JUNIPER NETWORKS

Junos® OS

Contrail Feature Guide for QFX5100 Switches

Release
14.1

Modified: 2015-10-06

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

Junos® OS Contrail Feature Guide for QFX5100 Switches
14.1

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

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

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

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

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

Supported Platforms

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

- QFX Series 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.xsl;
       }
     }
   }
   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.xsl; }
   ```

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 xiii defines notice icons used in this guide.

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**

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

Convention	Description	Examples
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	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.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric** ***metric*****>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  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.	• 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 http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/

- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

CHAPTER 1

OVSDB and VXLAN Overview

Understanding VXLANs

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

VXLAN Benefits

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

- You can theoretically create as many as 16 million VXLANs in an administrative domain (as opposed to 4094 VLANs on a Juniper Networks device).
  - MX Series routers and EX9200 switches support as many as 32K VXLANs, 32K multicast groups, and 8K virtual tunnel endpoints (VTEPs). This means that VXLANs based on MX Series routers provide network segmentation at the scale required by cloud builders to support very large numbers of tenants.
  - QFX 5100 switches support 4K VXLANs, 4K multicast groups, and 2K VTEPs.
- You can enable migration of virtual machines between servers that exist in separate Layer 2 domains by tunneling the traffic over Layer 3 networks. This functionality allows you to dynamically allocate resources within or between data centers without being constrained by Layer 2 boundaries or being forced to create large or geographically stretched Layer 2 domains.

Using VXLANs to create smaller Layer 2 domains that are connected over a Layer 3 network means that you don't need to use STP to converge the topology but can use more-robust routing protocols in the Layer 3 network instead. In the absence of STP, none of your links are blocked, which means you can get full value from all the ports that you purchase. Using routing protocols to connect your Layer 2 domains also allows you to load balance the traffic to ensure that you get the best use of your available bandwidth. Given the amount of east-west traffic that often flows within or between data centers, maximizing your network performance for that traffic is very important.

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

Video: Why Use an Overlay Network in a Data Center?

How Does VXLAN Work?

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

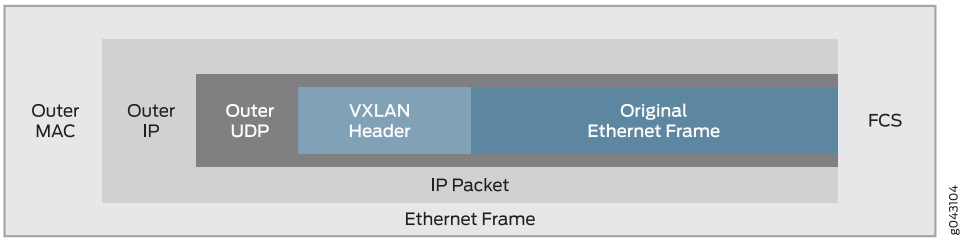
- Outer MAC destination address (MAC address of the tunnel endpoint VTEP)
- Outer MAC source address (MAC address of the tunnel source VTEP)
- Outer IP destination address (IP address of the tunnel endpoint VTEP)

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

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

Figure 1 on page 5 shows the VXLAN packet format.

Figure 1: VXLAN Packet Format



VXLAN Configuration Methods

You can configure VXLANs manually (without using a SDN controller) or you can use a SDN controller to create and manage VXLANs in a more automated and centralized manner. SDN controllers use the Open vSwitch Database (OVSDB) management protocol to provide a means through which controllers (such as a VMware NSX or Juniper Contrail controller) and Juniper Networks devices that support OVSDB can communicate.

Using a QFX5100 Switch with VXLANs

You can configure a QFX5100 switch to perform all of the following roles:

- Act as a transit Layer 3 switch for downstream hosts acting as VTEPs. In this configuration, you do not need to configure any VXLAN functionality on the switch. You do need to configure IGMP and PIM so that the switch can form the multicast trees for the VXLAN multicast groups. (See Manual VXLANs Require PIM on page 7 for more information.)
- Act as a Layer 2 gateway between virtualized and non-virtualized networks in the same data center or between data centers. For example, you can use a QFX5100 switch to connect a network that uses VXLANs to one that uses VLANs.
- Act as a Layer 2 gateway between virtualized networks in the same or different data centers and allow virtual machines to move (VMotion) between those networks and

data centers. For example, if you want to allow VMotion between devices in two different networks, you can create the same VLAN in both networks and put both devices on that VLAN. The QFX5100 switches connected to these devices, acting as VTEPs, can map that VLAN to the same VXLAN, and the VXLAN traffic can then be routed between the two networks.

NOTE: A QFX 5100 switch cannot route traffic between different VXLANs. To connect devices in different VXLANs you need a VXLAN-capable Layer 3 gateway, such as a Juniper Networks MX Series router.

Because the additional headers add 50 to 54 bytes, you might need to increase the MTU on a QFX5100 VTEP to accommodate larger packets. For example, if the switch is using the default MTU value of 1514 bytes and you want to forward 1500-byte packets over the VXLAN, you need to increase the MTU to allow for the increased packet size caused by the additional headers.

Changing the UDP Port on a QFX5100 Switch

Starting with Junos OS 14.1X53-D25, you can configure the UDP port used as the destination port for VXLAN traffic on a QFX5100 switch. To configure the VXLAN destination port to be something other than the default UDP port of 4789, enter

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

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

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

Using an MX Series Router or EX9200 Switch as a VTEP

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

- Act as a Layer 2 gateway between virtualized and non-virtualized networks in the same data center or between data centers. For example, you can use an MX Series router to connect a network that uses VXLANs to one that uses VLANs.
- Act as a Layer 2 gateway between virtualized networks in the same or different data centers and allow virtual machines to move (VMotion) between those networks and data centers.

- Act as a Layer 3 gateway to route traffic between different VXLANs in the same data center.
- Act as a Layer 3 gateway to route traffic between different VXLANs in different data centers over a WAN or the Internet using standard routing protocols or VPLS tunnels.

NOTE: If you want an MX Series router or EX9200 switch to be a VXLAN Layer 3 gateway, you must configure integrated routing and bridging (IRB) interfaces to connect the VXLANs, just as you do if you want to route traffic between VLANs.

Manual VXLANs Require PIM

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

You must also configure each VTEP in a given VXLAN to be a member of the same multicast group. (If possible, you should assign a different multicast group address to each VXLAN, though this is not required. Multiple VXLANs can share the same multicast group.) The VTEPs can then forward ARP requests they receive from their connected hosts to the multicast group. The other VTEPs in the group de-encapsulate the VXLAN information, and (assuming they are members of the same VXLAN) they forward the ARP request to their connected hosts. When the target host receives the ARP request, it responds with its MAC address, and its VTEP forwards this ARP reply back to the source VTEP. Through this process, the VTEPs learn the IP addresses of the other VTEPs in the VXLAN and the MAC addresses of the hosts connected to the other VTEPs.

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

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

Load Balancing VXLAN Traffic

On QFX5100 switches, the Layer 3 routes that form VXLAN tunnels use per-packet load balancing by default, which means that load balancing is implemented if there are ECMP

paths to the remote VTEP. This is different from normal routing behavior in which per-packet load balancing is not used by default. (Normal routing uses per-prefix load balancing by default.)

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

Using ping and traceroute With a VXLAN

On a QFX5100 switch, you can use the **ping** and **traceroute** commands to troubleshoot traffic flow through a VXLAN tunnel by including the **overlay** parameter and various options. You use these options to force the **ping** or **traceroute** packets to follow the same path as data packets through the VXLAN tunnel. In other words, you make the underlay packets (**ping** and **traceroute**) take the same route as the overlay packets (data traffic). See ping overlay and traceroute overlay for more information.

VXLAN Constraints on QFX5100 Switches

When configuring VXLANs on QFX5100 switches, be aware of the constraints in the following list. In this list, “Layer 3 side” refers to a network-facing interface that performs VXLAN encapsulation and de-encapsulation, and “Layer 2 side” refers to a server-facing interface that is a member of a VLAN that is mapped to a VXLAN.

- You can use VXLANs on a Virtual Chassis or Virtual Chassis Fabric if all of the members are QFX5100 switches. You cannot use VXLANs if any of the members is not a QFX5100 switch.
- VXLAN configuration is supported only in the default routing instance.
- A QFX 5100 switch cannot route traffic between different VXLANs.
- A physical interface cannot be a member of a VLAN and a VXLAN. That is, an interface that performs VXLAN encapsulation and de-encapsulation cannot also be a member of a VLAN. For example, if a VLAN that is a mapped to a VXLAN is a member of trunk port xe-0/0/0, any other VLAN that is a member of xe-0/0/0 must also be assigned to a VXLAN.
- Multichassis link aggregation groups (MC-LAGS) are not supported with VXLAN.
- IP fragmentation and defragmentation are not supported on the Layer 3 side.
- The following features are not supported on the Layer 2 side:
  - STP (any variant)
  - IGMP snooping
  - storm control
- Access port security features are not supported with VXLAN. For example, the following features are not supported:

  - DHCP snooping
  - dynamic ARP inspection
  - MAC limiting and MAC move limiting
- Ingress node replication is not supported. (You must use PIM to advertise and learn about reachability if you do not use VMware's NSX controller.)
- PIM-BIDIR and PIM-SSM are not supported with VXLANs.
- Class of service (CoS) features are not supported with VXLANs.
- If you configure a port-mirroring instance to mirror traffic egressing from an interface that performs VXLAN encapsulation, the source and destination MAC addresses of the mirrored packets are invalid. The original VXLAN traffic is not affected.

Related Documentation

- *Examples: Manually Configuring VXLANs on QFX Series Switches*
- *Example: Manually Configuring VXLANs on MX Series Routers*
- OVSDB Support on Juniper Networks Devices on page 9
- *mtu*

OVSDB Support on Juniper Networks Devices

Table 3 on page 9 lists the Juniper Networks devices that support the Open vSwitch Database (OVSDB) management protocol and the Junos OS releases in which OVSDB is supported. For each device and Junos OS release, the table outlines whether or not the OVSDB software is included in the Junos OS software (**jinstall**) package. If the OVSDB software is not included, the table also includes the name of the separate OVSDB software (**jsdn**) package that must be installed on the device in addition to the Junos OS release.

NOTE: The separate OVSDB software package release must be the same as the Junos OS release running on the device.

Table 3: OVSDB Support on Juniper Networks Devices

Juniper Networks Device	Junos OS Release	OVSDB Software Included in Junos OS Software (jinstall) Package?	Separate OVSDB Software (jsdn) Package Name
MX80 3D Universal Edge Routers	14.1R2 and later	No	jsdn-powerpc-*release*
MX240, MX480, MX960 3D Universal Edge Routers	14.1R2 and later	No	jsdn-i386-*release*
QFX5100 Switches	14.1X53-D10 through 14.1X53-D27	No	jsdn-i386-*release*
QFX5100 Switches	14.1X53-D30 and later	Yes	–

Related Documentation

- *Installing Open vSwitch Database Components on Juniper Networks Devices*

Features Supported on OVSDB-Managed Interfaces

Table 4 on page 10 lists features supported on Open vSwitch Database (OVSDB)-managed interfaces of QFX5100 switches, the release in which support is introduced, the environment in which the feature is supported, and where to find information about the feature.

Table 4: Features Supported on OVSDB-Managed Interfaces

Feature	Junos OS Release in Which Feature Is Introduced	Supported Environment	Where to Find Information
Classifiers	14.1X53-D30	Contrail	Understanding CoS on OVSDB-Managed VXLAN Interfaces
DSCP rewrite rules	14.1X53-D30	Contrail	Understanding CoS on OVSDB-Managed VXLAN Interfaces
Firewall filters	14.1X53-D30	Contrail	Example: Applying a Firewall Filter to OVSDB-Managed Interfaces
MAC limiting NOTE: MAC move limiting is not supported on OVSDB-managed interfaces.	14.1X53-D30	Contrail	Understanding MAC Limiting and MAC Move Limiting for Port Security
Schedulers	14.1X53-D30	Contrail	Understanding CoS on OVSDB-Managed VXLAN Interfaces
Storm control	14.1X53-D26	Contrail	Understanding Storm Control
Two-rate three-color markers	14.1X53-D30	Contrail	Example: Applying a Policer to OVSDB-Managed Interfaces

Understanding the OVSDB Protocol Running on Juniper Networks Devices

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

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

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

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

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

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

Related Documentation

- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

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

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

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

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

can also explicitly configure connections with the other controllers in the cluster, using the Junos OS CLI.

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

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

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

  NOTE: **The SSL connection requires a private key and certificates, which must be stored in the /var/db/certs directory of the Juniper Networks device.** See "Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers" on page 27.

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

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

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

Related Documentation

- Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 10

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

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

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

- Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic that originates in an OVSDB-managed VXLAN and is forwarded to interfaces within the same VXLAN

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

- 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

NOTE: **Only MX Series routers support the Layer 3 multicast traffic scenario.**

By default, Layer 2 BUM traffic that originates in an OVSDB-managed VXLAN is handled by one or more software virtual tunnel endpoints (VTEPs) service nodes, or top-of-rack service nodes (TSNs) in the same VXLAN. (This topic refers to the software VTEPs, service nodes, and TSNs collectively as *replicators*.) The table for remote multicast MAC addresses in the OVSDB schema for physical devices contains only one entry that has the keyword **unknown-dst** as the MAC string and a list of replicators.

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

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

NOTE: **Ingress node replication is supported on all Juniper Networks devices that support OVSDB except the QFX5100 switch.**

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 MX Series router 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*

Understanding Automatically Configured VXLANs in an OVSDB Environment

NOTE: This topic applies only to QFX5100 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 and after the automatic configuration.

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

