



Junos[®] OS

OVSDB and VXLAN on QFX5100 and QFX10000 Series Switches

Release
15.1



Published: 2015-04-16

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 OVSDb and VXLAN on QFX5100 and QFX10000 Series Switches

15.1

Copyright © 2015, 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	ix
	Documentation and Release Notes	ix
	Supported Platforms	ix
	Using the Examples in This Manual	ix
	Merging a Full Example	x
	Merging a Snippet	x
	Documentation Conventions	xi
	Documentation Feedback	xiii
	Requesting Technical Support	xiii
	Self-Help Online Tools and Resources	xiii
	Opening a Case with JTAC	xiv
Part 1	OVSDB and VXLAN	
Chapter 1	Understanding OVSDB	3
	Open vSwitch Database Support on Juniper Networks Devices	3
	Understanding the Junos OS Implementation of VXLAN and OVSDB in a VMware NSX for Multi-Hypervisor Environment for the Data Center	4
	Understanding the Open vSwitch Database Management Protocol Running on Juniper Networks Devices	7
	Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers	8
	Understanding How Layer 2 BUM Traffic and Layer 3 Routed Multicast Traffic Are Handled in VXLANs Managed by OVSDB	9
	Understanding Automatically Configured Virtual Extensible LANs in an Open vSwitch Database Environment	10
	Understanding Key Automatic Configuration Events	11
	What the Switch Actually Creates	12
	Automatic Association of an Access Interface to an Automatically Created VXLAN	13
	Automatic Association of a Trunk Interface to an Automatically Created VXLAN	14
	Open vSwitch Database Schema for Physical Devices	15

Chapter 2	Configuring OVSDB and VXLAN	19
	OVSDB and VXLAN Configuration Workflows	19
	Workflow for Juniper Networks Devices that Support the Automatic Configuration of OVSDB-Managed VXLANs	20
	Workflow for Juniper Networks Devices that Support the Manual Configuration of OVSDB-Managed VXLANs	21
	Installing Open vSwitch Database Components on Juniper Networks Devices . .	22
	Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers	23
	Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices	24
	VMware NSX Configuration for Juniper Networks Devices That Function as Virtual Tunnel Endpoints	26
	Creating a Gateway	27
	Creating a Gateway Service	27
	Creating a Logical Switch Port	28
	Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections Between Virtual and Physical Entities in a Data Center (Access Interfaces)	29
	Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections Between Virtual and Physical Entities in a Data Center (Trunk Interfaces)	37
Chapter 3	OVSDB Configuration Statements	49
	controller (OVSDB)	50
	inactivity-probe-duration	51
	interfaces (OVSDB)	52
	maximum-backoff-duration	53
	ovsdb	54
	ovsdb-managed	55
	port (OVSDB)	56
	protocol (OVSDB)	57
	traceoptions (OVSDB)	58
Chapter 4	OVSDB Monitoring Commands	61
	show ovsdb controller	62
	show ovsdb interface	65
	show ovsdb logical-switch	67
	show ovsdb mac	69
	show ovsdb statistics interface	73
	show ovsdb virtual-tunnel-end-point	75
	Verifying that a VMware NSX Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN Are Working Properly	76
Chapter 5	Troubleshooting OVSDB	79
	Troubleshooting a Nonoperational VMware NSX Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN	79

List of Figures

Part 1	OVSDB and VXLAN	
Chapter 1	Understanding OVSDB	3
	Figure 1: High-Level NSX for Multi-Hypervisor Architecture	5
	Figure 2: Integration of Juniper Networks Device That Implements VXLAN and OVSDb into NSX for Multi-Hypervisor Environment	6
Chapter 2	Configuring OVSDB and VXLAN	19
	Figure 3: VXLAN/OVSDB Layer 2 Gateway Topology	31
	Figure 4: VXLAN/OVSDB Layer 2 Gateway Topology	39

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	xi
	Table 2: Text and Syntax Conventions	xi
Part 1	OVSDB and VXLAN	
Chapter 1	Understanding OVSDB	3
	Table 3: OVSDB Support on Juniper Networks Devices	3
	Table 4: NSX for Multi-Hypervisor Components and Products That Can Be Implemented	4
	Table 5: Key Automatic Configuration Events	11
	Table 6: OVSDB Schema Tables	15
Chapter 2	Configuring OVSDB and VXLAN	19
	Table 7: Juniper Networks Devices Support of Automatic and Manual Configurations	19
	Table 8: OVSDB Configuration Workflow for Juniper Networks Devices that Support the Automatic Configuration of OVSDB-Managed VXLANs	20
	Table 9: OVSDB Configuration Workflow for Juniper Networks Devices that Support the Manual Configuration of OVSDB-Managed VXLANs	22
	Table 10: Create a Gateway in NSX Manager: Key Configurations	27
	Table 11: Create a Gateway Service in NSX Manager: Key Configurations	28
	Table 12: Create a Logical Switch Port in NSX Manager: Key Configurations	29
	Table 13: NSX Manager and Junos OS Entities That Must Be Configured	31
	Table 14: Components Configured on Juniper Networks Switch that Functions as Hardware VTEP	32
	Table 15: NSX Manager and Junos OS Entities That Must Be Configured	40
	Table 16: NSX Manager Configurations, and Resulting Automatically Configured by Juniper Networks Switch	41
	Table 17: Components Configured on Juniper Networks Switch that Functions as a Hardware VTEP	42
Chapter 4	OVSDB Monitoring Commands	61
	Table 18: show ovsdb controller Output Fields	63
	Table 19: show ovsdb interface Output Fields	65
	Table 20: show ovsdb logical-switch Output Fields	68
	Table 21: show ovsdb mac Output Fields	70
	Table 22: show ovsdb statistics interface Output Fields	73
	Table 23: show ovsdb virtual-tunnel-end-point Output Fields	75

About the Documentation

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

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:

- QFabric System
- QFX Series standalone switches

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 the *CLI User Guide*.

Documentation Conventions

Table 1 on page xi 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 xi 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.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none">Introduces or emphasizes important new terms.Identifies guide names.Identifies RFC and Internet draft titles.	<ul style="list-style-type: none">A policy <i>term</i> is a named structure that defines match conditions and actions.<i>Junos OS CLI User Guide</i>RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name domain-name
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none">To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level.The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [community-ids]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
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 at the Juniper Networks Technical Documentation 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 <https://www.juniper.net/cgi-bin/docbugreport/>.
- 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 JNASC 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

OVSDB and VXLAN

- [Understanding OVSDB on page 3](#)
- [Configuring OVSDB and VXLAN on page 19](#)
- [OVSDB Configuration Statements on page 49](#)
- [OVSDB Monitoring Commands on page 61](#)
- [Troubleshooting OVSDB on page 79](#)

CHAPTER 1

Understanding OVSDB

- [Open vSwitch Database Support on Juniper Networks Devices on page 3](#)
- [Understanding the Junos OS Implementation of VXLAN and OVSDB in a VMware NSX for Multi-Hypervisor Environment for the Data Center on page 4](#)
- [Understanding the Open vSwitch Database Management Protocol Running on Juniper Networks Devices on page 7](#)
- [Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)
- [Understanding How Layer 2 BUM Traffic and Layer 3 Routed Multicast Traffic Are Handled in VXLANs Managed by OVSDB on page 9](#)
- [Understanding Automatically Configured Virtual Extensible LANs in an Open vSwitch Database Environment on page 10](#)
- [Open vSwitch Database Schema for Physical Devices on page 15](#)

Open vSwitch Database Support on Juniper Networks Devices

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

[Table 3 on page 3](#) lists the Juniper Networks devices that support the Open vSwitch Database (OVSDB) management protocol. To run OVSDB on one of these devices, the device must have a Junos OS software package and a VMware NSX install package (from Juniper Networks) installed. The release number for both software packages must be the same.

For each device, the table also includes the Junos OS release in which OVSDB support is introduced and the VMware NSX install package.

Table 3: OVSDB Support on Juniper Networks Devices

Juniper Networks Device	Junos OS Release In Which OVSDB Support Introduced	VMware NSX Install Package
MX80 3D Universal Edge Router	14.1R2	jsdn-powerpc-release
MX240, MX480, MX960 3D Universal Edge Routers	14.1R2	jsdn-i386-release
QFX5100 Ethernet Switch	14.1X53-D10	jsdn-i386-release

Table 3: OVSDB Support on Juniper Networks Devices (*continued*)

Juniper Networks Device	Junos OS Release In Which OVSDB Support Introduced	VMware NSX Install Package
QFX10002 Ethernet Switch	15.1X53-D10	jsdn-i386-release
EX9200 Ethernet Switch	14.2	jsdn-i386-release

Related Documentation • [Installing Open vSwitch Database Components on Juniper Networks Devices on page 22](#)

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

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

Some Juniper Networks devices support Virtual Extensible LAN (VXLAN) and the Open vSwitch Database (OVSDB) management protocol. (For information about the Juniper Networks devices on which OVSDB is supported and the Juniper Networks Junos operating system (Junos OS) release in which support is introduced, see [“Open vSwitch Database Support on Juniper Networks Devices” on page 3](#).) 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 4 on page 4](#) outlines the components that compose this environment and products that are typically deployed for each component.

Table 4: NSX for Multi-Hypervisor Components and Products That Can Be Implemented

Component	Products
Cloud management platform (CMP)	CloudStack OpenStack Custom CMP
Network virtualization platform	NSX for Multi-Hypervisor
Hypervisor	Kernel-based Virtual Machine (KVM) Red Hat VMware ESXi Xen
NOTE: Juniper Networks supports only KVM and ESXi.	

