NOTE: In the Open vSwitch Database (OVSSDB) schema for physical devices, the logical switch table stores information about the Layer 2 broadcast domain that you configured in a VMware NSX or Contrail environment. In the NSX environment, the Layer 2 broadcast domain is known as a *logical switch*, while in the Contrail environment, the domain is known as a *virtual network*.

In the context of the `show ovssdb logical-switch` command, the term *logical switch* refers to the logical switch or virtual network that was configured in the NSX or Contrail environments, respectively, and the corresponding configuration that was pushed to the OVSSDB schema.

Display information about logical switches and the corresponding Virtual Extensible LANs (VXLANs), which were configured on the Juniper Networks device.

In the command output, each logical switch is identified by a universally unique identifier (UUID), which in the context of this command, is also known as a logical switch name.

The `show ovssdb logical-switch` command displays the state of the logical switch (**Flags**), which can be one of the following:

Created by Controller—A logical switch is configured. However, a corresponding VXLAN is not yet configured. In this state, the logical switch and corresponding VXLAN are not yet operational.

Created by L2ALD—A VXLAN is configured. However, a corresponding logical switch is not yet configured. In this state, the logical switch and corresponding VXLAN are not yet operational.

Created by both—A logical switch and a corresponding VXLAN are configured. In this state, the logical switch and corresponding VXLAN are operational.

Tunnel key mismatch—The VNIs specified in the logical switch and corresponding VXLAN configurations do not match. In this state, the logical switch and corresponding VXLAN are not yet operational.

Options **none**—Display information about all logical switches that are present in the OVSSDB schema for physical devices.

logical-switch-name—Display information about the specified logical switch.

Required Privilege Level admin

Related Documentation

- [OVSDb Schema for Physical Devices on page 14](#)
- [Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDb-Managed VXLAN on page 73](#)

List of Sample Output [show ovssdb logical-switch on page 107](#)
[show ovssdb logical-switch \(Specific Logical Switch\) on page 108](#)

Output Fields [Table 21 on page 107](#) lists the output fields for the **show ovssdb logical-switch** command. Output fields are listed in the approximate order in which they appear.

Table 21: show ovssdb logical-switch Output Fields

Field Name	Field Description
Logical Switch Name	UUID that is automatically generated and assigned to the logical switch. When you configure the corresponding VXLAN in the Junos OS CLI, you must specify the same UUID as the VXLAN name.
Flags	State of the logical switch. For possible states, see the Description section of this topic.
VNI	VNI that is configured for the logical switch and corresponding VXLAN.
Num of Remote MAC	The total number of remote media access control (MAC) addresses associated with the logical switch. These addresses are learned by software and hardware virtual tunnel endpoints (VTEPs).
Num of Local MAC	The total number of local MAC addresses associated with the logical switch. <i>Local MAC addresses</i> are addresses learned on the local physical ports.

Sample Output

show ovssdb logical-switch

```
user@host> show ovssdb logical-switch
Logical switch information:
Logical Switch Name: 24a76aff-7e61-4520-a78d-3eca26ad7510
Flags: Created by both
VNI: 3
Num of Remote MAC: 13
Num of Local MAC: 12
Logical Switch Name: 9b4f880e-dac8-4612-a832-97ad9dec270f
Flags: Created by Controller
VNI: 50
Num of Remote MAC: 0
Num of Local MAC: 0
Logical Switch Name: bc0da2da-6c16-44bf-b655-442484294ded
Flags: Created by Controller
VNI: 51
Num of Remote MAC: 0
Num of Local MAC: 0
```

show ovssdb logical-switch (Specific Logical Switch)

```
user@host> show ovssdb logical-switch 24a76aff-7e61-4520-a78d-3eca26ad7510
Logical switch information:
Logical Switch Name: 24a76aff-7e61-4520-a78d-3eca26ad7510
Flags: Created by both
VNI: 3
Num of Remote MAC: 13
Num of Local MAC: 12
```

show ovbdb mac

Syntax `show ovbdb mac`
 `<address mac-address>`
 `<local>`
 `<logical-switch logical-switch-uuid>`
 `<multicast>`
 `<remote>`
 `<unicast>`

Release Information Command introduced in Junos OS Release 14.1R2.
 Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches.
 Command introduced in Junos OS Release 14.2 for EX Series switches.

Description Display media access control (MAC) addresses, as well as information about the MAC addresses, learned by a Juniper Networks device that functions as a hardware virtual tunnel endpoint (VTEP). Using the Open vSwitch Database (OVSDb) management protocol, this hardware VTEP can learn about MAC addresses directly or from other software or hardware VTEPs. The MAC addresses learned directly by the hardware VTEP are known as *local addresses*, while the addresses learned from other software or hardware VTEPs are known as *remote addresses*.

Options Use one or more of the following options to display a more specific list of MAC addresses and information about the MAC addresses. For example, to display a list of local unicast MAC addresses, you can issue the **show ovbdb mac local unicast** command.

none—Display all MAC addresses, which includes all local, remote, unicast, and multicast addresses associated with all logical switches.

address *mac-address*—Display the specified MAC address.

count—(All Juniper Networks devices that support OVSDb except EX9200 switches)
 Display the number of MAC addresses learned by the Juniper Networks device. Using this option alone, the number includes all local, remote, unicast, and multicast MAC addresses associated with all logical switches in the logical switch table of the OVSDb schema for physical devices. You can use this option with one or more of the other options to display a more specific count of MAC addresses. For example, to display the number of local and remote unicast MAC addresses, you can issue the **show ovbdb mac count local remote unicast** command.

local—Display all local MAC addresses.

logical-switch *logical-switch-uuid*—Display all MAC addresses associated with the specified logical switch in the logical switch table of the OVSDb schema for physical devices.

multicast—Display all multicast MAC addresses.

remote—Display all remote MAC addresses.

unicast—Display all unicast MAC addresses.