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

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

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

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

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

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

Although the configuration of OVSDB-managed VXLANs is automated, there are some tasks that you must perform before and after the automatic configuration. Table 5 on page 15 includes a sequentially ordered workflow of tasks and events for the automatic configuration of OVSDB-managed VXLANs in an NSX environment, while Table 6 on page 17 includes the equivalent information for a Contrail environment. Your familiarity with these workflows will ensure that the automatic configuration of OVSDB-managed VXLANs is properly implemented.

In Table 5 on page 15, the NSX controller and Juniper Networks switch handle the events described in workflow numbers 4, 6, and 7. You must perform the tasks described in workflow numbers 1, 2, 3, 5, and 8. If you perform a task in a different order than that outlined in Table 5 on page 15, the automatic configuration might not work or the automatically configured OVSDB-managed VXLAN might not become functional.

Table 5: Workflow of Tasks and Events for the Automatic Configuration of OVSDB-Managed VXLANs in an NSX Environment

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
1	Enable the Juniper Networks switch to automatically configure an OVSDB-managed VXLAN.	You must manually enable this capability by entering the **set switch-options ovsdb-managed** configuration mode command on the switch.	–
2	On the Juniper Networks switch, configure each physical interface that is connected to a physical server so that the interface is managed by OVSDB.	For each physical interface, you must manually enter the **set protocols ovsdb interfaces *interface-name*** configuration mode command.	When entering the interface name, you do not need to include a logical unit number.
3	For each OVSDB-managed VXLAN that you want to implement, configure a logical switch.	You must manually configure the logical switch by using NSX Manager or the NSX API. See the documentation that accompanies NSX Manager or the NSX API.	A universally unique identifier (UUID) for the logical switch is automatically generated.

Table 5: Workflow of Tasks and Events for the Automatic Configuration of OVSDB-Managed VXLANs in an NSX Environment *(continued)*

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
4	Relevant information about the logical switch is pushed to the Juniper Networks switch.	The NSX controller pushes relevant information to the logical switch table in the OVSDB schema for physical devices. This schema resides in the Juniper Networks switch.	–
5	Create the following entities: • For each Juniper Networks switch that you deploy as a hardware VTEP, you create a gateway. • For each OVSDB-managed interface that you configured in workflow number 2, you create a gateway service. • For each interface that you plan to implement for a VXLAN, configure a logical switch port.	You must manually configure these entities by using NSX Manager or the NSX API. See the documentation that accompanies NSX Manager or the NSX API. Also see *VMware NSX Configuration for Juniper Networks Devices Functioning as Virtual Tunnel Endpoints*.	–
6	Relevant information about the gateway service and logical switch port are pushed to the Juniper Networks switch.	The NSX controller pushes this information to the Juniper Networks switch.	–
7	A corresponding VXLAN is automatically created. Based on the gateway service and logical switch port configured in NSX Manager or the NSX API, one or more interfaces are also created and associated with the VXLAN.	The Juniper Networks switch automatically creates the VXLAN and interface configuration.	For the name of the VXLAN, the Juniper Networks switch uses the UUID of the logical switch.
8	(Recommended) Verify that the logical switch, corresponding VXLAN, and associated interfaces are configured properly and are operational.	You can enter the **show ovsdb logical-switch** operational mode command on the Juniper Networks switch. In the output, check the **Flags** field for the logical switches that you configured as described in workflow number 2 to ensure that it displays **Created by both**.	If the output of the **show ovsdb logical-switch** operational mode command displays a state other than **Created by both**, see "Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN" on page 51.

In Table 6 on page 17, the Contrail controller and Juniper Networks switch handle the events described in workflow numbers 5, 8, and 9. You must perform all other tasks described in the table. If you perform a task in a different order than that outlined in Table 6 on page 17, the automatic configuration might not work or the automatically configured OVSDB-managed VXLAN might not become functional.

NOTE: Although you can perform the Contrail configurations outlined in Table 6 on page 17 in the Contrail Web user interface or in the Contrail REST API, Table 6 on page 17 only describes how to perform tasks in the Contrail Web user interface.

Table 6: Workflow of Tasks and Events for the Automatic Configuration of OVSDB-Managed VXLANs in a Contrail Environment

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
1	On the Juniper Networks switch, configure a unique hostname for the switch.	You must manually enter the **set system host-name *host-name*** configuration mode command on the switch.	If implementing a virtual chassis, be aware that all members of the virtual chassis must have the same hostname.
2	Enable the Juniper Networks switch to automatically configure an OVSDB-managed VXLAN.	You must manually enable this capability by entering the **set switch-options ovsdb-managed** configuration mode command on the switch.	–
3	On the Juniper Networks switch, configure each physical interface that is connected to a physical server so that the interface is managed by OVSDB.	For each physical interface, you must manually enter the **set protocols ovsdb interfaces *interface-name*** configuration mode command.	When entering the interface name, you do not need to include a logical unit number.
4	For each OVSDB-managed VXLAN that you want to implement, configure a virtual network in the Contrail Web user interface.	You must manually configure the virtual network by navigating to Configure > Networking > Networks. See Creating a Virtual Network.	See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.
5	Relevant information about the virtual network is pushed to the Juniper Networks switch.	The Contrail controller pushes relevant information to the logical switch table in the OVSDB schema for physical devices. This schema resides in the Juniper Networks switch.	–

Table 6: Workflow of Tasks and Events for the Automatic Configuration of OVSDB-Managed VXLANs in a Contrail Environment *(continued)*

Workflow Number	Task or Event	How Task or Event Is Handled	More Information About Task or Event
6	For each interface that you plan to implement for a VXLAN, configure a logical interface.	In the Contrail Web user interface, you must manually configure the logical interface by navigating to Configure > Physical Devices > Interfaces. For information about configuring a logical interface, see Using TOR Switches and OVSDB to Extend the Contrail Cluster to Other Instances.	See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.
7	For each Juniper Networks switch that you deploy as a hardware VTEP, you create a physical router.	In the Contrail Web user interface, you must manually configure the physical router by navigating to Configure > Physical Devices > Physical Routers. For information about configuring a physical router, see Using TOR Switches and OVSDB to Extend the Contrail Cluster to Other Instances.	See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.
8	Relevant information about the logical interfaces is pushed to the Juniper Networks switch.	The Contrail controller pushes this information to the Juniper Networks switch.	–
9	A corresponding VXLAN is automatically created. Based on the logical interface configured in the Contrail Web user interface, one or more interfaces are also created and associated with the VXLAN.	The Juniper Networks switch automatically creates the VXLAN and interface configurations.	For the name of the VXLAN, the Juniper Networks switch uses the prefix "Contrail-" and the UUID of the virtual network.
10	(Recommended) Verify that the virtual network, corresponding VXLAN, and interfaces are configured properly and are operational.	You can enter the **show ovsdb logical-switch** operational mode command on the Juniper Networks switch. In the output, check the **Flags** field for the virtual network that you configured as described in workflow number 4 to ensure that it displays **Created by both**.	If the output of the **show ovsdb logical-switch** operational mode command displays a state other than **Created by both**, see "Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN" on page 51.

What the Juniper Networks Switch Actually Creates

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

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

Note the following meanings for this sample configuration:

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

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

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

- Automatic Association of a Trunk Interface Supporting Untagged Packets to an Automatically Created VXLAN on page 19
- Automatic Association of a Trunk Interface Supporting Tagged Packets to an Automatically Created VXLAN on page 20

Automatic Association of a Trunk Interface Supporting Untagged Packets to an Automatically Created VXLAN

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

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

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

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

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

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

Instead of automatically configuring physical interface ge-1/0/0 as an access interface, which typically handles untagged packets, the Juniper Networks switch configures it as a trunk interface. The intent of this configuration is to support the division of physical interface ge-1/0/0 into multiple logical interfaces, some of which are associated with VXLANs that have untagged packets (for example, logical interface ge-1/0/0.0) and some of which are associated with VXLANs that handle tagged packets (for example, logical interfaces ge-1/0/0.10 and ge-1/0/0.20).

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

Automatic Association of a Trunk Interface Supporting Tagged Packets 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 tag. In this situation, the interface that connects the Juniper Networks switch to a physical server in an OVSDB environment is a trunk interface. 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 ID that you specified when configuring either the logical switch port (NSX), or the logical interface (Contrail). If you specified a valid VLAN ID other than 0 in either configuration, the switch creates a trunk interface that can handle tagged packets. (If you specified **0** as the VLAN ID, the switch creates a trunk interface that handles untagged packets.)

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

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

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

Related Documentation

- Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 10
- show ovsdb logical-switch on page 78

OVSDB Schema for Physical Devices

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

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

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

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

Table 7 on page 22 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 7 on page 22 also indicates when a particular table is not used in the Contrail environment.

Table 7: OVSDB Schema Tables

Table Name	Description	Source of Information	Command
Global table	Includes the top-level configuration for the Juniper Networks device.	Juniper Networks device	–
Manager table	Includes information about each SDN controller that is connected to the Juniper Networks device.	Juniper Networks device	show ovsdb controller
Physical switch table	Includes information about a Juniper Networks device that functions as a hardware VTEP. This table includes information only for the device on which the table resides.	Juniper Networks device	–
Physical port table	Includes information about OVSDB-managed interfaces.	Juniper Networks device	show ovsdb interface
Logical switch table	Includes the following information: • Logical switches, which you configured in a VMware NSX environment, or a virtual networks, which you configured in a Contrail environment. • The equivalent VXLANs, which were configured on the Juniper Networks device.	• SDN controller • Juniper Networks device	show ovsdb logical-switch
Logical binding statistics table	Includes statistics for OVSDB-managed interfaces.	Juniper Networks device	show ovsdb statistics interface
Physical locator table	Includes information about Juniper Networks devices configured as hardware VTEPs, software VTEPs, and service nodes in an NSX environment.	Juniper Networks device	show ovsdb virtual-tunnel-end-point
Physical locator set table	Includes a list of software VTEPs, service nodes, or top-of-rack service nodes (TSNs) for a logical switch.	Juniper Networks device	–

Table 7: OVSDB Schema Tables *(continued)*

Table Name	Description	Source of Information	Command
Unicast MACs remote table	Reachability information, including unicast MAC addresses, for entities in the virtualized network.	SDN controller	show ovsdb mac
Unicast MACs local table	Reachability information, including unicast MAC addresses, for entities in the physical network.	Juniper Networks device	show ovsdb mac
Multicast MACs remote table	Includes only one row. In this row, the MAC column includes the keyword **unknown dst** along with a list of software VTEPs, service nodes, or TSNs, which handle multicast traffic.	SDN controller	show ovsdb mac
Multicast MACs local table	NOTE: Only QFX5100 switches support this table. 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.	Juniper Networks device	show ovsdb mac

Related Documentation

- Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 10
- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

VXLAN Constraints on QFX5100 Switches

When configuring VXLANs on QFX5100 switches, be aware of the constraints in the following list. In this list, “Layer 3 side” refers to a network-facing interface that performs VXLAN encapsulation and de-encapsulation, and “Layer 2 side” refers to a server-facing interface that is a member of a VLAN that is mapped to a VXLAN.

- You can use VXLANs on a Virtual Chassis or Virtual Chassis Fabric if all of the members are QFX5100 switches. You cannot use VXLANs if any of the members is not a QFX5100 switch.
- VXLAN configuration is supported only in the default routing instance.

- A QFX 5100 switch cannot route traffic between different VXLANs.
- A physical interface cannot be a member of a VLAN and a VXLAN. That is, an interface that performs VXLAN encapsulation and de-encapsulation cannot also be a member of a VLAN. For example, if a VLAN that is a mapped to a VXLAN is a member of trunk port xe-0/0/0, any other VLAN that is a member of xe-0/0/0 must also be assigned to a VXLAN.
- Multichassis link aggregation groups (MC-LAGS) are not supported with VXLAN.
- IP fragmentation and defragmentation are not supported on the Layer 3 side.
- The following features are not supported on the Layer 2 side:
  - STP (any variant)
  - IGMP snooping
  - storm control
- Access port security features are not supported with VXLAN. For example, the following features are not supported:
  - DHCP snooping
  - dynamic ARP inspection
  - MAC limiting and MAC move limiting
- Ingress node replication is not supported. (You must use PIM to advertise and learn about reachability if you do not use VMware's NSX controller.)
- PIM-BIDIR and PIM-SSM are not supported with VXLANs.
- Class of service (CoS) features are not supported with VXLANs.
- If you configure a port-mirroring instance to mirror traffic egressing from an interface that performs VXLAN encapsulation, the source and destination MAC addresses of the mirrored packets are invalid. The original VXLAN traffic is not affected.

Related Documentation

- Understanding VXLANs on page 3
- *Examples: Manually Configuring VXLANs on QFX Series Switches*
- *Configuring VXLANs on a QFX5100 Switch*

PART 2

Configuration

CHAPTER 2

Configuring OVSDB and VXLAN

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

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

- **vtep-privkey.pem**
- **vtep-cert.pem**
- **ca-cert.pem**

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

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