Table 4: NSX for Multi-Hypervisor Components and Products That Can Be Implemented *(continued)*

Component	Products
Virtual switch	Open vSwitch (OVS) NSX vSwitch
SDN controller	NSX for Multi-Hypervisor controller
Overlay protocol	VXLAN
MAC learning protocol	OVSDb

Figure 1 on page 5 shows a high-level view of the architecture into which the NSX for Multi-Hypervisor platform fits, while Figure 2 on page 6 provides a more detailed representation of the components in the virtual and physical networks.

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

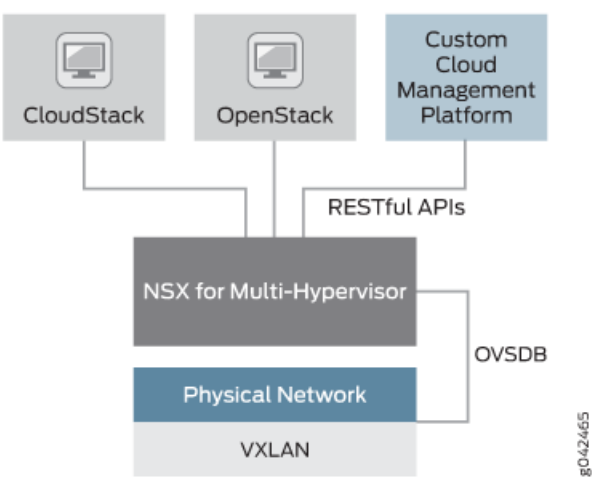
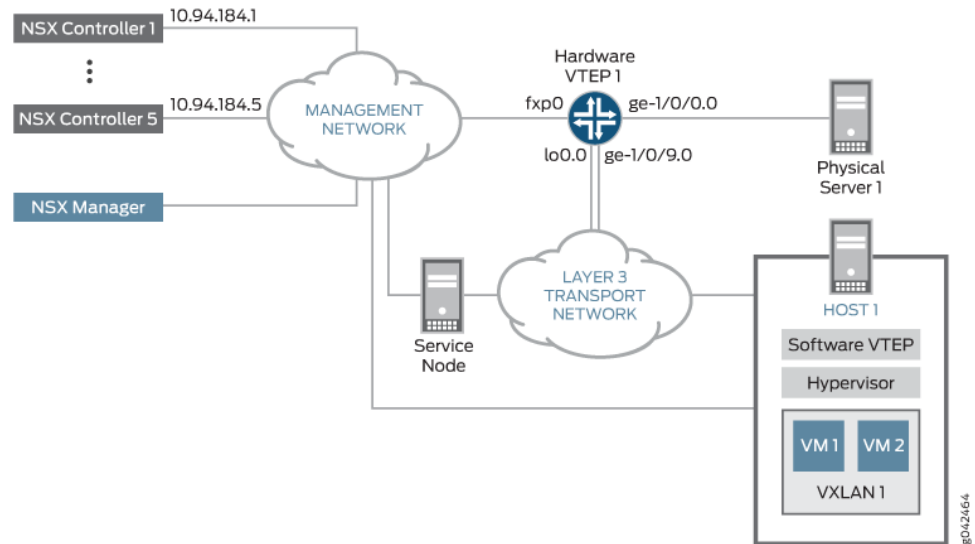


Figure 2: Integration of Juniper Networks Device That Implements VXLAN and OVSDB into NSX for Multi-Hypervisor Environment



In the data center topology shown in [Figure 2 on page 6](#), 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 enables 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 hardware VTEP in [Figure 2 on page 6](#) 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 Open vSwitch Database Management Protocol Running on Juniper Networks Devices on page 7](#)
- [Open vSwitch Database Schema for Physical Devices](#)
- [Open vSwitch Database Schema for Physical Devices on page 15](#)

Understanding the Open vSwitch Database Management Protocol Running on Juniper Networks Devices

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

The Juniper Networks Junos operating system (Junos OS) implementation of the Open vSwitch Database (OVSDDB) management protocol provides a means through which VMware NSX controllers and Juniper Networks devices that support OVSDDB can communicate. In an NSX multi-hypervisor environment, NSX controllers and Juniper Networks devices exchange control and statistical information, thereby enabling virtual machine (VM) traffic from entities in a virtual network to be forwarded to entities in a physical network and vice versa.

To enable communication between NSX controllers and Juniper Networks devices, the Junos OS implementation of OVSDDB includes an OVSDDB server and an OVSDDB client, both of which run on each Juniper Networks device that supports OVSDDB.

The OVSDDB server on a Juniper Networks device can communicate with an OVSDDB client on one or more NSX controllers. To establish a connection between a Juniper Networks device and an NSX controller, you must specify information about the 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 NSX controller port that you specify in the Junos OS configuration.

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

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

Some of the OVSDDB 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.

- Related Documentation**
- [Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)
 - [Open vSwitch Database Schema for Physical Devices](#)

- [Open vSwitch Database Schema for Physical Devices on page 15](#)

Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

The Juniper Networks Junos operating system (Junos OS) implementation of the Open vSwitch Database (OVSDB) management protocol provides a means through which VMware NSX controllers and Juniper Networks devices that support OVSDB can communicate. This implementation of OVSDB supports one cluster of NSX controllers, which includes three or five controllers as recommended by VMware.

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.

A Juniper Networks device exchanges control and statistical data with each NSX controller to which it is connected. Therefore, the benefits of connecting a Juniper Networks device to multiple NSX controllers include redundancy and load-balancing of the controller workload.

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

- IP address of the NSX 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. For more information about the files, including actions you must take to create and install some of the files, see [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers” on page 23](#).

- 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 NSX 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 Open vSwitch Database Management Protocol on Juniper Networks Devices*
- [Understanding the Open vSwitch Database Management Protocol Running on Juniper Networks Devices on page 7](#)

Understanding How Layer 2 BUM Traffic and Layer 3 Routed Multicast Traffic Are Handled in VXLANs Managed by OVSDB

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

The Juniper Networks Junos operating system (Junos OS) implementation of the Open vSwitch Database (OVSDB) management protocol provides a means through which VMware NSX 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.
- (This scenario applies only to MX Series routers and EX9200 switches.) Layer 3 multicast traffic that is received by an integrated routing and bridging (IRB) interface 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 service nodes in the same VXLAN. When this option is used, the table for remote multicast 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 software virtual tunnel endpoints (VTEPs) that host the service nodes.

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 software VTEPs. The software VTEP, and therefore, the service node 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 service node replicates the packet and forwards the replicas to all interfaces within the VXLAN.

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



NOTE: The QFX5100 and QFX10002 switches do not support ingress node replication.

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.

Related Documentation

- *Configuring OVSDB-Managed VXLANs*
- *Example: Setting Up Inter-VXLAN Unicast and Multicast Routing and OVSDB Connections in a Data Center*
- *Open vSwitch Database Schema for Physical Devices*

Understanding Automatically Configured Virtual Extensible LANs in an Open vSwitch Database Environment

Supported Platforms **QFX Series**



NOTE: This topic applies only to QFX5100 and QFX10002 switches, which support the automatic 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 the automatic configuration can begin.

On all other Juniper Networks devices that support OVSDB and VXLAN, you must manually configure OVSDB-managed VXLANs using the Junos OS CLI.

For information about the automatic and manual configuration tasks, see [“OVSDB and VXLAN Configuration Workflows” on page 19](#).

The Juniper Networks Junos operating system (Junos OS) implementation of the OVSDB management protocol provides a means through which VMware NSX controllers and Juniper Networks switches that support OVSDB can communicate. Support for OVSDB enables the switches in a physical network to be integrated into a virtual NSX environment.

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 *Virtual Extensible LAN (VXLAN)*. In an NSX environment, the same concept is known as a *logical switch*.

In general, for each automatically configured VXLAN and its connections to one or more physical servers, also known as *OVSDB-managed interfaces*, there must be an NSX-equivalent logical switch and OVSDB-managed interfaces configured in NSX Manager or in the NSX API. In fact, you must configure the logical switch and OVSDB-managed interfaces in NSX Manager or in the NSX API *before* the VXLAN and OVSDB-managed interfaces are automatically configured by the Juniper Networks device. This configuration task sequence is important because when automatically configuring a VXLAN and OVSDB-managed interfaces, the Juniper Networks switch uses some of the NSX configuration.

The following sections describe the automatic configuration of OVSDB-managed VXLANs:

- [Understanding Key Automatic Configuration Events on page 11](#)
- [What the Switch Actually Creates on page 12](#)

Understanding Key Automatic Configuration Events

The “[OVSDB and VXLAN Configuration Workflows](#)” on [page 19](#) topic provides a high-level workflow for configuring OVSDB on Juniper Networks switches that support the automatic configuration of OVSDB-managed VXLANs. After you complete the workflow, the Juniper Networks switch begins the automatic configuration process.

[Table 5 on page 11](#) describes the sequence of events that result in the automatic configuration of an OVSDB-managed VXLAN and interfaces associated with the VXLAN. Some of the events are tasks from the OVSDB configuration workflow that you perform, and other events are handled by the NSX controller or the Juniper Networks switch.

Table 5: Key Automatic Configuration Events

Sequence	Event	How this Event Impacts the Automatic Configuration of OVSDB-Managed VXLANs
1	You enter the set switch-options ovbdb-managed configuration mode command on the Juniper Networks switch.	Entering this command enables the Juniper Networks switch to automatically configure an OVSDB-managed VXLAN.
2	On the Juniper Networks switch, you configure each interface associated with a VXLAN so that it is managed by OVSDB.	The Juniper Networks switch pushes information about the OVSDB-managed interfaces to the OVSDB schema for physical devices. This schema resides in the Juniper Networks switch.
3	You configure a logical switch for each OVSDB-managed VXLAN that you plan to implement by using NSX Manager or the NSX API.	Based on the name and the VNI that you specify in the logical switch configuration, a UUID is automatically generated.

