Juniper Networks | Engineering Simplicity

How to Configure the NFX350

Published
2020-09-24

Junos

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

How to Configure the NFX350

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 https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation

IN THIS SECTION

Use this guide to perform initial provisioning, configure Junos OS features, chain multiple virtualized network functions, monitor, and manage the NFX350 Series devices.

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 https://www.juniper.net/documentation/.

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 https://www.juniper.net/books.

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 CLI Explorer.

Documentation Conventions

Table 1 on page xii 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 xii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
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*

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

Convention	Description	Examples
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		

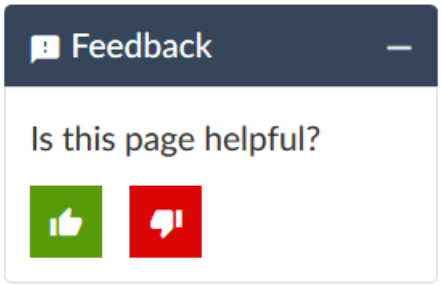
Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
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**.
> (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 so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide 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 Juniper Care or Partner Support Services 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 https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://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: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- 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 https://support.juniper.net/support/requesting-support/.

1

CHAPTER

Overview

NFX350 Overview

IN THIS SECTION

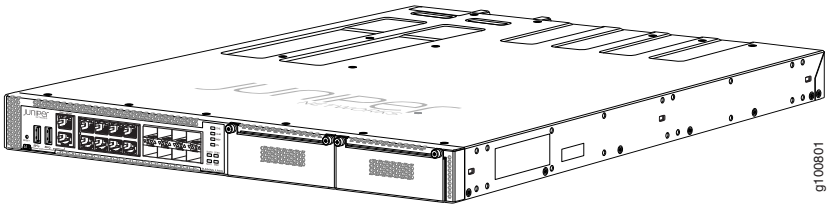
The Juniper Networks NFX350 Network Services Platform is a secure, automated, software-driven customer premises equipment (CPE) platform that delivers virtualized network and security services on demand. The NFX350 is part of the Juniper Cloud CPE solution, which leverages Network Functions Virtualization (NFV).

The NFX350 platform completes the uCPE portfolio to provide end-to-end platforms for medium, large, and extra-large deployments. In addition to IPsec and SD-WAN functionality, the NFX350 provides features such as LAN or WAN isolation, software and hardware resiliency, redundant power supply, Baseboard Management Controller, and serial over LAN.

The NFX350 has the Intel Skylake-D processor which provides increased throughput and cache. Integrated QAT helps accelerate applications that perform cryptographic operations such as IPsec.

Figure 1 on page 17 shows the NFX350 device.

Figure 1: NFX350 Device



Some typical deployment scenarios where you can use the NFX350 are:

- MSP/SP large/extra-large deployments requiring platform resiliency

- IOT gateway
- Resource-intensive deployments

Software Architecture

The architecture is designed to provide a unified control plane that functions as a single management point. Key components in the software include the JCP, JDM, Layer 2 data plane, Layer 3 data plane, and VNFs.

Figure 2 on page 18 and Figure 3 on page 19 illustrate the software architecture of the NFX350 in throughput, hybrid, and compute modes.

Figure 2: NFX350 NextGen Software Architecture (Throughput Mode)