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

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

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

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

   # ovs-pki init

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

   # ovs-pki req+sign vtep

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

Related Documentation

- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

Setting Up the OVSDB Management Protocol on Juniper Networks Devices

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

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

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

Before setting up OVSDB on a Juniper Networks device, you must create an SSL private key and certificate, if they don't already exist, and install them in the **/var/db/certs**

directory of the Juniper Networks device. See "Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers" on page 27.

To set up OVSDB on a Juniper Networks device:

1. Specify the IP address of the SDN controller.

   ```
   [edit protocols ovsdb]
   user@host# set controller ip-address
   ```

2. Specify SSL as the protocol that secures the connection between the Juniper Networks device and the SDN controller.

   ```
   [edit protocols ovsdb]
   user@host# set controller ip-address protocol ssl
   ```

3. Set the number of the port over which the connection to the SDN controller is made.

   ```
   [edit protocols ovsdb]
   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 ovsdb]
   user@host# set controller ip-address inactivity-probe-duration milliseconds
   ```

5. (Optional) Specify (in milliseconds) how long the device must wait before it can try to connect to the SDN controller again if the previous attempt failed.

   ```
   [edit protocols ovsdb]
   user@host# set controller ip-address maximum-backoff-duration milliseconds
   ```

6. (Optional) Repeat Steps 1 through 5 to configure a connection to an additional SDN controller in the NSX environment.

7. Specify that each physical interface that is connected to a physical server is managed by OVSDB.

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

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

NOTE: After completing this procedure, if you have any Juniper Networks device except a QFX5100 switch, you must manually configure OVSDB-managed VXLANs. See *Configuring OVSDB-Managed VXLANs*.

QFX5100 switches support the automatic configuration of OVSDB-managed VXLANs. On these switches, although the OVSDB-managed VXLAN configuration is automated, there are tasks that you must perform before and after the automatic configuration. See one of the following topics:

- For Junos OS Releases 14.1X53-D15 through 14.1X53-D25, see *Understanding Automatically Configured Virtual Extensible LANs in an Open vSwitch Database Environment*.
- For Junos OS Releases 14.1X53-D26 and later, see "Understanding Automatically Configured VXLANs in an OVSDB Environment" on page 14.

Related Documentation

- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs

When extending a Contrail topology to include physical servers that are connected to a Juniper Networks switch that supports the Open vSwitch Database (OVSDB) management protocol and Virtual Extensible LANs (VXLANs), you must perform the following tasks in the Contrail Web user interface:

- For each OVSDB-managed VXLAN that you plan to implement on the Juniper Networks switch, configure an equivalent virtual network. Also configure a logical interface to associate with the virtual network.
- Configure a physical router, which enables the Contrail controller to recognize the Juniper Networks device as a hardware virtual tunnel endpoint (VTEP).

This topic provides a high-level summary of the tasks that you must perform to configure a virtual network, logical interface, and physical router. Although you can create these virtual entities in either the Contrail Web user interface or in the Contrail REST API, this topic only describes how to perform the tasks in the Contrail Web user interface. Also, this topic does not include a complete procedure for each task. Rather, it includes key configuration details for ensuring the correct configuration of the virtual entities so that they function properly with their counterparts in the physical network.

For more information about performing the tasks described in this topic, see Creating a Virtual Network and Using TOR Switches and OVSDB to Extend the Contrail Cluster to Other Instances.

This topic describes the following tasks:

- Creating a Virtual Network on page 30
- Creating a Logical Interface on page 31
- Creating a Physical Router on page 32

Creating a Virtual Network

For each OVSDB-managed VXLAN that you plan to implement on a Juniper Networks switch, configure an equivalent virtual network in the Contrail Web user interface.

Table 8 on page 31 provides key configuration details to keep in mind when you configure a virtual network.

Table 8: Key Configuration Details for Creating a Virtual Network in the Contrail Web User Interface

Contrail Web User Interface Navigation	Configuration Field	Configuration Details
Configure > Networking > Networks	VXLAN Identifier in Advanced Options	• In the Contrail Web user interface, you can configure a VXLAN identifier mode so that VXLAN identifiers are either automatically configured or user configured. We recommend setting this mode to user configured, which enables you to initially configure a VXLAN identifier and modify it later as needed. • In a Junos OS environment, a VXLAN identifier is also known as a *VXLAN network identifier (VNI)*.

Creating a Logical Interface

In the Contrail Web user interface, you must create a logical interface for each virtual network that you created.

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

- The OVSDB-managed physical 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 OVSDB Management Protocol on Juniper Networks Devices" on page 28.
- The virtual network with which you want to associate the logical interface.

Table 9 on page 31 provides a summary of key configuration details to keep in mind when you configure a logical interface.

Table 9: Key Configuration Details for Creating a Logical Interface in the Contrail Web User Interface

Contrail Web User Interface Navigation	Configuration Field	Configuration Details
Configure > Physical Devices > Interfaces	Type	Select **Logical**.
	Name	When you specify a logical interface name, use the same naming convention for configuring a logical interface in the Junos OS CLI. A sample logical interface name is ge-1/0/0.10.
	Parent	Select an OVSDB-managed physical interface that is configured on the Juniper Networks device.
	Logical Interface Type	Select **L2 Server**.

Table 9: Key Configuration Details for Creating a Logical Interface in the Contrail Web User Interface *(continued)*

Contrail Web User Interface Navigation	Configuration Field	Configuration Details
	VLAN ID	• If you want the logical interface to handle untagged packets, specify **0**. • If you want the logical interface to handle tagged packets, specify **3** through **4000**. NOTE: VLAN IDs 1, 2, and 4094 are reserved. As a result, you must not specify these VLAN IDs.
	Virtual Network	Select the virtual network with which you want to associate the logical interface.

Creating a Physical Router

In the Contrail Web user interface, you must create a physical router, which enables the Contrail controller to recognize a Juniper Networks switch as a hardware VTEP.

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

- On the Juniper Networks switch, configure a hostname for the switch (**set system host-name *hostname***), an IP address for the management interface (**set interfaces *management-interface-name* unit 0 family inet address *ip-address/destination prefix***), and an IP address for the loopback interface (**set interfaces lo0 unit 0 family inet address *ip-address/destination prefix***). Also, set the loopback interface as the interface that identifies the switch as the hardware VTEP (**set vtep-source interface lo0.0**).
- Configure one or more virtual networks with which you want to associate the Juniper Networks switch.

Table 10 on page 32 provides a summary of key configuration details to keep in mind when you configure a physical router.

Table 10: Key Configuration Details for Creating a Physical Router in the Contrail Web User Interface

Contrail Web User Interface Navigation	Configuration Field	Configuration Details
Configure > Physical Devices > Physical Routers	Name	Specify the hostname that you configured on the Juniper Network switch.
	Management IP	Specify the IP address of the management interface on the Juniper Networks switch. The connection with the Contrail controller is made over this interface.
	VTEP address	Specify the IP address of the loopback interface on the Juniper Networks switch.

Table 10: Key Configuration Details for Creating a Physical Router in the Contrail Web User Interface *(continued)*

Contrail Web User Interface Navigation	Configuration Field	Configuration Details
	Virtual Network	Specify one or more virtual networks that serve as counterparts to the OVSDB-managed VXLANs that the Juniper Networks switch automatically configures.

Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections in a Contrail Environment (Trunk Interfaces That Support Untagged Packets)

In a physical network, a Juniper Networks switch that supports Virtual Extensible LAN (VXLAN) can function as a hardware virtual tunnel endpoint (VTEP). In this role, the Juniper Networks switch encapsulates in VXLAN packets Layer 2 Ethernet frames received from software applications that run directly on a physical server. 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 them to virtual machines (VMs).

In this VXLAN environment, you can also include Contrail controllers and implement the Open vSwitch Database (OVSDB) management protocol on the Juniper Networks switch that functions as a hardware VTEP.

The Junos OS implementation of OVSDB provides a means through which Contrail controllers and Juniper Networks switches can exchange MAC addresses of entities in the physical and virtual networks. This exchange of MAC addresses enables the Juniper Networks switch 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 switch as a hardware VTEP, which serves as a Layer 2 gateway, and set up this device with an OVSDB connection to a Contrail controller.

In this example, only one VXLAN is deployed. Given this scenario, the packets exchanged between an application running on a physical server and a VM in the VXLAN are untagged. Therefore, in this example, a trunk interface is used for the connection between the physical server and the switch, as well as a native VLAN. The native VLAN enables the trunk interface to handle the untagged packets.

Requirements

This example includes the following hardware and software components:

- A physical server on which software applications directly run.
- A QFX5100 switch running Junos OS Release 14.1X53-D30 or later.
- On the Juniper Networks switch, physical interface ge-1/0/0 provides a connection to physical server 1.
- A Contrail controller.
- A top-of-rack service node (TSN) that handles the replication and forwarding of Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic within the VXLAN used in this example.

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

- The Contrail Web user interface.
- A vRouter that includes VMs managed by a hypervisor, which includes a software VTEP.

NOTE: **All components in the Contrail environment (Contrail controller, TSN, Contrail Web user interface, and vRouters must be running Contrail Release 2.20.**

For information about the Contrail components, see Using TOR Switches and OVSDB to Extend the Contrail Cluster to Other Instances.

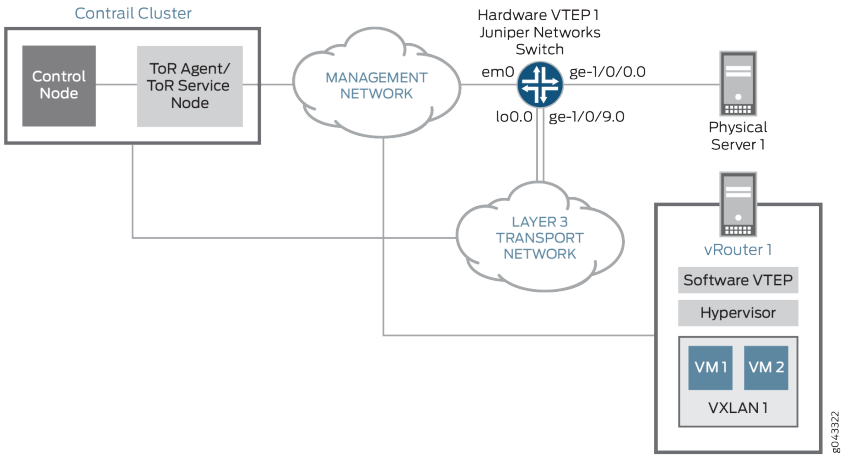
Before you begin:

- Create an SSL private key and certificate, if they do not already exist. The private key and certificate must be installed in the **/var/db/certs** directory of the Juniper Networks switch. See "Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers" on page 27.

Overview and Topology

Figure 2 on page 35 shows a topology in which a software application running directly on physical server 1 in the physical network needs to communicate with virtual machine VM 1 in VXLAN 1, and vice versa.

Figure 2: VXLAN-OVSDB Layer 2 Gateway Topology with a Contrail Controller



To establish communication between the software application on physical server 1 and VM 1 in VXLAN 1, a connection with a Contrail controller is configured on the management interface of the Juniper Networks switch.

Some entities in the VXLAN-OVSDB topology must be configured in both the Contrail Web user interface and on the Juniper Networks switch. Table 11 on page 35 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 the Contrail Web user interface can differ from the term used for essentially the same entity that is configured on the Juniper Networks switch. To prevent confusion, Table 11 on page 35 shows the Contrail Web user interface and Junos OS entities side-by-side.

Table 11: Contrail and Junos OS Entities That Must Be Configured for a VXLAN Layer 2 Gateway Topology with OVSDB Connections and Trunk Interfaces Supporting Untagged Packets

Entity	Entity to Be Configured in the Contrail Web User Interface	Entity to Be Configured on the Juniper Networks Switch
VXLAN 1	Virtual network for VXLAN 1	VXLAN 1 NOTE: The Juniper Networks switch automatically configures this VXLAN.
Physical interface (ge-1/0/0) between physical server 1 and Juniper Networks switch	–	OVSDB management. Specify that interface ge-1/0/0 is managed by OVSDB.

Table 11: Contrail and Junos OS Entities That Must Be Configured for a VXLAN Layer 2 Gateway Topology with OVSDB Connections and Trunk Interfaces Supporting Untagged Packets *(continued)*

Entity	Entity to Be Configured in the Contrail Web User Interface	Entity to Be Configured on the Juniper Networks Switch
One logical interface (ge-1/0/0.0) associated with VXLAN 1	One logical interface for VXLAN 1. For this interface, specify VLAN ID 0. NOTE: A VLAN ID of 0 indicates that the interface must handle untagged packets.	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)	Physical router	Hardware VTEP functionality. Configure the Juniper Networks switch to function as a hardware VTEP.

In the Contrail Web user interface, a virtual network is configured. In this configuration, a VXLAN identifier of 100 is specified. Also, the universally unique identifier (UUID) assigned to the virtual network is Contrail-28805c1d-0122-495d-85df-19abd647d772. Based on this configuration, the Juniper Networks switch automatically creates the following configuration for a Junos OS-equivalent VXLAN:

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

Based on the logical interface configuration (VLAN number 0) in the Contrail Web user interface, the Juniper Networks switch automatically creates the following configuration for a Junos OS-equivalent interface:

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

This sample configuration does the following:

- Configures physical interface ge-1/0/0 as a Layer 2 trunk interface.
- Creates a native VLAN with an ID of 4094.
- Creates logical interface ge-1/0/0.0, and specifies that it is a member of the native VLAN.
- Associates logical interface ge-1/0/0.0 with VXLAN Contrail-28805c1d-0122-495d-85df-19abd647d772.

As a result of the above configuration, logical interface ge-1/0/0.0 handles incoming untagged packets.

Table 12 on page 37 provides a summary of the VXLAN-OVSDB topology components that are configured on the Juniper Networks switch and the configuration settings for each component.

Table 12: Components Configured on the Juniper Networks Switch (Hardware VTEP) in a VXLAN Layer 2 Gateway Topology with OVSDB Connections and Trunk Interfaces Supporting Untagged Packets

Component	Setting
Contrail controller	IP address: 10.94.184.1
OVSDB-managed physcal interface	Interface name: ge-1/0/0 Native VLAN ID: 4094
VXLAN 1 and associated logical interface	NOTE: The Juniper Networks switch automatically configures the VXLAN and associated logical interface, which are based on the virtual network and associated logical interface configurations in the Contrail Web user interface. Therefore, no manual configuration is required. VXLAN name: Contrail-28805c1d-0122-495d-85df-19abd647d772 VNI: 100 Logical interface name: ge-1/0/0.0 Interface type: trunk Member of native VLAN 4094 Associated with VXLAN Contrail-28805c1d-0122-495d-85df-19abd647d772
OVSDB tracing operations	Filename: /var/log/ovsdb File size: 10 MB Flag: All
Hardware VTEP	Hostname: hw-vtep1 Source interface: loopback (lo0.0) Source IP address: 10.17.17.17/32
Handling of Layer 2 BUM traffic in VXLAN Contrail-28805c1d-0122-495d-85df-19abd647d772	TSN NOTE: By default, one or more TSNs handle Layer 2 BUM traffic within a VXLAN; therefore, no manual configuration is required.

Non-OVSDB and Non-VXLAN Configuration

CLI Quick Configuration

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