Table 5: Key Automatic Configuration Events (*continued*)

Sequence	Event	How this Event Impacts the Automatic Configuration of OVSDB-Managed VXLANs
4	The NSX controller pushes information about the logical switch to the OVSDB schema for physical devices.	–
5	<p>Create the following entities in NSX Manager or in the NSX API:</p> <ul style="list-style-type: none"> For each Juniper Networks switch that you deploy as a hardware virtual tunnel endpoint (VTEP), you create a gateway. For each OVSDB-managed interface that you configured in event number 2, you create a gateway service. For each interface that you plan to implement for a VXLAN, configure a logical switch port. 	<p>By performing this task, you essentially do the following:</p> <ul style="list-style-type: none"> Bind the OVSDB-managed interfaces that you created in event number 2 to a gateway. For each interface that you plan to implement for a VXLAN, create an interface and bind it to a logical switch configured in event number 3. <p>The NSX controller pushes the binding information to the Juniper Networks switch.</p>
6	The Juniper Networks switch automatically creates the VXLAN.	<p>The Juniper Networks switch uses relevant information from the logical switch configuration to automatically create the corresponding VXLAN.</p> <p>For the name of the VXLAN, the Juniper Networks switch uses the UUID of the logical switch.</p>
7	Based on the binding information received from the NSX controller, the Juniper Networks switch automatically creates the interfaces configured in event number 2 and associates the interfaces with the VXLAN created in event number 6.	Depending on the interface information received from the NSX controller, the Juniper Networks switch creates either an access interface or trunk interface. For more information, see “What the Switch Actually Creates” on page 12 .

What the Switch Actually Creates

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

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

Note the following about this sample configuration:

- The name of the VXLAN is 28805c1d-0122-495d-85df-19abd647d772, which assumes that the logical switch UUID is 28805c1d-0122-495d-85df-19abd647d772.
- For the virtual network identifier (VNI), the Juniper Networks switch uses the VNI specified in the logical switch configuration—in this case, VNI 100. If the Juniper Networks switch detects that the VNI 100 is a duplicate, that is, it detects another VXLAN with a VNI of 100, and that VNI is configured by manually entering the **set vlans *vlan-name* vxlan vni (1 – 16777214)** command in the Junos OS CLI, the switch deletes the manually configured VXLAN.
- If you need to modify or delete an OVSDB-managed VXLAN that was automatically created by a Juniper Networks switch, you must modify or delete the corresponding logical switch configuration in NSX Manager or in the NSX API. After you modify or

delete the logical switch configuration, the NSX controller pushes the update to the Juniper Networks switch, and the switch modifies or deletes its configuration accordingly. Modifying or deleting the VXLAN configuration by using the Junos OS CLI on the switch does not actually modify or delete the VXLAN.

Depending on the gateway service and logical switch port that you configure using NSX Manager or the NSX API, the Juniper Networks switch automatically creates and associates one or more interfaces with the VXLAN. The configuration generated by the switch depends on whether an interface is an access interface or a trunk interface. The following sections provide information about the configuration that the switch automatically generates for each interface type:

- [Automatic Association of an Access Interface to an Automatically Created VXLAN on page 13](#)
- [Automatic Association of a Trunk Interface to an Automatically Created VXLAN on page 14](#)

Automatic Association of an Access Interface to an Automatically Created VXLAN

In a network that has only a single VXLAN, all packets are untagged. In this situation, an access interface is typically used to connect the Juniper Networks switch to a physical server in an OVSDDB environment. The access interface accepts only untagged packets from the physical server.

To determine the type of interface to create and associate with an OVSDDB-managed VXLAN, the Juniper Networks switch uses the VLAN number that you specified when configuring the logical switch port in NSX Manager or in the NSX API. If you specified **0** as the VLAN number, the switch creates an access interface. (If you specified **1** through **4000** as the VLAN number, the switch creates a trunk interface.)

The following is a sample configuration of an access interface that is automatically created and associated with a VXLAN by a Juniper Networks switch:

```
set interfaces ge-1/0/0 encapsulation ethernet-bridge
set interfaces ge-1/0/0 unit 0
set vlans 28805c1d-0122-495d-85df-19abd647d772 interface ge-1/0/0.0
```

This sample configuration assumes the following:

- On a Juniper Networks switch, Interface ge-1/0/0 was configured as an OVSDDB-managed interface using the Junos OS CLI, and this configuration was pushed to the NSX controller by way of the OVSDDB schema for physical devices.
- In NSX Manager or in the NSX API, a logical switch was created, and that logical switch was assigned the UUID of 28805c1d-0122-495d-85df-19abd647d772.
- In NSX Manager or in the NSX API, a VXLAN Layer 2 gateway service was set up, and interface ge-1/0/0 was specified as a port in the gateway service.
- In NSX Manager or in the NSX API, 0 was specified as the VLAN number in the gateway services configuration.

Automatic Association of a Trunk Interface to an Automatically 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 VLAN ID. In this situation, the interface that connects the Juniper Networks switch to a physical server in an OVSDB environment is typically a trunk interface, and this interface accepts only tagged packets from the physical switch.

To determine the type of interface to create and associate with an OVSDB-managed VXLAN, the Juniper Networks switch uses the VLAN number that you specified when configuring the logical switch port in NSX Manager or in the NSX API. If you specified 1 through 4000 as the VLAN number, the switch creates a trunk interface. (If you specified 0 as the VLAN number, the switch creates an access interface.)

The following is a sample configuration of a trunk interface that is automatically created and associated with a VXLAN by a Juniper Networks switch:

```
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
```

This sample configuration assumes the following:

- On a Juniper Networks switch, Interface ge-1/0/0 was configured as an OVSDB-managed interface using the Junos OS CLI.
- In NSX Manager or in the NSX API, a logical switch was created, and that logical switch was assigned the UUID of 28805c1d-0122-495d-85df-19abd647d772.
- In NSX Manager or in the NSX API, a VXLAN Layer 2 gateway service was set up, and interface ge-1/0/0 was specified as a port in the gateway services configuration
- In NSX Manager or in the NSX API, 10 was specified as the VLAN number in the gateway services configuration.

The sample configuration creates a trunk interface (ge-1/0/0.10) that accepts incoming packets with a VLAN tag of 10 and adds a tag of 100 to each packet to identify the packet 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 Junos OS Implementation of VXLAN and OVSDB in a VMware NSX for Multi-Hypervisor Environment for the Data Center on page 4](#)
- [Understanding the Open vSwitch Database Management Protocol Running on Juniper Networks Devices on page 7](#)
- [show ovssdb logical-switch on page 67](#)
- [Verifying that a VMware NSX Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN Are Working Properly on page 76](#)

Open vSwitch Database Schema for Physical Devices

Supported Platforms [MX Series, QFX Series standalone switches](#)

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 VMware NSX controllers, the connections provide a means through which the Juniper Networks device and the controllers can communicate.

In an NSX multi-hypervisor environment, Juniper Networks devices that support OVSDB and NSX 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 NSX 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 or an NSX controller can collect MAC routes learned by entities in the physical or virtual 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 virtual 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 6 on page 15](#) 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 6: 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 for each NSX controller that is connected to the Juniper Networks device.	<ul style="list-style-type: none"> Juniper Networks device NSX controller 	show ovbdb controller
Physical switch table	Includes information about the Juniper Networks device on which a hardware VTEP is implemented. This table includes information only for the device on which the table resides.	Juniper Networks device	—

Table 6: 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 information about logical switches, which you configured in NSX Manager or in the NSX API, and the corresponding Virtual Extensible LANs (VXLANs), which was configured on the Juniper Networks device.	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.	Juniper Networks device	show ovssdb virtual-tunnel-end-point
Physical locator set table	Includes a list of service nodes for a logical switch.	Juniper Networks device	—
Unicast MACs remote table	Reachability information, including unicast MAC addresses, for entities in the virtual network.	NSX controller	show ovssdb mac
Unicast MACs local table	Reachability information, including unicast MAC addresses, for entities in the physical network.	Juniper Networks device that is configured as a hardware VTEP.	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 that host a cluster of service nodes, which handle multicast traffic.	NSX controller	show ovssdb mac

Table 6: OVSDb Schema Tables (*continued*)

Table Name	Description	Source of Information	Command
Multicast MACs local table	<p>NOTE: Only QFX5100 and QFX10002 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	show ovldb mac

- Related Documentation**
- [Understanding the Open vSwitch Database Management Protocol Running on Juniper Networks Devices on page 7](#)
 - [Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)

CHAPTER 2

Configuring OVSDB and VXLAN

- [OVSDB and VXLAN Configuration Workflows on page 19](#)
- [Installing Open vSwitch Database Components on Juniper Networks Devices on page 22](#)
- [Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers on page 23](#)
- [Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices on page 24](#)
- [VMware NSX Configuration for Juniper Networks Devices That Function as Virtual Tunnel Endpoints on page 26](#)
- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections Between Virtual and Physical Entities in a Data Center \(Access Interfaces\) on page 29](#)
- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections Between Virtual and Physical Entities in a Data Center \(Trunk Interfaces\) on page 37](#)

OVSDB and VXLAN Configuration Workflows

Some Juniper Networks devices that support Open vSwitch Database (OVSDB) and Virtual Extensible LAN (VXLAN) also support the automatic configuration of OVSDB-managed VXLANs. For the remaining Juniper Networks devices, you must manually configure OVSDB-managed VXLANs using the Junos OS CLI. [Table 7 on page 19](#) outlines each device and the configuration method that each device supports.