Required Privilege Level admin

Related Documentation • [OVSDB Schema for Physical Devices on page 14](#)

List of Sample Output [show ovssdb mac on page 110](#)
[show ovssdb mac address on page 111](#)
[show ovssdb mac logical-switch on page 111](#)
[show ovssdb mac local unicast on page 112](#)
[show ovssdb mac \(Count of All Local, Remote, Unicast, and Multicast MAC Addresses for All Logical Switches\) on page 112](#)

Output Fields [Table 22 on page 110](#) lists the output fields for the **show ovssdb mac** command. Output fields are listed in the approximate order in which they appear.

Table 22: show ovssdb mac Output Fields

Field Name	Field Description
Logical Switch Name	Universally unique identifier (UUID) of the logical switch.
MAC Address	MAC addresses of virtual machines (VMs).
IP Address	IP address of VMs. <i>NOTE:</i> If the IP addresses of VMs are not published by the SDN controller, this field displays 0.0.0.0.
Encapsulation	Encapsulation type.
VTEP Address	IP address of the hardware or software VTEP from which the MAC address was learned. Further, this VTEP can forward VM traffic to the associated host.
MAC Count	<i>NOTE:</i> This field is supported by all Juniper Networks devices that support OVSDB except EX9200 switches. Number of all or specified MAC addresses learned by the Juniper Networks device.

Sample Output

show ovssdb mac

```

user@host> show ovssdb mac
Logical Switch Name: 24a76aff-7e61-4520-a78d-3eca26ad7510
  Mac          IP          Encapsulation  Vtep
  Address      Address
02:00:00:00:03:01  0.0.0.0      Vxlan over Ipv4  10.255.18.22
02:00:00:00:03:02  0.0.0.0      Vxlan over Ipv4  10.255.18.22
02:00:00:00:03:03  0.0.0.0      Vxlan over Ipv4  10.255.18.22
02:00:00:00:03:04  0.0.0.0      Vxlan over Ipv4  10.255.18.22
02:00:00:00:03:05  0.0.0.0      Vxlan over Ipv4  10.255.18.22
04:00:00:00:03:05  0.0.0.0      Vxlan over Ipv4  10.255.18.22
06:00:00:00:03:01  0.0.0.0      Vxlan over Ipv4  10.255.18.22
06:00:00:00:03:02  0.0.0.0      Vxlan over Ipv4  10.255.18.22

```

```

06:00:00:00:03:03    0.0.0.0    Vxlan over Ipv4    10.255.18.22
06:00:00:00:03:04    0.0.0.0    Vxlan over Ipv4    10.255.18.22
06:00:00:00:03:05    0.0.0.0    Vxlan over Ipv4    10.255.18.22
40:b4:f0:06:6f:f0    0.0.0.0    Vxlan over Ipv4    10.255.18.22
ff:ff:ff:ff:ff:ff    0.0.0.0    Vxlan over Ipv4    10.100.100.1

Logical Switch Name: bf6d4fd4-f5f6-430c-8c37-4033ef1c55ab
Mac                IP                Encapsulation      Vtep
Address            Address
02:00:00:00:11:01    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:02    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:03    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:04    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:05    0.0.0.0          Vxlan over Ipv4    10.1.1.29
04:00:00:00:11:05    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:01    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:02    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:03    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:04    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:05    0.0.0.0          Vxlan over Ipv4    10.1.1.29
40:b4:f0:06:6f:f0    0.0.0.0          Vxlan over Ipv4    10.1.1.29
00:23:9c:5e:a7:f0    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:01    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:02    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:03    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:04    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:05    0.0.0.0          Vxlan over Ipv4    10.255.18.22
ff:ff:ff:ff:ff:ff    0.0.0.0          Vxlan over Ipv4    10.110.110.1
...

```

show ovsdb mac address

```
user@host> show ovsdb mac address 02:00:00:00:03:01
```

```

Mac                IP                Encapsulation      Vtep
Address            Address
02:00:00:00:03:01    0.0.0.0          Vxlan over Ipv4    10.255.18.22

```

show ovsdb mac logical-switch

```
user@host> show ovsdb mac logical-switch bf6d4fd4-f5f6-430c-8c37-4033ef1c55ab
```

```

Logical Switch Name: bf6d4fd4-f5f6-430c-8c37-4033ef1c55ab
Mac                IP                Encapsulation      Vtep
Address            Address
02:00:00:00:11:01    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:02    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:03    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:04    0.0.0.0          Vxlan over Ipv4    10.1.1.29
02:00:00:00:11:05    0.0.0.0          Vxlan over Ipv4    10.1.1.29
04:00:00:00:11:05    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:01    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:02    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:03    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:04    0.0.0.0          Vxlan over Ipv4    10.1.1.29
06:00:00:00:11:05    0.0.0.0          Vxlan over Ipv4    10.1.1.29
40:b4:f0:06:6f:f0    0.0.0.0          Vxlan over Ipv4    10.1.1.29
00:23:9c:5e:a7:f0    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:01    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:02    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:03    0.0.0.0          Vxlan over Ipv4    10.255.18.22
08:00:00:00:11:04    0.0.0.0          Vxlan over Ipv4    10.255.18.22