```
set interfaces ge-1/0/9 unit 0 family inet address 10.40.40.1/24
set routing-options static route 10.19.19.19/32 next-hop 10.40.40.2
set routing-options router-id 10.17.17.17
```

```
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-1/0/9.0
```

Step-by-Step Procedure

To configure the Layer 3 network over which the packets exchanged between physical server 1 and VM 1 are tunneled:

1. Configure the Layer 3 interface.

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

2. Set the routing options.

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

3. Configure the routing protocol.

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

OVSDB and VXLAN Configuration

CLI Quick Configuration

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

```
set system host-name hw-vtep1
set switch-options ovsdb-managed
set protocols ovsdb controller 10.94.184.1
set protocols ovsdb interfaces ge-1/0/0
set protocols ovsdb traceoptions file ovsdb
set protocols ovsdb traceoptions file size 10m
set protocols ovsdb traceoptions flag all
set interfaces lo0 unit 0 family inet address 10.17.17.17/32 primary
set interfaces lo0 unit 0 family inet address 10.17.17.17/32 preferred
set switch-options vtep-source-interface lo0.0
```

Step-by-Step Procedure

To configure the Juniper Networks switch as a hardware VTEP with an OVSDB connection to a Contrail controller:

1. Configure a unique hostname for the Juniper Networks switch.

   ```
   [edit system]
   user@switch# set host-name hw-vtep1
   ```

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

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

3. Configure a connection with a Contrail controller.

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

4. 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
```

5. 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
```

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

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

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

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

8. In the Contrail Web user interface, configure a virtual network for VXLAN 1. See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.

9. In the Contrail Web user interface, configure a logical interface for the virtual network that you created in Step 6. See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.

10. In the Contrail Web user interface, configure a physical router, which enables the Contrail controller to recognize the Juniper Networks switch as a VTEP. See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.

Verification

Confirm that the configuration is working properly:

Verifying the Logical Switch Configuration

Purpose In the Contrail Web user interface, you configured a virtual network for VXLAN 1. Using the same terminology as in the OVSDB schema for physical devices, the virtual network

is also known as a *logical switch*. Verify that the configuration of the logical switch with the UUID of Contrail-28805c1d-0122-495d-85df-19abd647d772 is present in the OVSDB schema and that the state (**Flags**) of the logical switch is **Created by both**.

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

```
user@switch> show ovsdb logical-switch
Logical switch information:
Logical Switch Name: Contrail-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 virtual network was configured in the Contrail Web user interface, and that the Juniper Networks switch automatically created the corresponding VXLAN. In this state, the virtual network and the VXLAN are operational.

If the state of the logical switch is something other than **Created by both**, see "Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN" on page 51.

Verifying the MAC Address of VM 1

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

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

```
user@switch> show ovsdb mac remote
Logical Switch Name: Contrail-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 Contrail-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 Contrail Controller Connection

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

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

```
user@switch> show ovsdb 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 state of the connection is **up**, in addition to other information about the connection. The **up** state indicates that OVSDB is enabled on the Juniper Networks switch.