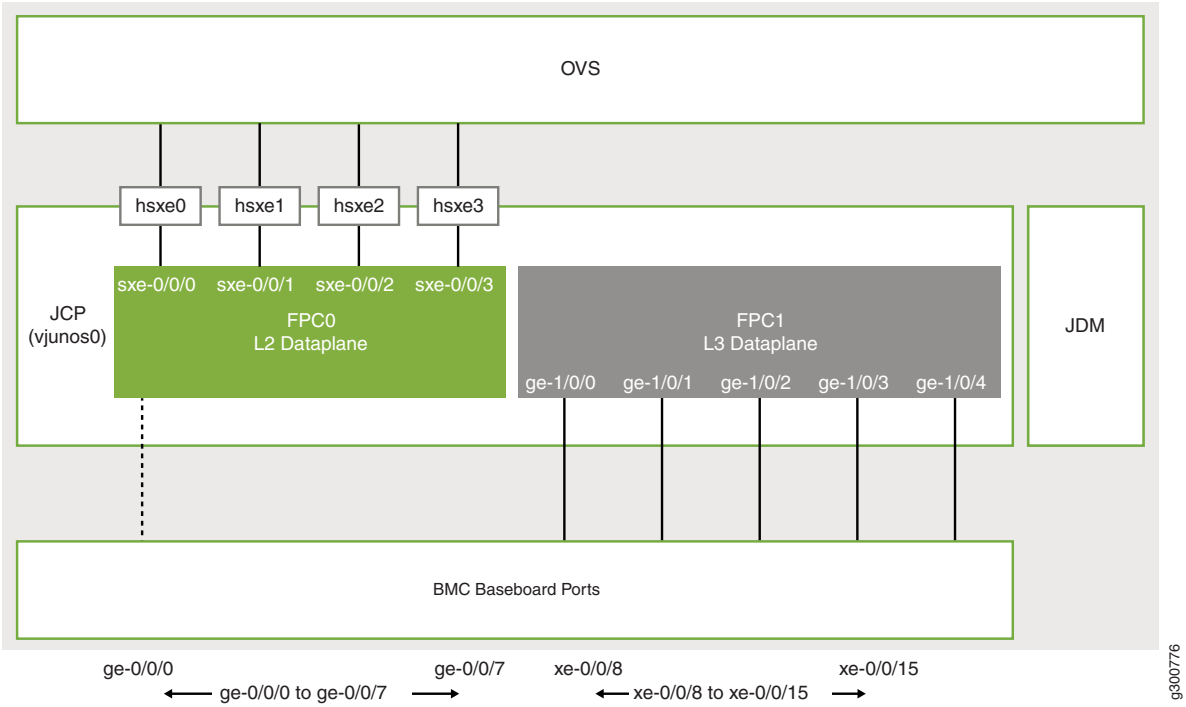
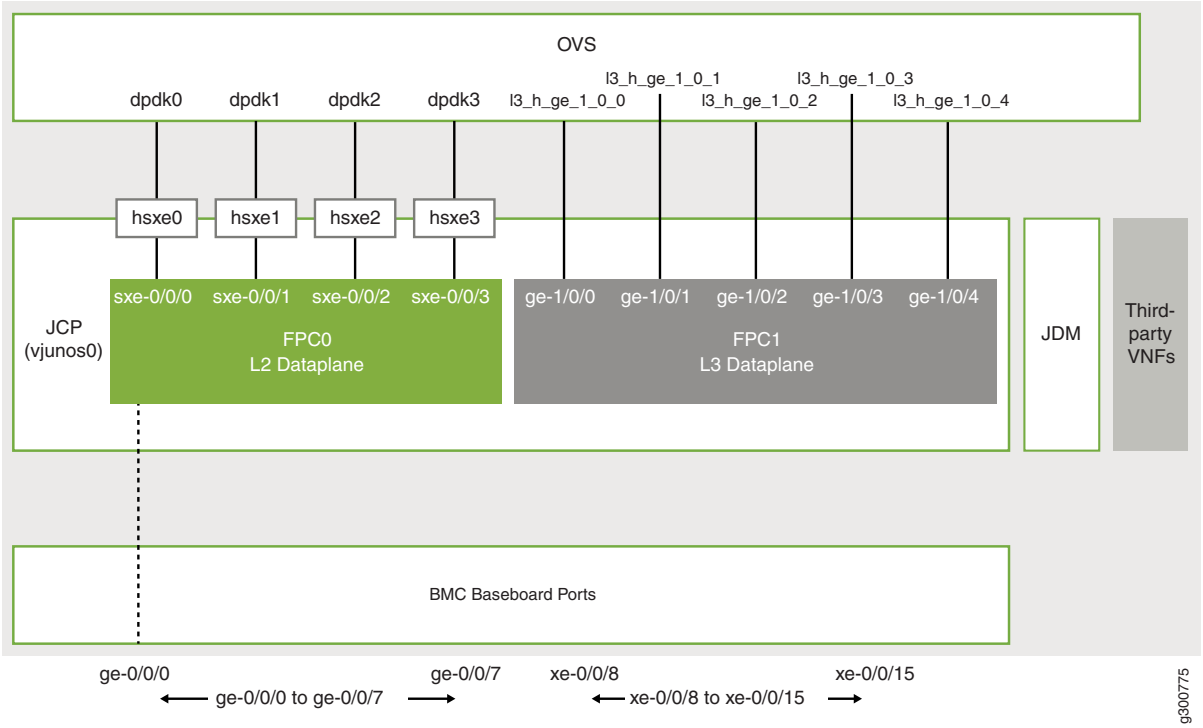