```

08:00:00:00:11:05	0.0.0.0	Vxlan over Ipv4	10.255.18.22
ff:ff:ff:ff:ff:ff	0.0.0.0	Vxlan over Ipv4	10.110.110.1

show ovssdb mac local unicast

```
user@host> show ovssdb mac local unicast
```

```
Logical Switch Name: 24a76aff-7e61-4520-a78d-3eca26ad7510
```

Mac Address	IP Address	Encapsulation	Vtep Address
02:00:00:00:03:01	0.0.0.0	Vxlan over Ipv4	10.255.181.72
02:00:00:00:03:02	0.0.0.0	Vxlan over Ipv4	10.255.181.72
02:00:00:00:03:03	0.0.0.0	Vxlan over Ipv4	10.255.181.72
02:00:00:00:03:04	0.0.0.0	Vxlan over Ipv4	10.255.181.72
02:00:00:00:03:05	0.0.0.0	Vxlan over Ipv4	10.255.181.72
04:00:00:00:03:05	0.0.0.0	Vxlan over Ipv4	10.255.181.72
06:00:00:00:03:01	0.0.0.0	Vxlan over Ipv4	10.255.181.72
06:00:00:00:03:02	0.0.0.0	Vxlan over Ipv4	10.255.181.72
06:00:00:00:03:03	0.0.0.0	Vxlan over Ipv4	10.255.181.72
06:00:00:00:03:04	0.0.0.0	Vxlan over Ipv4	10.255.181.72
06:00:00:00:03:05	0.0.0.0	Vxlan over Ipv4	10.255.181.72
40:b4:f0:06:6f:f0	0.0.0.0	Vxlan over Ipv4	10.255.181.72

```
...
```

show ovssdb mac (Count of All Local, Remote, Unicast, and Multicast MAC Addresses for All Logical Switches)

```
user@host> show ovssdb mac count
```

```
MAC count: 6877
```


show ovssdb statistics interface

Syntax	<code>show ovssdb statistics interface</code> <code><interface-name></code>
Release Information	Command introduced in Junos OS Release 14.1R2. Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Command introduced in Junos OS Release 14.2 for EX Series switches.
Description	Display statistics for Open vSwitch Database (OVSDb)-managed interfaces configured by using the interfaces interface-name statement in the [edit protocols ovssdb] hierarchy. When an interface is configured as OVSDb-managed, the collection of statistics for that interface begins, and the statistics displayed at any given time reflects the data collected up to that point.
Options	none —Display statistics for all configured OVSDb-managed interfaces. interface-name —Display statistics for the specified interface.
Required Privilege Level	admin
Related Documentation	<ul style="list-style-type: none"> interfaces on page 82
List of Sample Output	show ovssdb statistics interface on page 113 show ovssdb statistics interface (Specific Interface) on page 114
Output Fields	Table 23 on page 113 lists the output fields for the show ovssdb statistics interface command. Output fields are listed in the approximate order in which they appear.

Table 23: show ovssdb statistics interface Output Fields

Field Name	Field Description
Num of rx pkts	Number of packets received by the interface.
Num of tx pkts	Number of packets sent by the interface.
Num of rx bytes	Number of bytes received by the interface.
Num of tx bytes	Number of bytes sent by the interface.

Sample Output

show ovssdb statistics interface

```

user@host> show ovssdb statistics interface
Interface Name: ge-7/0/9.0
Num of rx pkts: 945           Num of tx pkts: 113280890
Num of rx bytes: 56700        Num of tx bytes: 57531319540

```

Interface Name: ge-7/0/10.0	
Num of rx pkts: 459	Num of tx pkts: 473840856
Num of rx bytes: 84747	Num of tx bytes: 45830738532
Interface Name: ge-7/0/11.0	
Num of rx pkts: 305	Num of tx pkts: 367483456
Num of rx bytes: 98974	Num of tx bytes: 33495468092

show ovsdb statistics interface (Specific Interface)

```
user@host> show ovsdb statistics interface ge-7/0/9.0
```

Interface Name: ge-7/0/9.0	
Num of rx pkts: 945	Num of tx pkts: 113280890
Num of rx bytes: 56700	Num of tx bytes: 57531319540

show ovssdb virtual-tunnel-end-point

Syntax	show ovssdb virtual-tunnel-end-point address <ip-address> encapsulation <encapsulation-type>
Release Information	Command introduced in Junos OS Release 14.1R2. Command introduced in Junos OS Release 14.1X53-D10 for QFX Series switches. Command introduced in Junos OS Release 14.2 for EX Series switches.
Description	Display information about the following entities that the Juniper Networks device has learned: <ul style="list-style-type: none"> • Other Juniper Networks devices that function as hardware virtual tunnel endpoints (VTEPs) • Software VTEPs • Service nodes • Top-of-rack service nodes (TSNs)
Options	none —Display information about all VTEPs, service nodes, and TSNs that the Juniper Networks device has learned. address ip-address —Display information about the entity with the specified IP address. encapsulation encapsulation-type —Display information about all entities with the specified encapsulation type.
Required Privilege Level	admin
List of Sample Output	show ovssdb virtual-tunnel-end-point on page 116 show ovssdb virtual-tunnel-end-point address (Specific Address) on page 116 show ovssdb virtual-tunnel-end-point encapsulation (Specific Encapsulation) on page 116 show ovssdb virtual-tunnel-end-point address (Specific Address) encapsulation (Specific Encapsulation) on page 116
Output Fields	Table 24 on page 115 lists the output fields for the show ovssdb virtual-tunnel-end-point command. Output fields are listed in the approximate order in which they appear.