Verifying the OVSDB-Managed Interface

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

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

```
user@switch> show ovsdb interface
Interface  VLAN ID Bridge-domain
ge-1/0/0   0       Contrail-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 Contrail-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 in a Contrail Environment (Trunk Interfaces That Support Tagged Packets) on page 41

Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections in a Contrail Environment (Trunk Interfaces That Support Tagged Packets)

In a physical network, a Juniper Networks switch that supports Virtual Extensible LANs (VXLANs) can function as a hardware virtual tunnel endpoint (VTEP). In this role, the Juniper Networks switch encapsulates in VXLAN packets Layer 2 Ethernet frames received from software applications that run directly on a physical server. 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 Contrail controllers and implement the Open vSwitch Database (OVSDB) management protocol on the Juniper Networks switch that functions as a hardware VTEP.

The Junos OS implementation of OVSDB provides a means through which Contrail controllers and Juniper Networks switches can exchange MAC addresses of entities in the physical and virtual networks. This exchange of MAC addresses enables the Juniper Networks switch 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 switch in the physical network.

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

In this example, two VXLANs are deployed. Given this scenario, the packets exchanged between the applications that are running on a physical server and the VMs in the VXLANs are tagged. As a result, trunk interfaces, which can handle the tagged packets, are used for the connection between the physical server and the Juniper Networks switch.

Requirements

This example includes the following hardware and software components:

- A physical server on which software applications directly run.
- A QFX5100 switch running Junos OS Release 14.1X53-D30 or later.
- On the Juniper Networks switch, physical interface ge-1/0/0 provides a connection to physical server 1.
- A Contrail controller.
- A top-of-rack service node (TSN) that handles the replication and forwarding of Layer 2 broadcast, unknown unicast, and multicast (BUM) traffic within the two VXLANs used in this example.

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

- The Contrail Web user interface.
- Two vRouters that include VMs. Each vRouter is managed by a hypervisor, and each hypervisor includes a software VTEP.

NOTE: All components in the Contrail environment (Contrail controller, TSN, Contrail Web user interface, and vRouters) must be running Contrail Release 2.20.

For information about the Contrail components, see Using TOR Switches and OVSDB to Extend the Contrail Cluster to Other Instances.

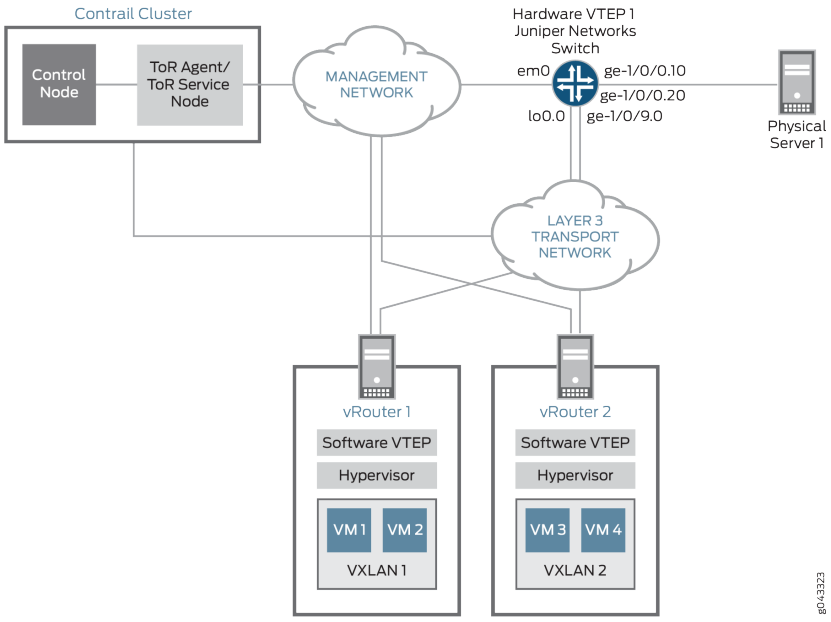
Before you begin:

- Create an SSL private key and certificate, if they do not already exist. The private key and certificate must be installed in the **/var/db/certs** directory of the Juniper Networks switch. See "Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers" on page 27.

Overview and Topology

Figure 2 on page 35 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. To enable this communication, a Juniper Networks switch is configured as hardware VTEP 1. Further, the Juniper Networks switch is connected to a Contrail controller by way of management interface em0 on the switch.

Figure 3: VXLAN/OVSDB Layer 2 Gateway Topology



Some entities in the VXLAN-OVSDB topology must be configured in both the Contrail Web user interface and on the Juniper Networks switch. Table 13 on page 44 provides a summary of the entities that must be configured and where they must be configured.

NOTE: The term used for an entity that is configured in the Contrail Web user interface can differ from the term used for essentially the same entity that is configured on the Juniper Networks switch. To prevent confusion, Table 13 on page 44 shows the Contrail Web user interface and the Junos OS entities side by side.

Table 13: Contrail and Junos OS Entities That Must Be Configured for a VXLAN Layer 2 Gateway Topology with OVSDB Connections and Trunk Interfaces Supporting Tagged Packets

Entity	Entity to Be Configured in the Contrail Web User Interface	Entity to Be Configured on the Juniper Networks Switch
VXLAN 1 VXLAN 2	Virtual network for VXLAN 1 Virtual network for VXLAN 2	VXLAN 1 VXLAN 2 NOTE: The Juniper Networks switch automatically configures these VXLANs.
Physical interface ge-1/0/0 between physical server 1 and Juniper Networks switch	–	OVSDB management. Specify that interface ge-1/0/0 is managed by OVSDB.
One logical interface (ge-1/0/0.10) associated with VXLAN 1 One logical interface (ge-1/0/0.20) associated with VXLAN 2	One logical interface for VXLAN 1. For this interface, specify VLAN ID 10. One logical interface for VXLAN 2. For this interface, specify VLAN ID 20. NOTE: A VLAN ID from 3 through 4000 indicates that the interface must handle tagged packets.	One logical interface (ge-1/0/0.10) for VXLAN 1. One logical interface (ge-1/0/0.20) for VXLAN 2. NOTE: The Juniper Networks switch automatically configures these logical interfaces.
Juniper Networks switch (hardware VTEP 1)	Physical router	Hardware VTEP functionality. Configure the Juniper Networks switch to function as a hardware VTEP.

Based on the configuration of the entities in the Contrail Web user interface as described in Table 13 on page 44, the Juniper Networks switch automatically creates VXLANs 1 and 2 and their associated logical interfaces. Table 14 on page 45 provides the relevant Contrail Web user interface configuration and the resulting VXLANs and associated logical interfaces that the Juniper Networks switch automatically configures.

Table 14: Contrail Web User Interface Configurations and Automatic Configurations by Juniper Networks Switch

Contrail Web User Interface Configuration: Virtual Network and Logical Interface	VXLAN and Associated Logical Interface Automatically Configured by the Juniper Networks Switch
Virtual network configuration: UUID: Contrail-28805c1d-0122-495d-85df-19abd647d772 VXLAN Identifier: 100 Logical Interface configuration: VLAN ID: 10	For VXLAN 1: **set vlans Contrail-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 Contrail-28805c1d-0122-495d-85df-19abd647d772 interfaces ge-1/0/0.10**
Virtual network configuration: UUID: Contrail-9acc24b3-7b0a-4c2e-b572-3370c3e1acff VXLAN Identifier: 200 Logical Interface configuration: VLAN ID: 20	For VXLAN 2: **set vlans Contrail-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 Contrail-9acc24b3-7b0a-4c2e-b572-3370c3e1acff interfaces ge-1/0/0.20**

NOTE: In the Contrail environment, a numerical value that identifies a VXLAN is known as a *VXLAN identifier*. In the Junos OS environment, the same numerical value is known as a *VXLAN network identifier (VNI)*.

For VXLANs 1 and 2, the Juniper Networks switch uses the UUIDs and VXLAN Identifiers that were provided for the corresponding virtual networks.

In the logical interface configurations in the Contrail Web user interface, VLAN ID values 10 and 20 and virtual network 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 that handle tagged packets. 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, interface ge-1/0/0.10 accepts packets with a VLAN tag of 10 from VXLAN 1, and interface ge-1/0/0.20 accepts packets with a VLAN tag of 20 from VXLAN 2. On receiving packets from VXLAN 1, a VLAN tag of 100 is added to the packets, and a VLAN tag of 200 is added to packets from VXLAN 2. These tags are added to the respective packet streams to map the VLAN ID in a particular VXLAN to the corresponding VNI.

Table 12 on page 37 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 15: Components Configured on Juniper Networks Switch (Hardware VTEP) in a VXLAN Layer 2 Gateway Topology with OVSDB Connections and Trunk Interfaces Supporting Tagged Packets

Components	Settings
Contrail controller	IP address: 10.94.184.1
OVSDB-managed interface	Interface name: ge-1/0/0
VXLAN 1 and associated logical interface	NOTE: The Juniper Networks switch automatically configures the VXLAN and associated logical interface, which are based on the virtual network and associated logical interface configurations in the Contrail Web user interface. Therefore, no manual configuration is required. VXLAN name: Contrail-28805c1d-0122-495d-85df-19abd647d772 VNI: 100 Logical interface name: ge-1/0/0.10 VLAN ID: 10 Interface type: trunk
VXLAN 2 and associated logical interface	NOTE: The Juniper Networks switch automatically configures the VXLAN and associated logical interface, which are based on the virtual network and associated logical interface configurations in the Contrail Web user interface. Therefore, no manual configuration is required. VXLAN name: Contrail-VXLAN 9acc24b3-7b0a-4c2e-b572-3370c3e1acff VNI: 200 Logical interface name: ge-1/0/0.20 VLAN ID: 20 Interface type: trunk
OVSDB tracing operations	Filename: /var/log/ovsdb File size: 10 MB Flag: All
Hardware VTEP functionality	Hostname: hw-vtep1 Source interface: loopback (lo0.0) Source IP address: 10.17.17.17/32

Table 15: Components Configured on Juniper Networks Switch (Hardware VTEP) in a VXLAN Layer 2 Gateway Topology with OVSDB Connections and Trunk Interfaces Supporting Tagged Packets *(continued)*

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

Non-OVSDB and Non-VXLAN Configuration

CLI Quick Configuration

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

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

Step-by-Step Procedure

To configure the Layer 3 network over which the packets exchanged between the physical servers and VMs are tunneled:

1. Configure the Layer 3 interface.

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

2. Set the routing options.

   ```
   user@switch# set static route 10.19.19.19/32 next-hop 10.40.40.2
   user@switch# set router-id 10.17.17.17
   ```

3. Configure the routing protocol.

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

OVSDB and VXLAN Configuration

CLI Quick Configuration

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

```
set system host-name hw-vtep1
set switch-options ovsdb-managed
set protocols ovsdb controller 10.94.184.1
set protocols ovsdb interfaces ge-1/0/0
set protocols ovsdb traceoptions file ovsdb
```

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

Step-by-Step Procedure

To configure the Juniper Networks switch as a hardware VTEP with an OVSDB connection to the Contrail controller:

1. Configure a unique hostname for the Juniper Networks switch.

   ```
   [edit system]
   user@switch# set host-name hw-vtep1
   ```

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

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

3. Configure a connection with the Contrail controller.

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

4. 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
   ```

5. 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
   ```

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

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

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

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

8. In the Contrail Web user interface, configure a virtual network for VXLAN 1 and a virtual network for VXLAN 2. See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.

9. In the Contrail Web user interface, configure a logical interface for each of the virtual networks that you created in Step 8. See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.

10. In the Contrail Web user interface, configure a physical router, which enables the Contrail controller to recognize the Juniper Networks switch as a VTEP. See "Contrail Configuration for Juniper Networks Devices That Function as Hardware VTEPs" on page 30.

Verification

Confirm that the configuration is working properly:

Verifying the Logical Switch Configuration

Purpose In the Contrail Web user interface, you configured a virtual network for VXLAN 1 and a virtual network for VXLAN 2. Using the same terminology as in the OVSDB schema for physical devices, a virtual network is also known as a *logical switch*. Verify that the configuration of the logical switches with the UUIDs of Contrail-28805c1d-0122-495d-85df-19abd647d772 and Contrail-9acc24b3-7b0a-4c2e-b572-3370c3e1acff are present in the OVSDB schema and that the state (**Flags**) of each logical switch is **Created by both**.

Action Issue the **show ovsdb logical-switch** command.