Figure 3: NFX350 NextGen Software Architecture (Hybrid or Compute Mode)



Key components of the system software include:

- Linux—The host OS, which functions as the hypervisor.
- VNF—A VNF is a virtualized implementation of a network device and its functions. Linux functions as the hypervisor, and it creates and runs the VNFs. The VNFs include functions such as firewalls, routers, and WAN accelerators.

  You can connect VNFs together as blocks in a chain to provide networking services. The NFX350 supports up to eight VNFs thereby enabling increased network functions and port density.
- JCP—Junos virtual machine (VM) running on the host OS, Linux. The JCP functions as the single point of management for all the components.

  The JCP supports:

  - Layer 2 to Layer 3 routing services
  - Layer 3 to Layer 4 security services
  - Layer 4 to Layer 7 advanced security services

  In addition, the JCP enables VNF lifecycle management.
- JDM—An application container that manages VNFs and provides infrastructure services. The JDM functions in the background. Users cannot access the JDM directly.

- L2 data plane—Manages Layer 2 traffic. The Layer 2 dataplane forwards the LAN traffic to the Open vSwitch (OVS) bridge, which acts as the NFV backplane. The Layer 2 dataplane is mapped to the virtual FPC0 on the JCP.
- L3 data plane—Provides data path functions for the Layer 3 to Layer 7 services. The Layer 3 data plane is mapped to the virtual FPC1 on the JCP.
- Open vSwitch (OVS) bridge—The OVS bridge is a VLAN-aware system bridge that acts as the NFV backplane to which the VNFs, FPC1, and FPC0 connect. Additionally, you can create custom OVS bridges to isolate connectivity between different VNFs.

  On NFX350, you can configure up to 72 OVS interfaces, which includes the VNF and FPC1 interfaces.

For the list of supported features, see Feature Explorer.

NFX350 Models

Table 3 on page 20 lists the NFX350 device models and its specifications. For more information, see the *NFX350 Hardware Guide*.

Table 3: NFX350 Series Device Models and Specifications

	NFX350-S1	NFX350-S2	NFX350-S3
CPU	8-core Intel Skylake D-2146NT	12-core Intel Skylake D-2166NT	16-core Intel Skylake D-2187NT
RAM	32 GB	64 GB	128 GB
Storage	100 GB SSD	100 GB SSD	100 GB SSD
Form Factor	Rack	Rack	Rack

Table 3: NFX350 Series Device Models and Specifications *(continued)*

	NFX350-S1	NFX350-S2	NFX350-S3
Ports	Eight 1-Gigabit Ethernet RJ-45 ports	Eight 1-Gigabit Ethernet RJ-45 ports	Eight 1-Gigabit Ethernet RJ-45 ports
	Eight 10-Gigabit Ethernet SFP+ ports	Eight 10-Gigabit Ethernet SFP+ ports	Eight 10-Gigabit Ethernet SFP+ ports
	One management/Intelligent Platform Management Interface (IPMI) port	One management/Intelligent Platform Management Interface (IPMI) port	One management/Intelligent Platform Management Interface (IPMI) port
	One console port (RJ-45 and mini-USB)	One console port (RJ-45 and mini-USB)	One console port (RJ-45 and mini-USB)
	Two USB 3.0 port	Two USB 3.0 port	Two USB 3.0 port
LTE support	Yes	Yes	Yes
Expansion module support	Two expansion module slots (one dual slot width NFX-LTE-AA/AE expansion module slot width expansion module)	Two expansion module slots (one dual slot width NFX-LTE-AA/AE expansion module slot width expansion module)	Two expansion module slots (one dual slot width NFX-LTE-AA/AE expansion module slot width expansion module)
Supported expansion modules	• NFX-LTE-AE—Expansion module with an LTE modem supporting the frequency bands in Europe and North America. • NFX-LTE-AA—Expansion module with an LTE modem supporting the frequency bands in Asia, Australia, and New Zealand.	• NFX-LTE-AE—Expansion module with an LTE modem supporting the frequency bands in Europe and North America. • NFX-LTE-AA—Expansion module with an LTE modem supporting the frequency bands in Asia, Australia, and New Zealand.	• NFX-LTE-AE—Expansion module with an LTE modem supporting the frequency bands in Europe and North America. • NFX-LTE-AA—Expansion module with an LTE modem supporting the frequency bands in Asia, Australia, and New Zealand.