Table 24: show ovssdb virtual-tunnel-end-point Output Fields

Field Name	Field Description
Encapsulation	Encapsulation type of entity.
IP Address	IP address of entity.
Num of MACs	Number of media access control (MAC) addresses learned by the entity.

Sample Output

show ovssdb virtual-tunnel-end-point

```

user@host> show ovssdb virtual-tunnel-end-point
Encapsulation      Ip Address      Num of MAC's
VXLAN over IPv4    10.255.181.43   24
VXLAN over IPv4    10.255.181.50   12
VXLAN over IPv4    10.255.181.72   24

```

show ovssdb virtual-tunnel-end-point address (Specific Address)

```

user@host> show ovssdb virtual-tunnel-end-point address 10.255.181.43
Encapsulation      Ip Address      Num of MAC's
VXLAN over IPv4    10.255.181.43   24

```

show ovssdb virtual-tunnel-end-point encapsulation (Specific Encapsulation)

```

user@host> show ovssdb virtual-tunnel-end-point encapsulation vxlan-over-ipv4
Encapsulation      Ip Address      Num of MAC's
VXLAN over IPv4    10.255.181.43   24
VXLAN over IPv4    10.255.181.50   12
VXLAN over IPv4    10.255.181.72   24

```

show ovssdb virtual-tunnel-end-point address (Specific Address) encapsulation (Specific Encapsulation)

```

user@host> show ovssdb virtual-tunnel-end-point address 10.255.181.43 encapsulation
vxlan-over-ipv4
Encapsulation      Ip Address      Num of MAC's
VXLAN over IPv4    10.255.181.43   24

```

show vpls mac-table

Syntax	<pre>show vpls mac-table <brief detail extensive summary> <bridge-domain <i>bridge-domain-name</i>> <instance <i>instance-name</i>> <interface <i>interface-name</i>> <logical-system (all <i>logical-system-name</i>)> <mac-address> <vlan-id <i>vlan-id-number</i>></pre>
Release Information	<p>Command introduced in Junos OS Release 8.5.</p> <p>Command introduced in Junos OS Release 15.1</p>
Description	Display learned virtual private LAN service (VPLS) media access control (MAC) address information.
Options	<p>none—Display all learned VPLS MAC address information.</p> <p>brief detail extensive summary—(Optional) Display the specified level of output.</p> <p>bridge-domain <i>bridge-domain-name</i>—(Optional) Display learned VPLS MAC addresses for the specified bridge domain.</p> <p>instance <i>instance-name</i>—(Optional) Display learned VPLS MAC addresses for the specified instance.</p> <p>interface <i>interface-name</i>—(Optional) Display learned VPLS MAC addresses for the specified instance.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Display learned VPLS MAC addresses for all logical systems or for the specified logical system.</p> <p>mac-address—(Optional) Display the specified learned VPLS MAC address information..</p> <p>vlan-id <i>vlan-id-number</i>—(Optional) Display learned VPLS MAC addresses for the specified VLAN.</p>
Required Privilege Level	view
List of Sample Output	<p>show vpls mac-table on page 118</p> <p>show vpls mac-table (with Layer 2 Services over GRE Interfaces) on page 119</p> <p>show vpls mac-table (with VXLAN enabled) on page 119</p> <p>show vpls mac-table count on page 119</p> <p>show vpls mac-table detail on page 120</p> <p>show vpls mac-table extensive on page 120</p>
Output Fields	<p>Table 25 on page 118 describes the output fields for the show vpls mac-table command. Output fields are listed in the approximate order in which they appear.</p>

Table 25: show vpls mac-table Output fields

Field Name	Field Description
Routing instance	Name of the routing instance.
Bridging domain	Name of the bridging domain.
MAC address	MAC address or addresses learned on a logical interface.
MAC flags	Status of MAC address learning properties for each interface: <ul style="list-style-type: none"> • S—Static MAC address configured. • D—Dynamic MAC address learned. • SE—MAC accounting is enabled. • NM—Nonconfigured MAC.
Logical interface	Name of the logical interface.
MAC count	Number of MAC addresses learned on a specific routing instance or interface.
Learning interface	Logical interface or logical Label Switched Interface (LSI) the address is learned on.
Base learning interface	Base learning interface of the MAC address. This field is introduced in Junos OS Release 14.2.
Learn VLAN ID/VLAN	VLAN ID of the routing instance or bridge domain in which the MAC address was learned.
VXLAN ID/VXLAN	VXLAN Network Identifier (VNI)
Layer 2 flags	Debugging flags signifying that the MAC address is present in various lists.
Epoch	Spanning Tree Protocol epoch number identifying when the MAC address was learned. Used for debugging.
Sequence number	Sequence number assigned to this MAC address. Used for debugging.
Learning mask	Mask of Packet Forwarding Engines where this MAC address was learned. Used for debugging.
IPC generation	Creation time of the logical interface when this MAC address was learned. Used for debugging.