```
user@switch> show ovsdb logical-switch
Logical switch information:
Logical Switch Name: Contrail-28805c1d-0122-495d-85df-19abd647d772
Flags: Created by both
VNI: 100
Num of Remote MAC: 1
Num of Local MAC: 0
Logical Switch Name: Contrail-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 configurations for the logical switches are present. The **Created by both** state indicates that the logical switches were configured in the Contrail Web user interface, and that the Juniper Networks switch automatically created the corresponding VXLANs. In this state, the virtual networks and VXLANs are operational.

If the state of the logical switches is something other than **Created by both**, see "Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN" on page 51.

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

Purpose Verify that the MAC addresses of VM 1, VM 3, and VM 4 are present in the OVSDB schema.

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

```
user@switch> show ovsdb mac remote
Logical Switch Name: Contrail-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
Logical Switch Name: Contrail-9acc24b3-7b0a-4c2e-b572-3370c3e1acff
 Mac                   IP                 Encapsulation      Vtep
  Address               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, VM 1, VM 3, and VM 4 are reachable through the Juniper Networks switch, which functions as a hardware VTEP.

Verifying the Contrail Controller Connection

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

Action Issue the **show ovsdb controller** operational mode command to verify that the Contrail controller connection state is **up**.

```
user@switch> show ovsdb 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 state of the connection is **up**, in addition to other information about the connection. 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 ovsdb interface** operational mode command, and verify that interface ge-1/0/0 is managed by OVSDB.

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

Meaning The output shows that interface ge-1/0/0 is managed by OVSDB. It also indicates that the interface is associated with VXLAN **Contrail-28805c1d-0122-495d-85df-19abd647d772**, which has a VLAN ID of **10**, and VXLAN **Contrail-9acc24b3-7b0a-4c2e-b572-3370c3e1acff**, which has a VLAN ID of **20**.

Related Documentation

- Example: Setting Up a VXLAN Layer 2 Gateway and OVSDB Connections in a Contrail Environment (Trunk Interfaces That Support Untagged Packets) on page 33

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

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

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

Cause

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

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

- On a QFX5100 switch, verify that the **set switch-options ovsdb-managed** configuration command was issued in the Junos OS CLI. Issuing this command and committing the configuration enable the Juniper Networks device to automatically create OVSDB-managed VXLANs.

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

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

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

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

- On a QFX5100 switch, two issues exist. First, despite the fact that the Juniper Networks device automatically creates OVSDB-managed VXLANs, this VXLAN was manually configured by using the Junos OS CLI. Second, a corresponding logical switch (NSX) or virtual network (Contrail) was not configured. To resolve both issues, configure a logical switch in the NSX environment or a virtual network in the Contrail environment. After the software-defined networking (SDN) controller pushes relevant logical switch or virtual network information to the Juniper Networks device, the device automatically creates a corresponding VXLAN and deletes the manually configured VXLAN.
- On all other Juniper Networks devices that support VXLAN and OVSDB, determine whether a corresponding logical switch is configured in the NSX environment or virtual network is configured in the Contrail environment. If a logical switch or virtual network is not configured, configure one, keeping in mind that a UUID is automatically generated for the logical switch or virtual network and that this UUID must be used as the name of the VXLAN. That is, the VXLAN name must be reconfigured with the logical switch or virtual network UUID.

  Another possibility is that the logical switch or virtual network configuration might exist, but the UUID of the entity might not match the VXLAN name. In the NSX or Contrail environment, check for a logical switch or virtual network, respectively, that has the same configuration as the VXLAN but has a different UUID.

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

- For a QFX5100 switch, check the configuration of the VNI in the NSX environment or the VXLAN Identifier in the Contrail environment to see whether it was changed after the Juniper Networks device created the corresponding VXLAN. If it was changed, update the VNI on the QFX5100 switch, using the Junos OS CLI.
- On all other Juniper Networks devices that support VXLAN and OVSDB, check the values of the VNI in the NSX environment or the VXLAN Identifier in the Contrail environment, and the Junos OS CLI. Change the incorrect value.

Related Documentation

- Understanding Automatically Configured VXLANs in an OVSDB Environment on page 14
- show ovsdb logical-switch on page 78
- show ovsdb commit failures on page 72
- clear ovsdb commit failures on page 70

PART 3

Configuration Statements and Operational Commands

CHAPTER 3

OVSDB Configuration Statements

controller (OVSDB)

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 running the Open vSwitch Database (OVSDB) management protocol and a software-defined networking (SDN) controller. You can connect a Juniper Networks device to more than one SDN controller for redundancy.

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

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

Connections to all SDN controllers are made on the management interface of the Juniper Networks device.

Options ***ip-address*** —IPv4 address of the SDN 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

- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

inactivity-probe-duration

Syntax inactivity-probe-duration *milliseconds*;

Hierarchy Level [edit protocols ovsdb controller]

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

Description Configure the maximum amount of time, in milliseconds, that the connection between a Juniper Networks device that supports the Open vSwitch Database (OVSDB) management protocol and a software-defined networking (SDN) controller can be inactive before an inactivity probe is sent.

Options ***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
- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

interfaces (OVSDB)

Syntax interfaces *interface-name*;

Hierarchy Level [edit protocols ovsdb]

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

Description Specify the physical interfaces on a Juniper Networks device that you want the Open vSwitch Database (OVSDB) protocol to manage. Typically, the only interfaces that need to be managed by OVSDB are interfaces that are connected to physical servers.

Options ***interface-name***—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
- *Configuring OVSDB-Managed VXLANs*

maximum-backoff-duration

Syntax maximum-backoff-duration *milliseconds*;

Hierarchy Level [edit protocols ovsdb controller]

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

Description Specify (in milliseconds) how long a Juniper Networks device that supports the Open vSwitch Database (OVSDB) management protocol waits before it tries again to connect with a software-defined networking (SDN) controller after a previous attempt has failed.

Options *milliseconds*—Number of milliseconds a Juniper Networks device waits before it tries again to connect with an SDN controller.

Range: 1000 through 4,294,967,295

Default: 1000

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

Related Documentation

- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

ovsdb

Syntax

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

Hierarchy Level [edit protocols]

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

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

The remaining statements are explained separately.

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

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

Related Documentation

- Understanding the OVSDB Protocol Running on Juniper Networks Devices on page 10
- *Configuring OVSDB-Managed VXLANs*

ovsdb-managed

Syntax ovsdb-managed;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* vxlan],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* vxlan],
[edit routing-instances *routing-instance-name* switch-options],
[edit routing-instances *routing-instance-name* vlans *vlan-name* vxlan],
[edit routing-instances *routing-instance-name* vxlan],
[edit switch-options]
[edit vlans *vlan-name* 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
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.

The specified VXLAN must have a VXLAN network identifier (VNI) configured, using the **vni** statement in the **[edit bridge-domains *bridge-domain-name* vxlan]**, **[edit routing-instance *routing-instance-name* vxlan]**, or **[edit vlans *vlan-name* vxlan]** hierarchy.

Also, for a VMware NSX environment, this implementation of OVSDB uses the multicast scheme described in "Understanding How Layer 2 BUM and Layer 3 Routed Multicast Traffic Are Handled with OVSDB" on page 13. Therefore, specifying the **multicast-group** statement in the **[edit bridge-domains *bridge-domain-name* vxlan]**, **[edit routing-instances *routing-instance-name* vxlan]**, or **[edit vlans *vlan-name* vxlan]** hierarchy has no effect.

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

Related Documentation
- *Configuring OVSDB-Managed VXLANs*

port (OVSDB)

Syntax port *number*;

Hierarchy Level [edit protocols ovsdb controller protocol]

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

Description Specify the software-defined networking (SDN) controller port to which a Juniper Networks device that supports the Open vSwitch Database (OVSDB) management protocol connects.

Options ***number***—Number of the SDN controller port.

Range: 1024 through 65,535

Default: 6632

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

Related Documentation

- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

protocol (OVSDB)

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 software-defined networking (SDN) 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, see "Creating and Installing an SSL Key and Certificate on a Juniper Networks Device for Connection with SDN Controllers" on page 27.

Options ***protocol***—Establish a secure connection to the SDN controller, using SSL or 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

- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

traceoptions (OVSDB)

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 switches only) Events for the automatic configuration of Virtual Extensible LANs (VXLANs).

ovs-client—OVSDB 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, *size*k to specify KB, *size*m to specify MB, or *size*g 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

- *Example: Setting Up Inter-VXLAN Routing and OVSDB Connections in a Data Center*

CHAPTER 4

VXLAN Configuration Statements

vtep-source-interface

Syntax vtep-source-interface *logical-interface*;

Hierarchy Level [edit switch-options,

Release Information Statement introduced in Junos OS Release 14.1X53-D10.

Description Configure a source interface for a VXLAN tunnel. You must provide the name of a logical interface configured on the loopback interface.

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

Related Documentation

- Understanding VXLANs on page 3
- *Configuring VXLANs on a QFX5100 Switch*
- *Examples: Manually Configuring VXLANs on QFX Series Switches*

CHAPTER 5

OVSDB Operational Commands

- clear ovsdb commit failures
- show ovsdb commit failures
- show ovsdb controller
- show ovsdb interface
- show ovsdb logical-switch
- show ovsdb mac
- show ovsdb statistics interface
- show ovsdb virtual-tunnel-end-point

clear ovsdb commit failures

Syntax clear ovsdb commit failures
<transaction-id>

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

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

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

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

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

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

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

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

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

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

Required Privilege Level clear

Related Documentation
- show ovsdb commit failures on page 72

List of Sample Output
clear ovsdb commit failures on page 71
clear ovsdb commit failures (Specific Transaction) on page 71

Sample Output

clear ovsdb commit failures

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

clear ovsdb commit failures (Specific Transaction)

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

show ovsdb commit failures

Syntax show ovsdb commit failures
<*transaction-id*>

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

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

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

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

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

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

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

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

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

Required Privilege Level admin

Related Documentation
- Understanding Automatically Configured VXLANs in an OVSDB Environment on page 14
- traceoptions (OVSDB) on page 65

List of Sample Output
show ovsdb commit failures on page 73
show ovsdb commit failure (Specific Transaction) on page 73

Output Fields Table 16 on page 73 lists the output fields for the **show ovsdb commit failures** command. Output fields are listed in the approximate order in which they appear.