Interfaces

The NFX350 device includes the following network interfaces:

- Eight 1-Gigabit Ethernet RJ-45 ports. The ports follow the naming convention, ge-0/0/*n*, where *n* ranges from 0 to 7. These ports are used for LAN connectivity.
- Eight 10-Gigabit uplink ports that support small form-factor pluggable plus (SFP+) transceivers. The ports follow the naming convention xe-0/0/*n*, where the value of *n* is ranges from 8 to 15. These ports are used as WAN uplink ports.
- A dedicated management port labeled **MGMT** (fxp0) functions as the out-of-band management interface. The fxp0 interface is assigned the IP address 192.168.1.1/24.
- Four static interfaces, sxe-0/0/0, sxe-0/0/1, sxe-0/0/2, and sxe-0/0/3, which connect the Layer 2 data plane (FPC0) to the OVS backplane.

> **NOTE:** By default, all the network ports connect to the Layer 2 data plane.

For the list of supported transceivers for your device, see https://apps.juniper.net/hct/product/#prd=NFX350.

Performance Modes

NFX350 devices provide the following operational modes:

- Throughput mode—Provides maximum resources (CPU and memory) for Junos software. The default mode is throughput mode.

  On throughput mode, you must map SR-IOV VF to Layer 3 Dataplane interfaces on an NFX350 device. Three SR-IOV (VFs) are reserved from each NIC (SXE or HSXE) to support a maximum of 12 Layer 3 Dataplane interfaces. For example:

  ```
  user@host# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface hsxe0
  ```

  > **NOTE:** You cannot create VNFs on Throughput mode.

- Hybrid mode—Provides a balanced distribution of resources between the Junos software and third-party VNFs.

  On hybrid mode, you can map Layer 3 Dataplane interfaces to either SR-IOV or OVS on an NFX350 device. For example:

  Map Layer 3 Dataplane interfaces to either SR-IOV:

  ```
  user@host# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface hsxe0
  ```

Map Layer 3 Dataplane interfaces to either OVS:

```
user@host# set vmhost virtualization-options interfaces ge-1/0/1
```

- Compute mode—Provides minimal resources for Junos software and maximum resources for third-party VNFs.

On hybrid or compute mode, you can create VNFs using the available CPUs on each mode. You can check the CPU availability by using the **show vmhost mode** command. Each VNF can have maximum user interfaces apart from the two management interfaces. You can attach the VNF interfaces to either OVS or SR-IOV interfaces.

> **NOTE:** You cannot attach single VNF interface to both SR-IOV and OVS. However, you can attach different interfaces from the same VNF to SR-IOV and OVS.
>
> Seven SR-IOV (VFs) are reserved from each NIC (SXE or HSXE) to create VNF interfaces, and supports up to a maximum of 28 SR-IOV VNF interfaces per device. You can view the available free VFs by using the **show system visibility network**.