Table 7: Juniper Networks Devices Support of Automatic and Manual Configurations

Juniper Networks Device	Supports Automatic Configuration of OVSDB-Managed VXLANs?	Supports Manual Configuration of OVSDB-Managed VXLANs?
MX80, MX240, MX480, MX960 routers	No	Yes
EX9200 switches	No	Yes
QFX5100 switches	Yes	No
QFX10002 switches	Yes	No

Because of the different configuration methods, the following configuration workflows exist:

- [Workflow for Juniper Networks Devices that Support the Automatic Configuration of OVSDB-Managed VXLANs on page 20](#)
- [Workflow for Juniper Networks Devices that Support the Manual Configuration of OVSDB-Managed VXLANs on page 21](#)

Workflow for Juniper Networks Devices that Support the Automatic Configuration of OVSDB-Managed VXLANs

Supported Platforms [QFX Series standalone switches](#)

[Table 8 on page 20](#) provides a high-level workflow for the tasks that you must perform to configure OVSDB on Juniper Networks devices that support the automatic configuration of OVSDB-managed VXLANs. You must perform the tasks in [Table 8 on page 20](#) 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 8 on page 20](#).

Table 8: OVSDB Configuration Workflow for Juniper Networks Devices that Support the Automatic Configuration of OVSDB-Managed VXLANs

Task Sequence	Task	Where To Find More Information
1	Install the Juniper Networks VMware NSX software package.	“Installing Open vSwitch Database Components on Juniper Networks Devices” on page 22
2	Create and install a Secure Sockets Layer (SSL) key and certificates.	“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers” on page 23
3	Enter the set switch-options ovsdb-managed configuration mode command on the Juniper Networks switch.	—
4	Explicitly configure a connection to at least one VMware NSX controller.	“Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices” on page 24
5	Specify that each interface associated with a VXLAN is to be managed by OVSDB.	“Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices” on page 24
6	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.

Table 8: OVSDDB Configuration Workflow for Juniper Networks Devices that Support the Automatic Configuration of OVSDDB-Managed VXLANs (*continued*)

Task Sequence	Task	Where To Find More Information
7	<ul style="list-style-type: none"> For each Juniper Network device on which OVSDDB-managed VXLANs and interfaces are configured, create a gateway. For each OVSDDB-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 properly 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 That Function as Virtual Tunnel Endpoints” on page 26</p>

After you successfully complete task 7 in [Table 8 on page 20](#), the Juniper Networks device automatically creates a VXLAN for each logical switch that you configured in task 6. The device also automatically creates and associates interfaces with each VXLAN. The automatically created interface configuration is based on the gateway service and logical switch ports that you configured in task 7. For more information about how the automatic configuration feature works, see [“Understanding Automatically Configured Virtual Extensible LANs in an Open vSwitch Database Environment”](#) on page 10.

For information about the scenarios in which these Juniper Networks devices are commonly deployed, see the following topics:

- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDDB Connections Between Virtual and Physical Entities in a Data Center \(Access Interfaces\)](#) on page 29
- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDDB Connections Between Virtual and Physical Entities in a Data Center \(Trunk Interfaces\)](#) on page 37

Workflow for Juniper Networks Devices that Support the Manual Configuration of OVSDDB-Managed VXLANs

Supported Platforms [EX Series, MX Series](#)

[Table 9 on page 22](#) provides a high-level workflow for the tasks that you must perform to configure OVSDDB on Juniper Networks devices that support the manual configuration of OVSDDB-managed VXLANs. You must perform the tasks in [Table 9 on page 22](#) for each Juniper Networks device that you plan to deploy in an OVSDDB 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 9 on page 22](#).

Table 9: OVSDB Configuration Workflow for Juniper Networks Devices that Support the Manual Configuration of OVSDB-Managed VXLANs

Task Sequence	Task	Where To Find More Information
1	Install the Juniper Networks VMware NSX software package.	“Installing Open vSwitch Database Components on Juniper Networks Devices” on page 22
2	Create and install an SSL key and certificates.	“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers” on page 23
3	Explicitly configure a connection to at least one NSX controller.	Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices
4	Specify that each interface associated with a VXLAN is to be managed by OVSDB.	Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices
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.	Configuring OVSDB-Managed VXLANs
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 properly 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 That Function as Virtual Tunnel Endpoints” on page 26</p>

For information about the 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 VMware NSX Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN Are Working Properly on page 76](#)

Installing Open vSwitch Database Components on Juniper Networks Devices

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

To install Open vSwitch Database (OVSDb) components on a Juniper Networks device, you must copy the Juniper Networks VMware NSX install package to the Juniper Networks device and then install the package. The install package name uses the following format:

`jsdn-packageID-release`

where:

- *packageID* identifies the package that must run on each Juniper Networks device.
- *release* identifies the release; for example, 15.1. The install package release and the Junos OS release running on the device must be the same.

For information about which Juniper Networks devices support OVSDb and the install package for each device, see [“Open vSwitch Database Support on Juniper Networks Devices” on page 3](#).

To install the VMware NSX install package on a Juniper Networks device:

1. Download the software package to the Juniper Networks device.
2. If a VMware NSX install package already exists on the Juniper Networks device, remove the package by issuing the **request system software delete** operational mode command.

```
user@device> request system software delete existing-ovsdb-package
```

3. Install the new software package by using the **request system software add** operational mode command.

```
user@device> request system software add path-to-ovsdb-package
```

Related Documentation

- [Understanding the Open vSwitch Database Management Protocol Running on Juniper Networks Devices on page 7](#)
- [OVSDb and VXLAN Configuration Workflows on page 19](#)

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

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

To secure a connection between a Juniper Networks device that supports the Open vSwitch Database (OVSDb) management protocol and one or more VMware NSX 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 the initial connection between a Juniper Networks device with OVSDB implemented and an NSX controller, the `ca-cert.pem` file is automatically generated, and then installed in the `/var/db/certs` directory on the device.

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 Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)
 - [OVSDB and VXLAN Configuration Workflows on page 19](#)

Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices

Supported Platforms [QFX Series standalone switches](#)

To implement the Open vSwitch Database (OVSDb) management protocol on a Juniper Networks device, you must explicitly configure a connection between the Juniper Networks device and at least one VMware NSX controller, using the Junos OS CLI.

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

You must also specify that any interface implemented for a Virtual Extensible LAN (VXLAN) is managed by OVSDb. These interfaces can be access or trunk interfaces. 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, you enable OVSDb to learn about the other hardware VTEPs and the MAC addresses learned by the hardware VTEPs.

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

- Ensure that the Juniper Networks device has a Juniper Networks VMware NSX software package installed, and that the software package release number is the same as the Junos OS release number running on the device. For more information, see [“Installing Open vSwitch Database Components on Juniper Networks Devices” on page 22](#).
- Create an SSL private key and certificate, and install them in the `/var/db/certs` directory of the Juniper Networks device. For more information, see [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers” on page 23](#).
- Enable the Juniper Networks device to automatically configure OVSDb-managed VXLANs and associated interfaces, enter the **set switch-options ovssdb-managed** configuration mode command on the device.
- Determine the IP address of the NSX controller.

To set up OVSDb on a Juniper Networks device:

1. Specify the IP address of the NSX controller.


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


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


```
[edit protocols ovssdb]
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 ovssdb]
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 NSX controller again if the previous attempt failed.

```
[edit protocols ovssdb]
```

```
user@host# set controller ip-address maximum-backoff-duration milliseconds
```

6. (Optional) Repeat steps 1 through 5 to explicitly configure a connection to an additional NSX controller in the same cluster of controllers.

7. Specify the interfaces that you want OVSDB to manage.

```
[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. For information about the remaining configuration tasks that you must perform in NSX Manager or in the NSX API, see [“OVSDB and VXLAN Configuration Workflows” on page 19](#).

**Related
Documentation**

- [Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)

VMware NSX Configuration for Juniper Networks Devices That Function as Virtual Tunnel Endpoints

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

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

- For each Juniper Networks device on which OVSDB-managed VXLANs and interfaces are configured, you must create an NSX-equivalent entity, which is known as a *gateway*.
- For each OVSDB-managed 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.

Performing the configurations described in this topic enables 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 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 to ensure 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 27](#)
- [Creating a Gateway Service on page 27](#)
- [Creating a Logical Switch Port on page 28](#)

Creating a Gateway

In NSX Manager, you must create a gateway for each Juniper Networks device on which OVSDDB-managed VXLANs and interfaces are configured. [Table 10 on page 27](#) provides a summary of key configuration fields in NSX Manager and how to configure them when creating a gateway.

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

NSX Manager Configuration Page/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 OVSDDB-managed interface that you configure on a Juniper Networks device. Creating a gateway service essentially does the following for each OVSDDB-managed 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 27](#).

This section provides a summary of key configuration fields in NSX Manager and how to configure them when creating a gateway service.

Before you start this task, make sure that you have configured the following:

- A gateway for the Juniper Networks device on which the OVSDB-managed interfaces are configured. For information, see [“Creating a Gateway” on page 27](#).
- The OVSDB-managed interfaces on the Juniper Networks device. For information about configuring OVSDB-managed interfaces on Juniper Networks devices that support the automatic configuration of VXLANs, see [“Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices” on page 24](#). For information about configuring OVSDB-managed interfaces on Juniper Networks devices that support the manual configuration of VXLANs, see *Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices*.

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

NSX Manager Configuration Page/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 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:

- Specifies a gateway service—for example, a VTEP Layer 2 gateway service.
- Binds the logical interface 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 27](#).

This section provides a summary of key configuration fields in NSX Manager and how to configure them when creating a logical switch port.

Before you start this task, you must configure the following:

- A logical switch that corresponds to the VXLAN with which the logical interface 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 interface with which the logical interface is associated. For information about configuring a gateway service, see [“Creating a Gateway Service” on page 27](#).

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

NSX Manager Configuration Page/Dialog Box	NSX Manager Configuration Field	How to Configure
Logical Switch	Logical Switch UUID	Select the UUID of a logical switch.
Attachment	Attachment Type	Select VTEP L2 Gateway .
Attachment	VTEP L2 Gateway Service UUID	Select the UUID of a gateway service.
Attachment	VLAN	To specify the port as an access interface, select 0 . To specify the port as a trunk interface, select 1 through 4000 .