Sample Output

show vpls mac-table

```

user@host> show vpls mac-table
MAC flags (S -static MAC, D -dynamic MAC,
          SE -Statistics enabled, NM -Non configured MAC)

Routing instance : vpls_ldp1
VLAN : 223
  MAC          MAC          Logical
  address      flags        interface
  00:90:69:9c:1c:5d  D          ge-0/2/5.400

```

MAC flags (S -static MAC, D -dynamic MAC,
SE -Statistics enabled, NM -Non configured MAC)

Routing instance : vpls_red
VLAN : 401

MAC address	MAC flags	Logical interface
00:00:aa:12:12:12	D	lsi.1051138
00:05:85:74:9f:f0	D	lsi.1051138

show vpls mac-table (with Layer 2 Services over GRE Interfaces)

user@host> show vpls mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls_4site:1000
Bridging domain : __vpls_4site:1000__, MAC

address	flags	MAC	Logical interface
00:01:01:00:01:f4	D,SE		ge-4/2/0.1000
00:02:01:33:01:f4	D,SE		lsi.1052004
00:03:00:32:01:f4	D,SE		lsi.1048840
00:04:00:14:01:f4	D,SE		lsi.1052005
00:02:01:33:02:f7	D,SE		gr-1/2/10.10

show vpls mac-table (with VXLAN enabled)

user@host> show vpls mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls_4site:1000
Bridging domain : __vpls_4site:1000__, VLAN : 4094,4093
VXLAN: Id : 300, Multicast group: 226.1.1.3

MAC address	MAC flags	Logical interface
00:01:01:00:01:f4	D,SE	ge-4/2/0.1000
00:02:01:33:01:f4	D,SE	lsi.1052004
00:03:00:32:01:f4	D,SE	lsi.1048840
00:04:00:14:01:f4	D,SE	lsi.1052005
00:02:01:33:02:f7	D,SE	vtep.1052010
00:04:00:14:02:f7	D,SE	vtep.1052011

show vpls mac-table count

user@host> show vpls mac-table count
0 MAC address learned in routing instance __juniper_private1__

MAC address count per interface within routing instance:

Logical interface	MAC count
lc-0/0/0.32769	0
lc-0/1/0.32769	0
lc-0/2/0.32769	0
lc-2/0/0.32769	0
lc-0/3/0.32769	0
lc-2/1/0.32769	0
lc-9/0/0.32769	0
lc-11/0/0.32769	0
lc-2/2/0.32769	0
lc-9/1/0.32769	0
lc-11/1/0.32769	0

1c-2/3/0.32769	0
1c-9/2/0.32769	0
1c-11/2/0.32769	0
1c-11/3/0.32769	0
1c-9/3/0.32769	0

MAC address count per learn VLAN within routing instance:

Learn VLAN ID	MAC count
0	0

1 MAC address learned in routing instance vpls_ldp1

MAC address count per interface within routing instance:

Logical interface	MAC count
lsi.1051137	0
ge-0/2/5.400	1

MAC address count per learn VLAN within routing instance:

Learn VLAN ID	MAC count
0	1

1 MAC address learned in routing instance vpls_red

MAC address count per interface within routing instance:

Logical interface	MAC count
ge-0/2/5.300	1

MAC address count per learn VLAN within routing instance:

Learn VLAN ID	MAC count
0	1

show vpls mac-table detail

```

user@host> show vpls mac-table detail
MAC address: 00:90:69:9c:1c:5d
Routing instance: vpls_ldp1
Learning interface: ge-0/2/5.400
Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
Epoch: 0                               Sequence number: 1
Learning mask: 0x1                       IPC generation: 0

MAC address: 00:90:69:9c:1c:5d
Routing instance: vpls_red
Learning interface: ge-0/2/5.300
Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
Epoch: 0                               Sequence number: 1
Learning mask: 0x1                       IPC generation: 0

```

show vpls mac-table extensive

```

user@host> show vpls mac-table extensive

MAC address: 00:10:00:01:00:00
Routing instance: vpls_1
Bridging domain: __vpls_1__, VLAN : NA
Learning interface: lsi.1049165
Base learning interface: lsi.1049165
Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
Epoch: 0                               Sequence number: 1
Learning mask: 0x00000001

```



```
MAC address: 00:10:00:01:00:01
Routing instance: vpls_1
  Bridging domain: __vpls_1__, VLAN : NA
  Learning interface: lsi.1049165
  Base learning interface: lsi.1049165
  Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
  Epoch: 0                               Sequence number: 1
  Learning mask: 0x00000001

MAC address: 00:10:00:01:00:02
Routing instance: vpls_1
  Bridging domain: __vpls_1__, VLAN : NA
  Learning interface: lsi.1049165
  Base learning interface: lsi.1049165
  Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
  Epoch: 0                               Sequence number: 1
  Learning mask: 0x00000001

MAC address: 00:10:00:01:00:03
Routing instance: vpls_1
  Bridging domain: __vpls_1__, VLAN : NA
  Learning interface: lsi.1049165
  Base learning interface: lsi.1049165
  Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
  Epoch: 0                               Sequence number: 1
  Learning mask: 0x00000001
```