Table 16: show ovsdb commit failures Output Fields

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

Sample Output

show ovsdb commit failures

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

show ovsdb commit failure (Specific Transaction)

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

show ovsdb controller

Syntax
show ovsdb 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 software-defined networking (SDN) controllers to which the Juniper Networks device is connected.

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

address *ip-address*—Display information about the SDN controller at the specified IP address.

Required Privilege Level
admin

Related Documentation
- Understanding How to Set Up Open vSwitch Database Connections Between Juniper Networks Devices and SDN Controllers on page 11

List of Sample Output
show ovsdb controller on page 75
show ovsdb controller address on page 75

Output Fields
Table 17 on page 74 lists the output fields for the **show ovsdb controller** command. Output fields are listed in the approximate order in which they appear.

Table 17: show ovsdb controller Output Fields

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

Sample Output

show ovsdb controller

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

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

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

show ovsdb controller address

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

show ovsdb interface

Syntax show ovsdb 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.

Description Display information about Open vSwitch Database (OVSDB)-managed interfaces configured by using the **interfaces *interface-name*** statement in the [**edit protocols ovsdb**] hierarchy.

Options **none**—Display information about all OVSDB-managed interfaces.

interface-name—Display information about the specified OVSDB-managed interface.

Required Privilege Level admin

Related Documentation
- *Configuring OVSDB-Managed VXLANs*
- show ovsdb statistics interface on page 85

List of Sample Output
show ovsdb interface on page 76
show ovsdb (Specific Interface) on page 77

Output Fields Table 18 on page 76 lists the output fields for the **show ovsdb interface** command. Output fields are listed in the approximate order in which they appear.

Table 18: show ovsdb interface Output Fields

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

Sample Output

show ovsdb interface

```
user@host> show ovsdb 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 ovsdb (Specific Interface)

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

show ovsdb logical-switch

Syntax show ovsdb logical-switch
<*logical-switch-name*>

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

Description

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

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

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

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

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

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

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

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

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

Options

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

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

Required Privilege Level admin

Related Documentation
- Troubleshooting a Nonoperational Logical Switch and Corresponding Junos OS OVSDB-Managed VXLAN on page 51

List of Sample Output
show ovsdb logical-switch on page 79
show ovsdb logical-switch (Specific Logical Switch) on page 79

Output Fields Table 19 on page 79 lists the output fields for the **show ovsdb logical-switch** command. Output fields are listed in the approximate order in which they appear.

Table 19: show ovsdb logical-switch Output Fields

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

Sample Output

show ovsdb logical-switch

```
user@host> show ovsdb 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 ovsdb logical-switch (Specific Logical Switch)

```
user@host> show ovsdb 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 ovsdb mac

Syntax

show ovsdb 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 are known as local addresses, while the addresses learned from other software or hardware VTEPs are known as remote addresses.

Options

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

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

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

count—(All Juniper Networks devices that support 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 in the logical switch table of the OVSDB schema for physical devices. You can use this option with one or more of the other options to display a more specific count of MAC addresses. For example, to display the number of local and remote unicast MAC addresses, you can issue the **show ovsdb mac count local remote unicast** command.

local—Display all local MAC addresses.

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

multicast—Display all multicast MAC addresses.

remote—Display all remote MAC addresses.

unicast—Display all unicast MAC addresses.

Required Privilege Level admin

List of Sample Output

Output Fields Table 20 on page 82 lists the output fields for the **show ovsdb mac** command. Output fields are listed in the approximate order in which they appear.

Table 20: show ovsdb mac Output Fields

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

Sample Output

show ovsdb mac

```
user@host> show ovsdb mac
Logical Switch Name: 24a76aff-7e61-4520-a78d-3eca26ad7510
  Mac                 IP                Encapsulation     Vtep
  Address             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                              Address
  02:00:00:00:11:01      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  02:00:00:00:11:02      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  02:00:00:00:11:03      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  02:00:00:00:11:04      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  02:00:00:00:11:05      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  04:00:00:00:11:05      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  06:00:00:00:11:01      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  06:00:00:00:11:02      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  06:00:00:00:11:03      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  06:00:00:00:11:04      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  06:00:00:00:11:05      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  40:b4:f0:06:6f:f0      0.0.0.0           Vxlan over Ipv4    10.1.1.29
  00:23:9c:5e:a7:f0      0.0.0.0           Vxlan over Ipv4    10.255.18.22
  08:00:00:00:11:01      0.0.0.0           Vxlan over Ipv4    10.255.18.22
  08:00:00:00:11:02      0.0.0.0           Vxlan over Ipv4    10.255.18.22
  08:00:00:00:11:03      0.0.0.0           Vxlan over Ipv4    10.255.18.22
  08:00:00:00:11:04      0.0.0.0           Vxlan over Ipv4    10.255.18.22
  08:00:00:00:11:05      0.0.0.0           Vxlan over Ipv4    10.255.18.22
  ff:ff:ff:ff:ff:ff      0.0.0.0           Vxlan over Ipv4    10.110.110.1
...
```

show ovsdb mac address

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

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

show ovsdb mac logical-switch

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

show ovsdb mac local unicast

```
user@host> show ovsdb mac local unicast
Logical Switch Name: 24a76aff-7e61-4520-a78d-3eca26ad7510
  Mac                 IP                Encapsulation      Vtep
  Address             Address                              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 ovsdb mac (Count of All Local, Remote, Unicast, and Multicast MAC Addresses for All Logical Switches)

```
user@host> show ovsdb mac count
MAC count: 6877
```

show ovsdb statistics interface

Syntax show ovsdb 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.

Description Display statistics for Open vSwitch Database (OVSDB)-managed interfaces configured by using the **interfaces *interface-name*** statement in the [**edit protocols ovsdb**] hierarchy.

When an interface is configured as OVSDB-managed, the collection of statistics for that interface begins, and the statistics displayed at any given time reflects the data collected up to that point.

Options **none**—Display statistics for all configured OVSDB-managed interfaces.

interface-name—Display statistics for the specified interface.

Required Privilege Level admin

Related Documentation
- interfaces on page 59

List of Sample Output
show ovsdb statistics interface on page 85
show ovsdb statistics interface (Specific Interface) on page 86

Output Fields Table 21 on page 85 lists the output fields for the **show ovsdb statistics interface** command. Output fields are listed in the approximate order in which they appear.

Table 21: show ovsdb statistics interface Output Fields

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

Sample Output

show ovsdb statistics interface

```
user@host> show ovsdb statistics interface
Interface Name: ge-7/0/9.0
Num of rx pkts: 945                    Num of tx pkts: 113280890
Num of rx bytes: 56700                  Num of tx bytes: 57531319540
Interface Name: ge-7/0/10.0
```

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

show ovsdb statistics interface (Specific Interface)

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

show ovsdb virtual-tunnel-end-point

Syntax
show ovsdb 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.

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)
- Software VTEPs
- Service nodes
- Top-of-rack service nodes (TSNs)

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

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

encapsulation *encapsulation-type*—Display information about all entities with the specified encapsulation type.

Required Privilege Level
admin

List of Sample Output
show ovsdb virtual-tunnel-end-point on page 88
show ovsdb virtual-tunnel-end-point address (Specific Address) on page 88
show ovsdb virtual-tunnel-end-point encapsulation (Specific Encapsulation) on page 88
show ovsdb virtual-tunnel-end-point address (Specific Address) encapsulation (Specific Encapsulation) on page 88

Output Fields
Table 22 on page 87 lists the output fields for the **show ovsdb virtual-tunnel-end-point** command. Output fields are listed in the approximate order in which they appear.

Table 22: show ovsdb 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 ovsdb virtual-tunnel-end-point

```
user@host> show ovsdb 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 ovsdb virtual-tunnel-end-point address (Specific Address)

```
user@host> show ovsdb 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 ovsdb virtual-tunnel-end-point encapsulation (Specific Encapsulation)

```
user@host> show ovsdb 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 ovsdb virtual-tunnel-end-point address (Specific Address) encapsulation (Specific Encapsulation)

```
user@host> show ovsdb 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
```

CHAPTER 6

VXLAN Monitoring Commands

Monitor a Remote VTEP Interface

Purpose Monitor traffic details for a remote VTEP interface.

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

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

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

Related Documentation

- Understanding VXLANs on page 3
- *Configuring VXLANs on a QFX5100 Switch*
- *Examples: Manually Configuring VXLANs on QFX Series Switches*

ping overlay

Syntax ping overlay [count *value*] [hash-parameters source-mac *src-mac*] [source-host *ip-address*] [ttl *value*] tunnel-dst *ip-address* tunnel-src *ip-address* tunnel-type vxlan vni *id*

Release Information Command introduced in Junos OS Release 14.1X53-D30 for QFX Series switches.

Description On a QFX5100 switch, force the **ping** packets to follow the same path as data packets through a VXLAN tunnel. In other words, make the underlay packets (**ping** packets) take the same route as the overlay packets (data traffic).

Options

count *value*—Number of pings to send (1-65535).

hash-parameters source-mac *src-mac*—Not supported.

source-host *ip-address*—IP address of the host (virtual machine or bare metal server) that is the source of the tunnel. The default address is 127.0.0.1.

ttl *value*—TTL value to use in the ping packets (1-255).

tunnel-dst *ip-address*—IP address of the remote virtual tunnel endpoint (VTEP).

tunnel-src *ip-address*—IP address of the source VTEP.

tunnel-type vxlan—Value must be **vxlan**.

vni *id* —Value of the VXLAN network identifier that identifies the overlay segment (0-16777215).

Additional Information

Required Privilege Level

Related Documentation

- Understanding VXLANs on page 3
- *Examples: Manually Configuring VXLANs on QFX Series Switches*
- VXLAN Constraints on QFX5100 Switches on page 8
- **traceroute overlay on page 101**

show bridge mac-table

Syntax

show bridge mac-table
<brief | count | detail | extensive>
<bridge-domain (all | *bridge-domain-name*)>
<global-count>
<interface *interface-name*>
<*mac-address*>
<vlan-id (all-vlan | *vlan-id*)>

Release Information

Command introduced in Junos OS Release 8.4.

Description

(MX Series routers only) Display Layer 2 MAC address information.

Options

none—Display all learned Layer 2 MAC address information.

brief | count | detail | extensive—(Optional) Display the specified level of output.

bridge-domain (all | *bridge-domain-name*)—(Optional) Display learned Layer 2 MAC addresses for all bridging domains or for the specified bridging domain.

global-count—(Optional) Display the total number of learned Layer 2 MAC addresses on the system.

instance *instance-name*—(Optional) Display learned Layer 2 MAC addresses for the specified routing instance.

interface *interface-name*—(Optional) Display learned Layer 2 MAC addresses for the specified interface.

mac-address—(Optional) Display the specified learned Layer 2 MAC address information.

vlan-id (all-vlan | *vlan-id*)—(Optional) Display learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.

Additional Information

When Layer 2 protocol tunneling is enabled, the tunneling MAC address 01:00:0c:cd:cd:d0 is installed in the MAC table. When the Cisco Discovery Protocol (CDP), Spanning Tree Protocol (STP), or VLAN Trunk Protocol (VTP) is configured for Layer 2 protocol tunneling on an interface, the corresponding protocol MAC address is installed in the MAC table.

Required Privilege Level

view

List of Sample Output

show bridge mac-table on page 93
show bridge mac-table (with PBB-EVPN enabled) on page 94
show bridge mac-table (with VXLAN enabled) on page 94
show bridge mac-table count on page 94
show bridge mac-table detail on page 95

Output Fields

Table 23 on page 93 describes the output fields for the **show bridge mac-table** command. Output fields are listed in the approximate order in which they appear.

Table 23: show bridge mac-table Output fields

Field Name	Field Description
Routing instance	Name of the routing instance.
Bridging domain	Name of the bridging domain.
MAC address	MAC address or addresses learned on a logical interface.
MAC flags	Status of MAC address learning properties for each interface: • **S**—Static MAC address is configured. • **D**—Dynamic MAC address is configured. • **L**—Locally learned MAC address is configured. • **C**—Control MAC address is configured. • **SE**—MAC accounting is enabled. • **NM**—Non-configured MAC. • **R**—Remote PE MAC address is configured.
Logical interface	Name of the logical interface.
MAC count	Number of MAC addresses learned on the specific routing instance or interface.
Learning interface	Name of the logical interface on which the MAC address was learned.
Learning VLAN	VLAN ID of the routing instance or bridge domain in which the MAC address was learned.
VXLAN ID/VXLAN	VXLAN Network Identifier (VNI)
Layer 2 flags	Debugging flags signifying that the MAC address is present in various lists.
Epoch	Spanning Tree Protocol epoch number identifying when the MAC address was learned. Used for debugging.
Sequence number	Sequence number assigned to this MAC address. Used for debugging.
Learning mask	Mask of the Packet Forwarding Engines where this MAC address was learned. Used for debugging.
IPC generation	Creation time of the logical interface when this MAC address was learned. Used for debugging.

Sample Output

show bridge mac-table

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

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

show bridge mac-table (with PBB-EVPN enabled)

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

Routing instance : pbbn10
 Bridging domain : bda, VLAN : 100
   MAC                 MAC      Logical          NH     RTR
   address             flags    interface        Index  ID
   00:26:88:5f:67:b0   DC                        1048581 1048581
   00:51:51:51:51:51   DC                        1048581 1048581
   00:52:52:52:52:52   DC                        1048576 1048576
   01:1e:83:00:03:e8   DC                        1048580 0
   01:1e:83:00:07:d0   DC                        1048579 0
   a8:d0:e5:5b:01:c8   DC                        1048576 1048576
```