Related Documentation • [OVSDB and VXLAN Configuration Workflows on page 19](#)

Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections Between Virtual and Physical Entities in a Data Center (Access Interfaces)

Supported Platforms [QFX Series standalone switches](#)

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 as a hardware VTEP, which serves as a Layer 2 gateway, and set up this device with an OVSDB connection to an NSX controller.

- [Requirements on page 30](#)
- [Overview and Topology on page 30](#)
- [Configuration on page 33](#)
- [Verification on page 35](#)

Requirements

The topology for this example includes the following hardware and software components:

- A physical server on which software applications directly run.
- A Juniper Networks switch that is configured to function as a hardware VTEP. This switch can be either a QFX5100 switch running Junos OS release 14.1X53-D15 or later, or a QFX10002 switch running Junos OS Release 15.1X53-D10 or later. The switch must also run a Juniper Networks VMware NSX software package. The Junos OS release number and the VMware NSX software package release number must be the same. For example, if the switch runs Junos OS Release 14.1X53-D15, the VMware NSX software package for release 14.1X53-D15 must be run to provide OVSDB functionality.
- 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 the configuration, you need to perform the following tasks:

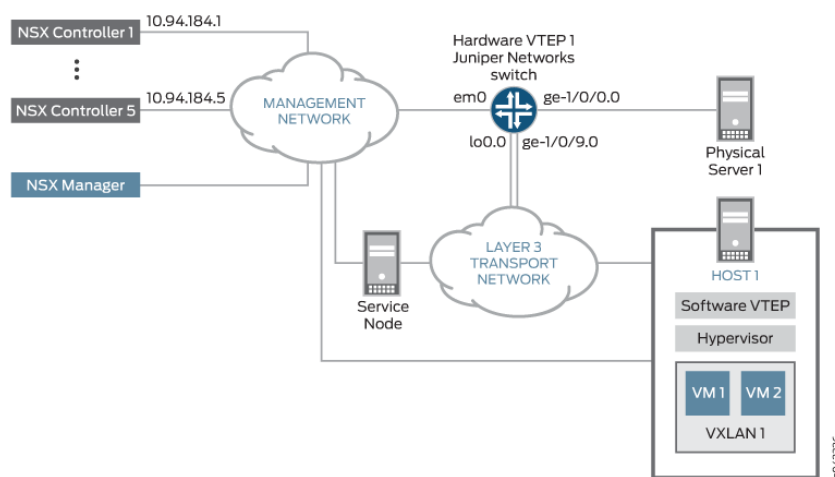
- On the Juniper Networks switch, install the Juniper Networks VMware NSX software package. For more information, see [“Installing Open vSwitch Database Components on Juniper Networks Devices” on page 22](#).
- Create an SSL private key and certificate, and install them in the `/var/db/certs` directory of the Juniper Networks device. For more information, see [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers” on page 23](#).
- Using NSX Manager, specify the IP address of the service node.

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

Overview and Topology

[Figure 3 on page 31](#) 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. To enable this communication, a Juniper Network switch is configured as Hardware VTEP 1.

Figure 3: VXLAN/OVSDb Layer 2 Gateway Topology



To establish communication between the software application on Physical Server 1 and VM 1 in VXLAN 1, some entities in the VXLAN-OVSDb topology must be configured in both NSX Manager and on the Juniper Networks switch. [Table 13 on page 31](#) 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 be different than the term used for essentially the same entity configured on the Junos Network switch. To prevent any confusion, [Table 13 on page 31](#) shows the NSX Manager and Junos OS entities side-by-side.

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 Juniper Networks Switch
VXLAN 1	Logical switch for VXLAN 1	VXLAN 1 NOTE: The Juniper Networks switch automatically configures this VXLAN.
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.	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 0. NOTE: A VLAN number of 0 indicates that the port is an access port.	One logical interface (ge-1/0/0.0) for VXLAN 1 NOTE: The Juniper Networks switch automatically configures this logical interface.
Juniper Networks switch (Hardware VTEP 1)	Gateway	—

To enable the Juniper Networks switch to automatically configure VXLAN 1 and logical interface ge-1/0/0.0, the **set switch-options ovssdb-managed** configuration mode command must be issued.

On the management interface of the Juniper Networks switch, a connection with an NSX controller is explicitly configured, by using the Junos OS CLI.

Based on the configuration of a logical switch (UUID 28805c1d-0122-495d-85df-19abd647d772) in NSX Manager, the Juniper Networks switch automatically creates a configuration for a Junos OS-equivalent VXLAN. A sample configuration is as follows:

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

On the Juniper Networks switch, Interface ge-1/0/0, which connects the switch to Physical Server 1, is configured as an OVSDB-managed interface. In NSX Manager, interface ge-1/0/0 is specified in the gateway service configuration, and in the logical port configuration, Interface ge-1/0/0 is configured as an access interface and is bound to the logical switch with the UUID of 28805c1d-0122-495d-85df-19abd647d772. Based on these configurations in the Junos OS CLI and NSX Manager, the Juniper Networks device automatically configures interface ge-1/0/0 as a Layer 2 access interface and associates logical interface ge-1/0/0.0 with VXLAN 28805c1d-0122-495d-85df-19abd647d772 by creating the following configuration:

```
set interfaces ge-1/0/0 encapsulation ethernet-bridge  
set interfaces ge-1/0/0 unit 0  
set vlans 28805c1d-0122-495d-85df-19abd647d772 interface ge-1/0/0.0
```

Each VXLAN-encapsulated packet must include a source IP address, which identifies the source hardware or software VTEP, in the outer IP header. In this example, the IP address of the loopback interface (lo0) on the Juniper Networks switch is used for Hardware VTEP 1.

Within VXLAN 28805c1d-0122-495d-85df-19abd647d772, Layer 2 BUM packets are replicated by the service node, which then forwards the replicas to all interfaces within the VXLAN. Having the service node handle the Layer 2 BUM traffic is the default behavior, and no configuration is required on the Juniper Networks device.

In this example, the tracing of all OVSDB events is configured. The output of the OVSDB events is placed in a file named **ovssdb**, which is stored in the **/var/log** directory. By default, a maximum of 10 trace files can exist, and the configured maximum size of each file is 10 MB.

[Table 14 on page 32](#) 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 14: Components Configured on Juniper Networks Switch that Functions as Hardware VTEP

Components	Settings
NSX controller	IP address: 10.94.184.1

Table 14: Components Configured on Juniper Networks Switch that Functions as Hardware VTEP (*continued*)

Components	Settings
OVSDb-managed interface	<p>Interface name: ge-1/0/0.</p> <p>NOTE: The Juniper Networks switch automatically creates this logical interface configuration, which is based on the gateway services configuration and the 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: access</p> <p>Associated with VXLAN 28805c1d-0122-495d-85df-19abd647d772</p>
OVSDb-managed VXLAN	<p>NOTE: The Juniper Networks device automatically creates this VXLAN configuration, which is based on the NSX-equivalent 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>

Configuration

To configure this example, you must perform these tasks:

- [Performing the Non-OVSDb and Non-VXLAN Configuration on page 34](#)
- [Configuring the Juniper Networks Switch as Hardware VTEP 1 on page 35](#)

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.

Non-OVSDb and non-VXLAN configuration:

```
set chassis network-services enhanced-ip
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
```

OVSDB and VXLAN configuration:

```
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
```



NOTE: After you configure and commit the OVSDB and VXLAN configuration in the Junos OS CLI, you must configure a logical switch for VXLAN 1. You must also configure a gateway for the Juniper Networks switch that functions as a hardware VTEP. And finally, you must configure a gateway service and a logical switch port for logical interface ge-1/0/0.0. All of these configuration must be performed in NSX Manager so that the Juniper Networks switch can automatically configure the VXLAN and associated logical interface. For more information about performing these configurations in NSX Manager, see the documentation that accompanies NSX Manager. For more information about key NSX Manager configuration details when configuring the gateway, gateway service, and logical switch port, see [“VMware NSX Configuration for Juniper Networks Devices That Function as Virtual Tunnel Endpoints” on page 26](#).

Performing the Non-OVSDB and Non-VXLAN Configuration

Step-by-Step Procedure

To perform the non-OVSDB and non-VXLAN configuration:

1. Configure the Layer 3 network.

```
[edit chassis]
user@switch# set network-services enhanced-ip
[edit interfaces]
user@switch# set ge-1/0/9 unit 0 family inet address 10.40.40.1/24
[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
[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
```

Configuring the Juniper Networks Switch as Hardware VTEP 1

Step-by-Step Procedure

To configure the Juniper Networks switch as Hardware VTEP 1 and with an OVSDb connection to an NSX controller, follow these steps:

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

[edit switch-options]
user@switch# set **ovsdb-managed**
2. Explicitly configure a connection with an NSX controller.

[edit protocols]
user@switch# set **ovsdb 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 **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. For information about configuring a logical switch, see the VMware documentation that accompanies NSX Manager.
8. In NSX Manager, configure a gateway for the Juniper Networks device, and configure a gateway service and logical switch port for the logical interface (ge-1/0/0.0). For more information about the tasks you must perform and key NSX Manager configuration details, see [“VMware NSX Configuration for Juniper Networks Devices That Function as Virtual Tunnel Endpoints” on page 26](#).

Verification

- [Verifying the Logical Switch on page 36](#)
- [Verifying the MAC Address of VM 1 on page 36](#)

- [Verifying the NSX Controller Connection on page 36](#)
- [Verifying the OVSDb-Managed Interface on page 37](#)

Verifying the Logical Switch

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 state (**Flags**) of the logical switch is **Created by both**.

Action Issue the **show ovssdb logical-switch** operational mode 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 Juniper Networks device 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 VMware NSX Logical Switch and Corresponding Junos OS OVSDb-Managed VXLAN” on page 79](#).