CHAPTER 8

VXLAN Operational Commands

- `show bridge mac-table`
- `show vpls mac-table`

show bridge mac-table

Syntax	<pre>show bridge mac-table <brief count detail extensive> <bridge-domain (all <i>bridge-domain-name</i>)> <global-count> <interface <i>interface-name</i>> <mac-address> <vlan-id (all-vlan <i>vlan-id</i>)></pre>
Release Information	<p>Command introduced in Junos OS Release 8.4.</p> <p>Command introduced in Junos OS Release 15.1.</p>
Description	(MX Series routers only) Display Layer 2 media access control (MAC) address information.
Options	<p>none—Display all learned Layer 2 MAC address information.</p> <p>brief count detail extensive—(Optional) Display the specified level of output.</p> <p>bridge-domain (all <i>bridge-domain-name</i>)—(Optional) Display learned Layer 2 MAC addresses for all bridging domains or for the specified bridging domain.</p> <p>global-count—(Optional) Display the total number of learned Layer 2 MAC addresses on the system.</p> <p>instance <i>instance-name</i>—(Optional) Display learned Layer 2 MAC addresses for the specified routing instance.</p> <p>interface <i>interface-name</i>—(Optional) Display learned Layer 2 MAC addresses for the specified interface.</p> <p>mac-address—(Optional) Display the specified learned Layer 2 MAC address information.</p> <p>vlan-id (all-vlan <i>vlan-id</i>)—(Optional) Display learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.</p>
Additional Information	When Layer 2 protocol tunneling is enabled, the tunneling MAC address 01:00:0c:cd:cd:d0 is installed in the MAC table. When the Cisco Discovery Protocol (CDP), Spanning Tree Protocol (STP), or VLAN Trunk Protocol (VTP) is configured for Layer 2 protocol tunneling on an interface, the corresponding protocol MAC address is installed in the MAC table.
Required Privilege Level	view
List of Sample Output	<p>show bridge mac-table on page 126</p> <p>show bridge mac-table (with Layer 2 Services over GRE Interfaces) on page 126</p> <p>show bridge mac-table (with VXLAN Enabled) on page 126</p> <p>show bridge mac-table count on page 127</p> <p>show bridge mac-table detail on page 127</p> <p>show bridge mac-table instance pbb-evpn on page 128</p>

Output Fields Table 26 on page 125 describes the output fields for the **show bridge mac-table** command. Output fields are listed in the approximate order in which they appear.

Table 26: show bridge mac-table Output Fields

Field Name	Field Description
Routing instance	Name of the routing instance.
Bridging domain	Name of the bridging domain.
MAC address	MAC address or addresses learned on a logical interface.
MAC flags	Status of MAC address learning properties for each interface: <ul style="list-style-type: none"> • S—Static MAC address is configured. • D—Dynamic MAC address is configured. • L—Locally learned MAC address is configured. • C—Control MAC address is configured. • SE—MAC accounting is enabled. • NM—Non-configured MAC. • R—Remote PE MAC address is configured.
Logical interface	Name of the logical interface.
MAC count	Number of MAC addresses learned on the specific routing instance or interface.
Learning interface	Name of the logical interface on which the MAC address was learned.
Learning VLAN	VLAN ID of the routing instance or bridge domain in which the MAC address was learned.
VXLAN ID/VXLAN	VXLAN Network Identifier (VNI).
Layer 2 flags	Debugging flags signifying that the MAC address is present in various lists.
Epoch	Spanning Tree Protocol epoch number identifying when the MAC address was learned. Used for debugging.
Sequence number	Sequence number assigned to this MAC address. Used for debugging.
Learning mask	Mask of the Packet Forwarding Engines where this MAC address was learned. Used for debugging.
IPC generation	Creation time of the logical interface when this MAC address was learned. Used for debugging.

Sample Output

show bridge mac-table

```
user@host> show bridge mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : default-switch
Bridging domain : test1, VLAN : 1
  MAC          MAC      Logical   NH      RTR
  address      flags    interface Index   ID
01:00:0c:cc:cc:cc S,NM    NULL
01:00:0c:cc:cc:cd S,NM    NULL
01:00:0c:cd:cd:d0 S,NM    NULL
64:87:88:6a:17:d0 D        ae0.1
64:87:88:6a:17:f0 D        ae0.1
```

show bridge mac-table (with Layer 2 Services over GRE Interfaces)

```
user@host> show bridge mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : default-switch
Bridging domain : vlan-1, VLAN : 1
  MAC          MAC      Logical   NH      RTR
  address      flags    interface Index   ID
00:01:01:00:01:f7 D,SE    gr-1/2/10.0
00:03:00:32:01:f7 D,SE    gr-1/2/10.0
00:00:21:11:11:10 DL        ge-1/0/0.0
00:00:21:11:11:11 DL        ge-1/1/0.0

Routing instance : default-switch
Bridging domain : vlan-2, VLAN : 2
  MAC          MAC      Logical   NH      RTR
  address      flags    interface Index   ID
00:02:01:33:01:f7 D,SE    gr-1/2/10.1
00:00:21:11:21:10 DL        ge-1/0/0.1
00:00:21:11:21:11 DL        ge-1/1/0.1
```

show bridge mac-table (with VXLAN Enabled)

```
user@host> show bridge mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : default-switch
Bridging domain : vlan-1, VLAN : 1
VXLAN: Id : 100, Multicast group: 226.1.1.1
  MAC          MAC      Logical   NH      RTR
  address      flags    interface Index   ID
00:01:01:00:01:f7 D,SE    vtep.1052010
00:03:00:32:01:f7 D,SE    vtep.1052011
00:00:21:11:11:10 DL        ge-1/0/0.0
00:00:21:11:11:11 DL        ge-1/1/0.0
```

```

Routing instance : default-switch
Bridging domain : vlan-2, VLAN : 2, VXLAN : 200
VXLAN: Id : 200, Multicast group: 226.1.1.2
MAC          MAC          Logical
address      flags       interface
00:02:01:33:01:f7 D,SE    vtep.1052010
00:04:00:14:01:f7 D,SE    vtep.1052011
00:00:21:11:21:10 DL       ge-1/0/0.1
00:00:21:11:21:11 DL       ge-1/1/0.1