show bridge mac-table (with VXLAN enabled)

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

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


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

show bridge mac-table count

```
user@host> show bridge mac-table count
```

```
2 MAC address learned in routing instance vs1 bridge domain vlan100

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

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

0 MAC address learned in routing instance vs1 bridge domain vlan200

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

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

show bridge mac-table detail

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

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

show vpls mac-table

Syntax

show vpls mac-table
<**brief | detail | extensive | summary**>
<**bridge-domain** *bridge-domain-name*>
<**instance** *instance-name*>
<**interface** *interface-name*>
<**logical-system (all |** *logical-system-name***)**>
<*mac-address*>
<**vlan-id** *vlan-id-number*>

Release Information

Command introduced in Junos OS Release 8.5.

Description

(MX960 routers only) Display learned VPLS MAC address information.

Options

none—Display all learned VPLS MAC address information.

brief | detail | extensive | summary—(Optional) Display the specified level of output.

bridge-domain ***bridge-domain-name***—(Optional) Display learned VPLS MAC addresses for the specified bridge domain.

instance ***instance-name***—(Optional) Display learned VPLS MAC addresses for the specified instance.

interface ***interface-name***—(Optional) Display learned VPLS MAC addresses for the specified instance.

logical-system (all | ***logical-system-name*****)**—(Optional) Display learned VPLS MAC addresses for all logical systems or for the specified logical system.

mac-address—(Optional) Display the specified learned VPLS MAC address information..

vlan-id ***vlan-id-number***—(Optional) Display learned VPLS MAC addresses for the specified VLAN.

Required Privilege Level

view

List of Sample Output

show vpls mac-table on page 97
show vpls mac-table (with VXLAN enabled) on page 98
show vpls mac-table count on page 98
show vpls mac-table detail on page 99
show vpls mac-table extensive on page 99

Output Fields

Table 24 on page 96 describes the output fields for the **show bridge mac-table** command. Output fields are listed in the approximate order in which they appear.

Table 24: show vpls mac-table Output fields

Field Name	Field Description
Routing instance	Name of the routing instance.

Table 24: show vpls mac-table Output fields *(continued)*

Field Name	Field Description
Bridging domain	Name of the bridging domain.
MAC address	MAC address or addresses learned on a logical interface.
MAC flags	Status of MAC address learning properties for each interface: • **S**—Static MAC address configured. • **D**—Dynamic MAC address learned. • **SE**—MAC accounting is enabled. • **NM**—Nonconfigured MAC.
Logical interface	Name of the logical interface.
MAC count	Number of MAC addresses learned on a specific routing instance or interface.
Learning interface	Logical interface or logical Label Switched Interface (LSI) the address is learned on.
Learn VLAN ID/VLAN	VLAN ID of the routing instance or bridge domain in which the MAC address was learned.
VXLAN ID/VXLAN	VXLAN Network Identifier (VNI)
Layer 2 flags	Debugging flags signifying that the MAC address is present in various lists.
Epoch	Spanning Tree Protocol epoch number identifying when the MAC address was learned. Used for debugging.
Sequence number	Sequence number assigned to this MAC address. Used for debugging.
Learning mask	Mask of Packet Forwarding Engines where this MAC address was learned. Used for debugging.
IPC generation	Creation time of the logical interface when this MAC address was learned. Used for debugging.

Sample Output

show vpls mac-table

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

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

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

Routing instance : vpls_red
```

```
VLAN : 401
   MAC                MAC      Logical
   address            flags    interface
   00:00:aa:12:12:12  D        lsi.1051138
   00:05:85:74:9f:f0  D        lsi.1051138
```

show vpls mac-table (with VXLAN enabled)

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

Routing instance : vpls_4site:1000
 Bridging domain : __vpls_4site:1000__, VLAN : 4094,4093
  VXLAN: Id : 300, Multicast group: 226.1.1.3
   MAC                MAC      Logical
   address            flags    interface
   00:01:01:00:01:f4  D,SE     ge-4/2/0.1000
   00:02:01:33:01:f4  D,SE     lsi.1052004
   00:03:00:32:01:f4  D,SE     lsi.1048840
   00:04:00:14:01:f4  D,SE     lsi.1052005
   00:02:01:33:02:f7  D,SE     vtep.1052010
   00:04:00:14:02:f7  D,SE     vtep.1052011
```

show vpls mac-table count

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

  MAC address count per interface within routing instance:
    Logical interface        MAC count
    lc-0/0/0.32769                   0
    lc-0/1/0.32769                   0
    lc-0/2/0.32769                   0
    lc-2/0/0.32769                   0
    lc-0/3/0.32769                   0
    lc-2/1/0.32769                   0
    lc-9/0/0.32769                   0
    lc-11/0/0.32769                  0
    lc-2/2/0.32769                   0
    lc-9/1/0.32769                   0
    lc-11/1/0.32769                  0
    lc-2/3/0.32769                   0
    lc-9/2/0.32769                   0
    lc-11/2/0.32769                  0
    lc-11/3/0.32769                  0
    lc-9/3/0.32769                   0

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

1 MAC address learned in routing instance vpls_ldp1

  MAC address count per interface within routing instance:
    Logical interface        MAC count
    lsi.1051137                      0
    ge-0/2/5.400                     1

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

```
                0                   1

1 MAC address learned in routing instance vpls_red

  MAC address count per interface within routing instance:
    Logical interface        MAC count
    ge-0/2/5.300                     1

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID            MAC count
                0                    1
```

show vpls mac-table detail

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

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

show vpls mac-table extensive

```
user@host> show vpls mac-table extensive
MAC address: 00:00:aa:12:12:12
  Routing instance: vpls_ldp1
   Learning interface: lsi.1051137
   Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
   Epoch: 0                          Sequence number: 1
   Learning mask: 0x1                IPC generation: 0

MAC address: 00:05:85:74:9f:f0
  Routing instance: vpls_ldp1
   Learning interface: lsi.1051137
   Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
   Epoch: 0                          Sequence number: 1
   Learning mask: 0x1                IPC generation: 0

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

MAC address: 00:00:aa:12:12:12
  Routing instance: vpls_red
   Learning interface: lsi.1051138
   Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
   Epoch: 0                          Sequence number: 0
   Learning mask: 0x1                IPC generation: 0

MAC address: 00:05:85:74:9f:f0
```

```
Routing instance: vpls_red
 Learning interface: lsi.1051138
 Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
 Epoch: 0                          Sequence number: 0
 Learning mask: 0x1                IPC generation: 0
```

traceroute overlay

Syntax traceroute overlay [count *value*] [hash-parameters source-mac *src-mac*] [source-host *ip-address*] [ttl *value*] tunnel-dst *ip-address* tunnel-src *ip-address* tunnel-type vxlan vni *id*

Release Information Command introduced in Junos OS Release 14.1X53-D30 for QFX Series switches.

Description On a QFX5100 switch, force the **traceroute** packets to follow the same path as data packets through a VXLAN tunnel. In other words, make the underlay packets (**traceroute** packets) take the same route as the overlay packets (data traffic).

Options

count *value*—Number of packets to send (1-65535).

hash-parameters source-mac *src-mac*—Not supported.

source-host *ip-address*—IP address of the host (virtual machine or bare metal server) that is the source of the tunnel. The default address is 127.0.0.1.

ttl *value*—TTL value to use in the traceroute packets (1-255).

tunnel-dst *ip-address*—IP address of the remote virtual tunnel endpoint (VTEP).

tunnel-src *ip-address*—IP address of the source VTEP.

tunnel-type vxlan—Value must be **vxlan**.

vni *id* —Value of the VXLAN network identifier that identifies the overlay segment (0-16777215).

Additional Information

Required Privilege Level

Related Documentation

- Understanding VXLANs on page 3
- *Examples: Manually Configuring VXLANs on QFX Series Switches*
- VXLAN Constraints on QFX5100 Switches on page 8
- **ping overlay on page 91**

Verifying VXLAN Reachability

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

Action

```
user@switch> show ethernet-switching vxlan-tunnel-end-point remote
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx
<default>                 0   10.1.1.2     lo0.0    0
RVTEP-IP         IFL-Idx   NH-Id
10.1.1.2         559       1728
VNID             MC-Group-IP
100              232.1.1.1
```

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

Related Documentation

- Understanding VXLANs on page 3
- *Configuring VXLANs on a QFX5100 Switch*
- *Examples: Manually Configuring VXLANs on QFX Series Switches*

Verifying That a Local VXLAN VTEP is Configured Correctly

Purpose Verify that a local VTEP is correct..

Action

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

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

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

Related Documentation

- Understanding VXLANs on page 3
- *Configuring VXLANs on a QFX5100 Switch*
- *Examples: Manually Configuring VXLANs on QFX Series Switches*

Verifying MAC Learning from a Remote VTEP

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

Action user@switch> show ethernet-switching table

```
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)
Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN1                00:00:00:ff:ff:ff   D             -   vtep.12345
    VLAN1                00:10:94:00:00:02   D         -       xe-0/0/0.0
```

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

Related Documentation

- Understanding VXLANs on page 3
- *Configuring VXLANs on a QFX5100 Switch*
- *Examples: Manually Configuring VXLANs on QFX Series Switches*