Verifying the MAC Address of VM 1

Purpose Verify that the MAC address of VM 1 is present in the OVSDb schema for physical devices.

Action Issue the **show ovssdb mac remote** operational mode command.

```
user@switch> show ovssdb mac remote
Logical Switch Name: 28805c1d-0122-495d-85df-19abd647d772
  Mac          IP          Encapsulation  Vtep
  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 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 Issue the **show ovssdb controller** operational mode 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 Juniper Networks switch.

Verifying the OVSDb-Managed Interface

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

Action Issue the **show ovssdb interface** operational mode command to verify that interface ge-1/0/0 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.

Related Documentation

- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDb Connections Between Virtual and Physical Entities in a Data Center \(Trunk Interfaces\)](#) on page 37

Example: Setting Up a VXLAN Layer 2 Gateway and OVSDb Connections Between Virtual and Physical Entities in a Data Center (Trunk Interfaces)

Supported Platforms [QFX Series standalone switches](#)

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 as a hardware VTEP, which serves as a Layer 2 gateway, and set up this device with an OVSDB connection to an NSX controller.

- [Requirements on page 38](#)
- [Overview and Topology on page 39](#)
- [Configuration on page 43](#)
- [Verification on page 45](#)

Requirements

The topology for this example includes the following hardware and software components:

- A physical server on which software applications directly run.
- A Juniper Networks switch that is configured to function as a hardware VTEP. This switch can be either a QFX5100 switch running Junos OS Release 14.1X53-D15 or later, or a QFX10002 switch running Junos OS Release 15.1X53-D10 or later. The switch must also run a Juniper Networks VMware NSX software package. The Junos OS release number and the VMware NSX software package release number must be the same. For example, if the QFX5100 switch runs Junos OS Release 14.1X53-D15, the VMware NSX software package for release 14.1X53-D15 must be run to provide OVSDB functionality.
- 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 used in this example.
- 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 need to perform the following tasks:

- On the Juniper Networks switch, install the Juniper Networks VMware NSX software package. For more information, see [“Installing Open vSwitch Database Components on Juniper Networks Devices” on page 22](#).
- Create an SSL private key and certificate, and install them in the `/var/db/certs` directory of the Juniper Networks switch. For more information, see [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers” on page 23](#).

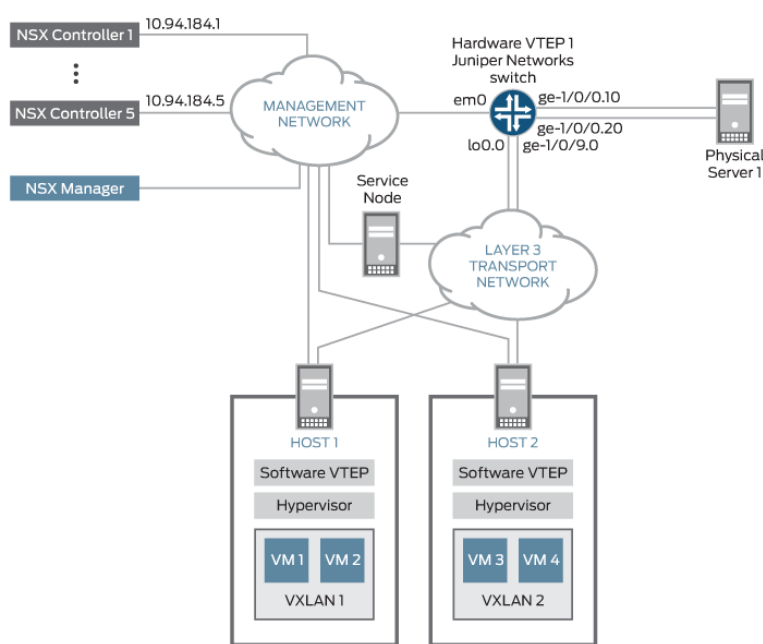
- Using NSX Manager, specify the IP address of the service node.

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

Overview and Topology

Figure 4 on page 39 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 4: VXLAN/OVSDB Layer 2 Gateway Topology



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 13 on page 31 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 be different than the term used for essentially the same entity configured on the Junos Network switch. To prevent confusion, Table 13 on page 31 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 automatically 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.	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 automatically configures these logical interfaces.
Juniper Networks switch (Hardware VTEP 1)	Gateway	—

To enable the Juniper Networks switch to automatically configure the VXLANs and their associated interfaces, the **set switch-options ovssdb-managed** configuration mode command must be issued.

On the management interface of the Juniper Networks switch, a connection with an NSX controller is explicitly configured, by using the Junos OS CLI.

Based on the configuration of the entities in NSX Manager as described in [Table 13 on page 31](#), the Juniper Networks switch automatically creates VXLANs 1 and 2 and their associated logical interfaces. [Table 16 on page 41](#) provides the relevant NSX Manager configuration and the resulting VXLANs and associated logical interfaces that the Juniper Networks switch automatically configures.

Table 16: NSX Manager Configurations, and Resulting Automatically Configured by Juniper Networks Switch

Relevant NSX Manager Configuration: Logical Switch and Logical Switch Port	VXLANs and Associated Logical Interfaces Automatically 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.

Each VXLAN-encapsulated packet must include a source IP address, which identifies the source hardware or software VTEP, in the outer IP header. In this example, the IP address of the loopback interface (lo0) on the Juniper Networks switch is used for Hardware VTEP 1.

Within VXLAN 28805c1d-0122-495d-85df-19abd647d772 and VXLAN 9acc24b3-7b0a-4c2e-b572-3370c3e1acff, Layer 2 BUM packets are replicated by the service node, which then forwards the replicas to all interfaces in the VXLANs. Having the service node handle the Layer 2 BUM traffic is the default behavior, and no configuration is required on the Juniper Networks switch.

In this example, the tracing of all OVSDB events is configured. The output of the OVSDB events is placed in a file named **ovsdb**, which is stored in the **/var/log** directory. By default, a maximum of 10 trace files can exist, and the configured maximum size of each file is 10 MB.

[Table 14 on page 32](#) 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 Configured on Juniper Networks Switch that Functions as a Hardware VTEP

Components	Settings
NSX controller	IP address: 10.94.184.1
OVSDB-managed interface	Interface name: ge-1/0/0
VXLAN 1 and associated logical interface	<p>NOTE: The Juniper Networks switch automatically 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 automatically 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>

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

Components	Settings
Handling of Layer 2 BUM traffic within VXLAN 28805c1d-0122-495d-85df-19abd647d772 and within VXLAN 9acc24b3-7b0a-4c2e-b572-3370c3e1acff	Service node NOTE: By default, one or more service nodes handle Layer 2 BUM traffic in a VXLAN; therefore, no configuration is required.

Configuration

To configure this example, you must perform these tasks:

- [Performing the Non-OVSDb and Non-VXLAN Configuration on page 44](#)
- [Configuring the Juniper Networks Switch as Hardware VTEP 1 on page 44](#)

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.

Non-OVSDb and non-VXLAN configuration:

```
set chassis network-services enhanced-ip
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
```

OVSDb and VXLAN configuration:

```
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
```



NOTE: After you configure and commit the OVSDB and VXLAN configuration in the Junos OS CLI, you must configure a logical switch for VXLAN 1 and another for VXLAN 2. You must also configure a gateway for the Juniper Networks switch that functions as a hardware VTEP. You must also configure a gateway service for OVSDB-managed interface ge-1/0/0. And finally, you must configure 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. All of these configurations must be performed in NSX Manager so that the Juniper Networks switch can automatically configure the VXLANs and associated logical interfaces. For information about performing these configurations in NSX Manager, see the documentation that accompanies NSX Manager. For information about key NSX Manager configuration details when configuring the gateway, gateway service, and logical switch port, see [“VMware NSX Configuration for Juniper Networks Devices That Function as Virtual Tunnel Endpoints”](#) on page 26.

Performing the Non-OVSDB and Non-VXLAN Configuration

Step-by-Step Procedure

To perform the non-OVSDB and non-VXLAN configuration:

1. Configure the Layer 3 network.

```
[edit chassis]
user@switch# set network-services enhanced-ip
[edit interfaces]
user@switch# set ge-1/0/9 unit 0 family inet address 10.40.40.1/24
[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
[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
```

Configuring the Juniper Networks Switch as Hardware VTEP 1

Step-by-Step Procedure

To configure the Juniper Networks switch as Hardware VTEP 1 and with an OVSDB connection to an NSX controller, follow these steps:

1. Enable the Juniper Networks switch to automatically configure OVSDB-managed VXLANs and associated interfaces.

```
[edit switch-options]
user@switch# set ovsdb-managed
```

2. Explicitly configure a connection with an NSX controller.

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

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

```
[edit protocols]
user@switch# set ovssdb interfaces ge-1/0/0
```
4. Set up OVSDb tracing operations.

```
[edit protocols]
user@switch# set ovssdb traceoptions file ovssdb
user@switch# set ovssdb traceoptions file size 10m
user@switch# set ovssdb 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. For information about configuring a logical switch, 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.

For information about configuring these entities in NSX Manager, see the documentation that accompanies NSX Manager. For information about the key NSX Manager configuration details, see “[VMware NSX Configuration for Juniper Networks Devices That Function as Virtual Tunnel Endpoints](#)” on page 26.

Verification

- [Verifying the Logical Switch on page 45](#)
- [Verifying the MAC Addresses of VM 1, VM 3, and VM 4 on page 46](#)
- [Verifying the NSX Controller Connection on page 46](#)
- [Verifying the OVSDb-Managed Interface on page 47](#)

Verifying the Logical Switch

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 state (**Flags**) of the logical switches is **Created by both**.

Action Issue the **show ovssdb logical-switch** operational mode 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 automatically 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 VMware NSX Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN” on page 79](#).

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 Issue the **show ovssdb mac remote** operational mode 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 Issue the **show ovssdb controller** operational mode 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 Issue the **show ovssdb interface** operational mode 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**.