```

show bridge mac-table count

```

user@host> show bridge mac-table count
2 MAC address learned in routing instance vs1 bridge domain vlan100

MAC address count per interface within routing instance:
Logical interface      MAC count
ge-11/0/3.0           1
ge-11/1/4.100         0
ge-11/1/1.100         0
ge-11/1/0.100         0
xe-10/2/0.100         1
xe-10/0/0.100         0

MAC address count per learn VLAN within routing instance:
Learn VLAN ID         MAC count
0                     2

0 MAC address learned in routing instance vs1 bridge domain vlan200

MAC address count per interface within routing instance:
Logical interface      MAC count
ge-11/1/0.200         0
ge-11/1/1.200         0
ge-11/1/4.200         0
xe-10/0/0.200         0
xe-10/2/0.200         0

MAC address count per learn VLAN within routing instance:
Learn VLAN ID         MAC count
0                     0

```

show bridge mac-table detail

```

user@host> show bridge mac-table detail
MAC address: 00:00:00:19:1c:db
Routing instance: vs1
Bridging domain: vlan100
Learning interface: ge-11/0/3.0   Learning VLAN: 0
Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
Epoch: 4                         Sequence number: 0
Learning mask: 0x800              IPC generation: 0

MAC address: 00:00:00:59:3a:2f
Routing instance: vs1
Bridging domain: vlan100
Learning interface: xe-10/2/0.100 Learning VLAN: 0
Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
Epoch: 7                         Sequence number: 0
Learning mask: 0x400              IPC generation: 0

```

show bridge mac-table instance pbb-evpn

```
user@host> show bridge mac-table instance pbb-evpn
Routing instance : pbb-evpn
Bridging domain : isid-bd10000, ISID : 10000
MAC          MAC      Logical      NH      RTR
address      flags   interface   Index   ID
00:19:e2:b0:76:eb  D      cbp.1000
aa:bb:cc:dd:ee:f2  DC
aa:bb:cc:dd:ee:f3  DC          1048576 1048576
                1048575 1048575
```


show vpls mac-table

Syntax	<pre>show vpls mac-table <brief detail extensive summary> <bridge-domain <i>bridge-domain-name</i>> <instance <i>instance-name</i>> <interface <i>interface-name</i>> <logical-system (all <i>logical-system-name</i>)> <mac-address> <vlan-id <i>vlan-id-number</i>></pre>
Release Information	<p>Command introduced in Junos OS Release 8.5.</p> <p>Command introduced in Junos OS Release 15.1</p>
Description	Display learned virtual private LAN service (VPLS) media access control (MAC) address information.
Options	<p>none—Display all learned VPLS MAC address information.</p> <p>brief detail extensive summary—(Optional) Display the specified level of output.</p> <p>bridge-domain <i>bridge-domain-name</i>—(Optional) Display learned VPLS MAC addresses for the specified bridge domain.</p> <p>instance <i>instance-name</i>—(Optional) Display learned VPLS MAC addresses for the specified instance.</p> <p>interface <i>interface-name</i>—(Optional) Display learned VPLS MAC addresses for the specified instance.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Display learned VPLS MAC addresses for all logical systems or for the specified logical system.</p> <p>mac-address—(Optional) Display the specified learned VPLS MAC address information..</p> <p>vlan-id <i>vlan-id-number</i>—(Optional) Display learned VPLS MAC addresses for the specified VLAN.</p>
Required Privilege Level	view
List of Sample Output	<p>show vpls mac-table on page 130</p> <p>show vpls mac-table (with Layer 2 Services over GRE Interfaces) on page 131</p> <p>show vpls mac-table (with VXLAN enabled) on page 131</p> <p>show vpls mac-table count on page 131</p> <p>show vpls mac-table detail on page 132</p> <p>show vpls mac-table extensive on page 132</p>
Output Fields	<p>Table 25 on page 118 describes the output fields for the show vpls mac-table command. Output fields are listed in the approximate order in which they appear.</p>

Table 27: show vpls mac-table Output fields

Field Name	Field Description
Routing instance	Name of the routing instance.
Bridging domain	Name of the bridging domain.
MAC address	MAC address or addresses learned on a logical interface.
MAC flags	Status of MAC address learning properties for each interface: <ul style="list-style-type: none"> • S—Static MAC address configured. • D—Dynamic MAC address learned. • SE—MAC accounting is enabled. • NM—Nonconfigured MAC.
Logical interface	Name of the logical interface.
MAC count	Number of MAC addresses learned on a specific routing instance or interface.
Learning interface	Logical interface or logical Label Switched Interface (LSI) the address is learned on.
Base learning interface	Base learning interface of the MAC address. This field is introduced in Junos OS Release 14.2.
Learn VLAN ID/VLAN	VLAN ID of the routing instance or bridge domain in which the MAC address was learned.
VXLAN ID/VXLAN	VXLAN Network Identifier (VNI)
Layer 2 flags	Debugging flags signifying that the MAC address is present in various lists.
Epoch	Spanning Tree Protocol epoch number identifying when the MAC address was learned. Used for debugging.
Sequence number	Sequence number assigned to this MAC address. Used for debugging.
Learning mask	Mask of Packet Forwarding Engines where this MAC address was learned. Used for debugging.
IPC generation	Creation time of the logical interface when this MAC address was learned. Used for debugging.