Related Documentation

- [Example: Setting Up a VXLAN Layer 2 Gateway and OVSDb Connections Between Virtual and Physical Entities in a Data Center \(Access Interfaces\)](#) on page 29

CHAPTER 3

OVSDb Configuration Statements

- [controller \(OVSDb\) on page 50](#)
- [inactivity-probe-duration on page 51](#)
- [interfaces \(OVSDb\) on page 52](#)
- [maximum-backoff-duration on page 53](#)
- [ovsdb on page 54](#)
- [ovsdb-managed on page 55](#)
- [port \(OVSDb\) on page 56](#)
- [protocol \(OVSDb\) on page 57](#)
- [traceoptions \(OVSDb\) on page 58](#)

controller (OVSDB)

Supported Platforms [EX Series](#), [MX Series](#), [QFX Series standalone switches](#)

Syntax `controller ip-address {
 inactivity-probe-duration milliseconds;
 maximum-backoff-duration milliseconds;
 protocol protocol {
 port number;
 }
}`

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 Configure a connection between a Juniper Networks device and a VMware NSX controller. The Juniper Networks device must be running a release that supports the Open vSwitch Database (OVSDB) management protocol and have the Juniper Networks NSX VMware software package installed. The NSX VMware software package release must be the same as the Junos OS release running on the device.

The Junos OS implementation of OVSDB supports one cluster of NSX controllers, which includes three or five controllers as recommended by VMware.

To implement OVSDB on a Juniper Networks device, you must explicitly configure a connection to at least 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.

Options *ip-address*—IPv4 address of the NSX controller.

The remaining statements are explained separately.

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

Related Documentation

- [Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices](#)
- [Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)

inactivity-probe-duration

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
Syntax	inactivity-probe-duration <i>milliseconds</i> ;
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 VMware NSX controller can be inactive before an inactivity probe is sent.
Options	milliseconds —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 Open vSwitch Database Management Protocol on Juniper Networks Devices• Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8

interfaces (OVSDB)

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
Syntax	interfaces <i>interface-name</i> ;
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 interfaces to be managed by the Open vSwitch Database (OVSDB) management protocol. 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">• Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices• Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices on page 24

maximum-backoff-duration

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
Syntax	maximum-backoff-duration <i>milliseconds</i> ;
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 re-attempts to connect with a VMware NSX controller if a previous attempt failed.
Options	<i>milliseconds</i> —Number of milliseconds a Juniper Networks device waits before it re-attempts to connect with an NSX 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 Open vSwitch Database Management Protocol on Juniper Networks Devices • Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8

ovsdb

Supported Platforms EX Series, MX Series, QFX Series standalone switches

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 Juniper Networks device must have a Junos OS release and the Juniper Networks VMware NSX software package installed. The VMware NSX software package release must be the same as the Junos OS release.

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 Open vSwitch Database Management Protocol Running on Juniper Networks Devices on page 7](#)
- [Configuring OVSDb-Managed VXLANs](#)

ovsdb-managed

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
Syntax	ovsdb-managed;
Hierarchy Level	[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]
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>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 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, this implementation of OVSDb uses the multicast scheme described in “Understanding How Layer 2 BUM Traffic and Layer 3 Routed Multicast Traffic Are Handled in VXLANs Managed by OVSDb” on page 9. 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	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"> • Configuring OVSDb-Managed VXLANs

port (OVSDb)

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
Syntax	port <i>number</i> ;
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 VMware NSX controller port to which a Juniper Networks device that supports the Open vSwitch Database (OVSDb) management protocol connects.
Options	<i>number</i> —Port number of NSX 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">• <i>Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices</i>• Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8

protocol (OVSDb)

Supported Platforms [EX Series](#), [MX Series](#), [QFX Series standalone switches](#)

Syntax `protocol protocol {
 port number;
}`

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 security protocol that protects the connection between a Juniper Networks device that supports the Open vSwitch Database (OVSDb) management protocol and a VMware NSX controller.

The Secure Sockets Layer (SSL) connection requires a private key and certificates, which must be stored in the `/var/db/certs` directory of the Juniper Networks device. For more information about the files, including actions you must take to create and install some of the files, see [“Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with VMware NSX Controllers”](#) on page 23.

Options *protocol*—Establish a secure connection to the NSX controller, using SSL or the Transmission Control Protocol (TCP).



NOTE: SSL is the only supported connection protocol.

Default: `ssl`

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

- [Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices](#)
- [Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)

traceoptions (OVSDB)

Supported Platforms EX Series, MX Series, QFX Series standalone switches

Syntax

```
traceoptions {  
    file <filename> <files number> <match regular-expression> <no-world-readable |  
    world-readable> <size size>;  
    flag flag;  
    no-remote-trace;  
}
```

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 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 **file filename**—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.
Default: **/var/log/vgd**

files number—(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.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option and a filename.

Range: 2 through 1000 files

Default: 10 files

flag flag—Tracing operation to perform. You can include one or more of the following flags:

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—(QFX5100 and QFX10002 switches only) Events for the automatic 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 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>Example: Setting Up Inter-VXLAN Unicast Routing and OVSDDB Connections in a Data Center</i>• <i>Example: Setting Up Inter-VXLAN Unicast and Multicast Routing and OVSDDB Connections in a Data Center</i>
------------------------------	---

CHAPTER 4

OVSDDB Monitoring Commands

- `show ovssdb controller`
- `show ovssdb interface`
- `show ovssdb logical-switch`
- `show ovssdb mac`
- `show ovssdb statistics interface`
- `show ovssdb virtual-tunnel-end-point`
- Verifying that a VMware NSX Logical Switch and Corresponding Junos OS OVSSDB-Managed VXLAN Are Working Properly on page 76

show ovssdb controller

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

Syntax `show ovssdb controller`
`<address ip-address>`

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 VMware NSX controllers to which the Juniper Networks device is connected. This command displays information for NSX controllers with connections to a Juniper Networks device that are made in the following ways:

- With explicit configuration—You explicitly configure a connection with an NSX controller using the Junos OS CLI.
- Without explicit configuration—After you explicitly configure a connection with an NSX controller, the Juniper Networks device can establish a connection other NSX controllers in the same cluster. For example, you explicitly configure a connection with NSX controller 1, and the Juniper Networks device learns about NSX controllers 2 and 3 in the same cluster after NSX controller 1 pushes information about controllers 2 and 3. The Juniper Networks device then establishes connections with NSX controllers 2 and 3.

Options **none**—Display information about all NSX controllers to which the Juniper Networks device is connected.

address ip-address—(Optional) Display information about the NSX controller with the specified IP address.

Required Privilege Level admin

Related Documentation

- [Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices](#)
- [Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices on page 24](#)
- [Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and Controllers on page 8](#)

List of Sample Output [show ovssdb controller on page 63](#)
[show ovssdb controller address on page 63](#)

Output Fields [Table 18 on page 63](#) lists the output fields for the **show ovssdb controller** command. Output fields are listed in the approximate order in which they appear.

Table 18: show ovssdb controller Output Fields

Field Name	Field Descriptions
Controller IP address	IP address of an NSX controller to which the Juniper Networks device is connected.
Controller protocol	Protocol used by the Juniper Networks device to initiate a connection with an NSX controller.
Controller port	NSX controller port to which the Juniper Networks device is connected.
Controller connection	State of the connection between the Juniper Networks device and an NSX controller.
Controller seconds-since-connect	Number of seconds since the connection between the Juniper Networks device and NSX controller was established.
Controller seconds-since-disconnect	Number of seconds since the connection between the Juniper Networks device and NSX controller was dropped.
Controller connection status	Status of the connection between the Juniper Networks device and an NSX controller.

Sample Output

show ovssdb controller

```

user@host> show ovssdb 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 ovssdb controller address

```

user@host> show ovssdb 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

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
Syntax	show ovssdb interface <interface-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	Display information about Open vSwitch Database (OVSDb)-managed interfaces configured by using the interfaces interface-name statement in the [edit protocols ovssdb] hierarchy.
Options	none —Display information about all OVSDb-managed interfaces. interface-name —(Optional) Display information about the specified OVSDb-managed interface.
Required Privilege Level	admin
Related Documentation	<ul style="list-style-type: none"> • Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices • Setting Up the Open vSwitch Database Management Protocol on Juniper Networks Devices on page 24 • show ovssdb statistics interface on page 73
List of Sample Output	show ovssdb interface on page 66 show ovssdb (Specific Interface) on page 66
Output Fields	Table 19 on page 65 lists the output fields for the show ovssdb interface command. Output fields are listed in the approximate order in which they appear.

Table 19: 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: MX Series routers and EX9200 switches do not support this field.
Bridge domain or VLAN	Bridge domain or VLAN under which the VXLAN is created. NOTE: MX Series routers and EX9200 switches do not support this field.

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 ovssdb (Specific Interface)

```
user@host> show ovssdb interface ge-7/0/9.0
Interface          VLAN ID          Bridge-domain
ge-7/0/9.0
```

show ovssdb logical-switch

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
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	<p>Display information about logical switches, which you configured in VMware NSX Manager or in the NSX API, and the corresponding Virtual Extensible LANs (VXLANs), which were configured on the Juniper Networks device.</p> <p>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.</p> <p>The show ovssdb logical-switch command displays the state of the logical switch (Flags), which can be one of the following:</p> <p>Created by Controller—A logical switch was configured in NSX Manager or in the NSX API. In this state, the logical switch and corresponding VXLAN are not yet operational.</p> <p>Created by L2ALD—A VXLAN was configured on a Juniper Networks device. In this state, the logical switch and corresponding VXLAN are not yet operational.</p> <p>Created by both—A logical switch was configured in NSX Manager or in the NSX API, and a corresponding VXLAN was configured on a Juniper Networks device. In this state, the logical switch and corresponding VXLAN are operational.</p> <p>Tunnel key mismatch—The VXLAN network identifiers (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.</p>
Options	<p>none—Display information about all logical switches that are present in the OVSDb schema for physical devices.</p> <p>logical-switch-name—(Optional) Display information about the specified logical switch.</p>
Required Privilege Level	admin
Related Documentation	<ul style="list-style-type: none"> Open vSwitch Database Schema for Physical Devices Troubleshooting a Nonoperational VMware NSX Logical Switch and Corresponding Junos OS OVSDb-Managed VXLAN on page 79
List of Sample Output	show ovssdb logical-switch on page 68 show ovssdb logical-switch (Specific Logical Switch) on page 68

Output Fields Table 20 on page 68 lists the output fields for the **show ovssdb logical-switch** command. Output fields are listed in the approximate order in which they appear.

Table 20: show ovssdb logical-switch Output Fields

Field Name	Field Description
Logical Switch Name	UUID that is automatically generated by NSX and assigned to the logical switch after you configure it in NSX Manager or in the NSX API. The corresponding VXLAN configured on the Juniper Networks device must use the same UUID for 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. The VNI configured for the logical switch and corresponding VXLAN must be the same.
Num of Remote MAC	The total number of remote MAC addresses associated with the logical switch. These addresses are learned by software and hardware virtual tunnel endpoints (VTEPs) in the NSX environment.
Num of Local MAC	The total number of local MAC addresses associated with the logical switch. These addresses are learned on the physical ports of the Juniper Networks devices.

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 ovssdb mac

Supported Platforms EX Series, MX Series, QFX Series standalone switches

Syntax show ovssdb 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 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 in the Junos OS environment are known as *local addresses*, and the addresses learned from other software or hardware VTEPs in the NSX environment 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 ovssdb 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*—(Optional) Display the specified MAC address.

count—(All Juniper Networks devices that support OpenFlow except EX9200 switches)
 (Optional) 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. 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 ovssdb mac count local remote unicast** command.

local—(Optional) Display all local MAC addresses.

logical-switch *logical-switch-uuid*—(Optional) Display all MAC addresses associated with the specified logical switch.

multicast—(Optional) Display all multicast MAC addresses.

remote—(Optional) Display all remote MAC addresses.

unicast—(Optional) Display all unicast MAC addresses.

Required Privilege Level	admin
Related Documentation	<ul style="list-style-type: none"> • <i>Open vSwitch Database Schema for Physical Devices</i>
List of Sample Output	show ovssdb mac on page 70 show ovssdb mac (Specific MAC Address) on page 71 show ovssdb mac (Specific Logical Switch) on page 71 show ovssdb mac (Local Unicast MAC Addresses) on page 72 show ovssdb mac (Count of All Local, Remote, Unicast, and Multicast MAC Addresses For All Logical Switches) on page 72
Output Fields	Table 21 on page 70 lists the output fields for the show ovssdb mac command. Output fields are listed in the approximate order in which they appear.

Table 21: show ovssdb mac Output Fields

Field Name	Field Description
Logical Switch Name	Universally unique identifier (UUID) of the logical switch. The UUID is automatically generated by NSX and assigned to the logical switch after you configure it in NSX Manager or in the NSX API.
MAC Address	MAC addresses of virtual machines (VMs).
IP Address	IP address of VMs. NOTE: If the IP addresses of VMs are not published by the NSX 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	NOTE: This field is supported by all Juniper Networks devices that support OpenFlow 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 ovssdb mac (Specific MAC Address)

```
user@host> show ovssdb 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 ovssdb mac (Specific Logical Switch)

```
user@host> show ovssdb 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 MAC Addresses)

```
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 ovbdb statistics interface

Supported Platforms	EX Series, MX Series, QFX Series standalone switches
Syntax	show ovbdb statistics interface <interface-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	Display statistics for Open vSwitch Database (OVSDb)-managed interfaces configured by using the interfaces interface-name statement in the [edit protocols ovbdb] hierarchy. When an interface is configured as OVSDb-managed, the collection of statistics for that interface begins. When you issue the show ovbdb statistics interface command, the statistics displayed include data collected since the inception of the interface as OVSDb-managed and up to the current time.
Options	none —Display statistics for all configured OVSDb-managed interfaces. interface-name —(Optional) Display statistics for the specified interface.
Required Privilege Level	admin
Related Documentation	<ul style="list-style-type: none"> interfaces on page 52
List of Sample Output	show ovbdb statistics interface on page 73 show ovbdb statistics interface (Specific Interface) on page 74
Output Fields	Table 22 on page 73 lists the output fields for the show ovbdb statistics interface command. Output fields are listed in the approximate order in which they appear.

Table 22: show ovbdb statistics interface Output Fields

Field Name	Field Descriptions
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 ovbdb statistics interface

```
user@host> show ovbdb 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 ovssdb statistics interface (Specific Interface)

```
user@host> show ovssdb 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

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

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:

- Other Juniper Networks devices that function as hardware virtual tunnel endpoints (VTEPs) in the physical network
- Hardware and software VTEPs in the virtual network
- Service nodes in the virtual network

Options **none**—Display information about all VTEPs and service nodes that the Juniper Networks device has learned.

address *ip-address*—(Optional) Display information about the entity with specified IP address.

encapsulation *encapsulation-type*—(Optional) 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 76](#)
[show ovssdb virtual-tunnel-end-point \(Specific VTEP\) on page 76](#)
[show ovssdb virtual-tunnel-end-point \(Specific Encapsulation Type\) on page 76](#)
[show ovssdb virtual-tunnel-end-point \(Specific VTEP and Encapsulation Type\) on page 76](#)

Output Fields [Table 23 on page 75](#) 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 23: 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 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 (Specific VTEP)

```
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 (Specific Encapsulation Type)

```
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 (Specific VTEP and Encapsulation Type)

```
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
```

Verifying that a VMware NSX Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN Are Working Properly

Supported Platforms [EX Series](#), [MX Series](#), [QFX Series standalone switches](#)

Purpose Verify the following:

- A logical switch, which is configured by using VMware NSX Manager or the NSX API, is learning MAC addresses in the NSX environment.
- The corresponding Open vSwitch Database (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 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 and corresponding OVSDB-managed VXLAN are learning MAC addresses in their respective environments and exchanging them, 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

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

If the output in the **Flags** field displays a state other than **Created by both**, see [“Troubleshooting a Nonoperational VMware NSX Logical Switch and Corresponding Junos OS OVSDDB-Managed VXLAN” on page 79](#).

Related Documentation

- [show ovssdb logical-switch on page 67](#)

CHAPTER 5

Troubleshooting OVSDb

- Troubleshooting a Nonoperational VMware NSX Logical Switch and Corresponding Junos OS OVSDb-Managed VXLAN on page 79

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

Supported Platforms [EX Series, MX Series, QFX Series standalone switches](#)

Problem **Description:** A logical switch, which is configured by using VMware NSX Manager or the NSX API, and the corresponding Open vSwitch Database (OVSDb)-managed Virtual Extensible LAN (VXLAN), which is configured on a Juniper Networks device, are not exchanging MAC addresses learned in the NSX and Junos OS environments, respectively. Also, 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 NSX Manager or in the NSX API, but a corresponding 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, but a corresponding logical switch is not configured in NSX Manager or in the NSX API.
- If the **Flags** field displays **Tunnel key mismatch**, the VXLAN network identifiers (VNIs) in the logical switch and corresponding VXLAN configurations do not match.

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

- On the Juniper Networks device on which the corresponding VXLAN configuration should reside, determine whether or not the VXLAN configuration exists.
- If your Juniper Networks device supports the automatic configuration of VXLANs, and the VXLAN configuration does not exist, 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 device to automatically create OVSDB-managed VXLANs. If this command was already issued, another possible cause is that the L2ALD daemon has become non-functional. 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.

- If your Juniper Networks device supports the manual configuration of VXLANs, and the VXLAN configuration does not exist, configure it using the procedure in *Configuring OVSDB-Managed VXLANs*. If the VXLAN configuration exists, check the VXLAN name to make sure that it is the universally unique identifier (UUID) of the logical switch. Also, check the VNI to make sure that the value is the same as the value in the logical switch configuration.

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

- On Juniper Networks devices that support the automatic configuration of VXLANs, two issues exist. First, despite the fact that the device automatically creates OVSDB-managed VXLANs, this VXLAN was manually configured by using the Junos OS CLI. Second, a corresponding logical switch was not configured. To resolve both issues, configure a logical switch in NSX Manager or in the NSX API. After the NSX controller pushes relevant logical switch information to the device, the device automatically creates a corresponding VXLAN and deletes the manually configured VXLAN.
- On Juniper Networks devices that support the manual configuration of VXLANs, determine whether or not a corresponding logical switch is configured in NSX Manager or in the NSX API. If a logical switch is not configured, configure one, keeping in mind that NSX automatically generates a UUID for the logical switch and that this UUID must be specified as the name of the VXLAN. Another possibility is that the logical switch might exist, but the logical switch UUID might not be specified as the VXLAN name. In NSX Manager or in the NSX API, check for a logical switch 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 Juniper Networks devices that support the automatic configuration of VXLANs, check the configuration of the VNI in NSX Manager or in the NSX API to see whether it was changed after the device created the corresponding VXLAN. If it was changed, update the VNI on the Juniper Networks device, using the Junos OS CLI.
- For Juniper Networks devices that support the manual configuration of VXLANs, check the values of the VNI in NSX Manager or in the NSX API and the Junos OS CLI, and change the incorrect value.

Related Documentation

- [Understanding Automatically Configured Virtual Extensible LANs in an Open vSwitch Database Environment on page 10](#)
- [Understanding How to Manually Configure Virtual Extensible LANs in an Open vSwitch Database Environment](#)
- [show ovssdb logical-switch on page 67](#)