Sample Output

show vpls mac-table

```

user@host> show vpls mac-table
MAC flags (S -static MAC, D -dynamic MAC,
          SE -Statistics enabled, NM -Non configured MAC)

Routing instance : vpls_ldp1
VLAN : 223
  MAC          MAC          Logical
  address      flags        interface
00:90:69:9c:1c:5d  D          ge-0/2/5.400

```

MAC flags (S -static MAC, D -dynamic MAC,
SE -Statistics enabled, NM -Non configured MAC)

Routing instance : vpls_red
VLAN : 401

MAC address	MAC flags	Logical interface
00:00:aa:12:12:12	D	lsi.1051138
00:05:85:74:9f:f0	D	lsi.1051138

show vpls mac-table (with Layer 2 Services over GRE Interfaces)

user@host> show vpls mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls_4site:1000
Bridging domain : __vpls_4site:1000__, MAC

address	flags	MAC	Logical interface
00:01:01:00:01:f4	D,SE		ge-4/2/0.1000
00:02:01:33:01:f4	D,SE		lsi.1052004
00:03:00:32:01:f4	D,SE		lsi.1048840
00:04:00:14:01:f4	D,SE		lsi.1052005
00:02:01:33:02:f7	D,SE		gr-1/2/10.10

show vpls mac-table (with VXLAN enabled)

user@host> show vpls mac-table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : vpls_4site:1000
Bridging domain : __vpls_4site:1000__, VLAN : 4094,4093
VXLAN: Id : 300, Multicast group: 226.1.1.3

MAC address	MAC flags	Logical interface
00:01:01:00:01:f4	D,SE	ge-4/2/0.1000
00:02:01:33:01:f4	D,SE	lsi.1052004
00:03:00:32:01:f4	D,SE	lsi.1048840
00:04:00:14:01:f4	D,SE	lsi.1052005
00:02:01:33:02:f7	D,SE	vtep.1052010
00:04:00:14:02:f7	D,SE	vtep.1052011

show vpls mac-table count

user@host> show vpls mac-table count
0 MAC address learned in routing instance __juniper_private1__

MAC address count per interface within routing instance:

Logical interface	MAC count
lc-0/0/0.32769	0
lc-0/1/0.32769	0
lc-0/2/0.32769	0
lc-2/0/0.32769	0
lc-0/3/0.32769	0
lc-2/1/0.32769	0
lc-9/0/0.32769	0
lc-11/0/0.32769	0
lc-2/2/0.32769	0
lc-9/1/0.32769	0
lc-11/1/0.32769	0

1c-2/3/0.32769	0
1c-9/2/0.32769	0
1c-11/2/0.32769	0
1c-11/3/0.32769	0
1c-9/3/0.32769	0

MAC address count per learn VLAN within routing instance:

Learn VLAN ID	MAC count
0	0

1 MAC address learned in routing instance vpls_ldp1

MAC address count per interface within routing instance:

Logical interface	MAC count
lsi.1051137	0
ge-0/2/5.400	1

MAC address count per learn VLAN within routing instance:

Learn VLAN ID	MAC count
0	1

1 MAC address learned in routing instance vpls_red

MAC address count per interface within routing instance:

Logical interface	MAC count
ge-0/2/5.300	1

MAC address count per learn VLAN within routing instance:

Learn VLAN ID	MAC count
0	1

show vpls mac-table detail

user@host> show vpls mac-table detail

MAC address: 00:90:69:9c:1c:5d

Routing instance: vpls_ldp1

Learning interface: ge-0/2/5.400

Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel

Epoch: 0

Sequence number: 1

Learning mask: 0x1

IPC generation: 0

MAC address: 00:90:69:9c:1c:5d

Routing instance: vpls_red

Learning interface: ge-0/2/5.300

Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel

Epoch: 0

Sequence number: 1

Learning mask: 0x1

IPC generation: 0

show vpls mac-table extensive

user@host> show vpls mac-table extensive

MAC address: 00:10:00:01:00:00

Routing instance: vpls_1

Bridging domain: __vpls_1__, VLAN : NA

Learning interface: lsi.1049165

Base learning interface: lsi.1049165

Layer 2 flags: in_hash, in_ifd, in_ifl, in_vlan, in_rtt, kernel, in_ifbd

Epoch: 0

Sequence number: 1

Learning mask: 0x00000001

```
MAC address: 00:10:00:01:00:01
Routing instance: vpls_1
  Bridging domain: __vpls_1__, VLAN : NA
  Learning interface: lsi.1049165
  Base learning interface: lsi.1049165
  Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
  Epoch: 0                               Sequence number: 1
  Learning mask: 0x00000001

MAC address: 00:10:00:01:00:02
Routing instance: vpls_1
  Bridging domain: __vpls_1__, VLAN : NA
  Learning interface: lsi.1049165
  Base learning interface: lsi.1049165
  Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
  Epoch: 0                               Sequence number: 1
  Learning mask: 0x00000001

MAC address: 00:10:00:01:00:03
Routing instance: vpls_1
  Bridging domain: __vpls_1__, VLAN : NA
  Learning interface: lsi.1049165
  Base learning interface: lsi.1049165
  Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
  Epoch: 0                               Sequence number: 1
  Learning mask: 0x00000001
```