When switching between operational modes, ensure that resource and configuration conflicts do not occur. Before switching to a mode, issue the **show system visibility cpu** and **show vmhost mode** to check the availability of CPUs. For example, if you move from compute mode that supports VNFs to throughput mode that does not support VNFs, conflicts occur. For example:

```
user@host# run request vmhost mode throughput
```

```
error: Mode cannot be changed; Reason: No CPUs are available for VNFs in the desired
 mode, but there is atleast one VNF currently configured
```

If the Layer 3 dataplane is not mapped to SR-IOV, then switching from hybrid or compute mode to throughput mode results in an error.

Benefits and Uses

The NFX350 provides the following benefits:

- Highly scalable architecture that supports multiple Juniper VNFs and third-party VNFs on a single device. The modular software architecture provides high performance and scalability for routing, switching, and security enhanced by carrier-class reliability.
- Integrated security, routing, and switching functionality in a single control plane simplifies management and deployment.
- A variety of flexible deployments. A distributed services deployment model ensures high availability, performance, and compliance. The device provides an open framework that supports industry standards, protocols, and seamless API integration.
- Wireless WAN support through the LTE module provides more flexibility in deployments.
- Secure boot feature safeguards device credentials, automatically authenticates system integrity, verifies system configuration, and enhances overall platform security.
- Automated configuration eliminates complex device setup and delivers a plug-and-play experience.
- Increased storage capacity through two external hard disks.

Junos OS Releases Supported on NFX Series Hardware

The Table 4 on page 24 provides details of Junos OS software releases supported on the NFX Series devices.

NOTE: Linux bridge mode is supported on NFX250 devices only up to Junos OS Release 18.4.

Table 4: Supported Junos OS Releases on NFX Series Devices

NFX Series Platform	Supported Junos OS Release	Software Package	Software Downloads Page
NFX150	18.1R1 or later	nfx-3 jinstall-host-nfx-3-x86-64-<*release-number*>- secure-signed.tgz install-media-host-usb-nfx-3-x86-64-<*release-number*>- secure.img	NFX150 Software Download Page

Table 4: Supported Junos OS Releases on NFX Series Devices *(continued)*

NFX Series Platform	Supported Junos OS Release	Software Package	Software Downloads Page
NFX250	15.1X53-D45, 15.1X53-D47, 15.1X53-D470, and 15.1X53-D471	nfx-2 jinstall-host-nfx-2-flex-x86-64-<*release-number*>-secure-signed.tgz install-media-host-usb-nfx-2-flex-x86-64-<*release-number*>-secure.img	NFX250 Software Download Page
	17.2R1 through 19.1R1		
	19.1 R1 or later	nfx-3 jinstall-host-nfx-3-x86-64-<*release-number*>-secure-signed.tgz install-media-host-usb-nfx-3-x86-64-<*release-number*>-secure.img	NFX250 Software Download Page
NFX350	19.4 R1 or later	nfx-3 jinstall-host-nfx-3-x86-64-<*release-number*>-secure-signed.tgz install-media-host-usb-nfx-3-x86-64-<*release-number*>-secure.img	NFX350 Software Download Page

SEE ALSO

NFX250 Overview

Baseboard Management Controller Overview

IN THIS SECTION

A Baseboard Management Controller (BMC) is a specialized micro controller, used for remotely managing and recovering NFX350 device.

> **NOTE:** You cannot access BMC through the Management port or console.

Managing BMC

IN THIS SECTION

Using Junos CLI, you can upgrade BMC firmware, check the BMC firmware version, and perform device power cycle.

Perform Power Cycle of the NFX350 Device

You can perform power cycle of the mainboard CPU or device using Junos CLI.

> **NOTE:** The **power-cycle** command performs only the power cycle of mainboard CPU. You cannot perform the power cycle of BMC and CPLD by using this command.

To perform a power cycle:

```
user@host> request vmhost power-cycle

Power cycle the vmhost ? [yes,no] (yes)
```

BMC Firmware Upgrade

You can upgrade BMC using Junos CLI.

> **NOTE:** For BMC upgrade, the image version must be greater than the currently running BMC firmware version.

- To upgrade BMC after copying the firmware to the device file system:

  > **NOTE:** The NFX350 device remains operational during the BMC upgrade process.

  `user@host>` **request system firmware upgrade jfirmware bmc file *BMC-firmware-path***

  For example,

  `user@host>` **request system firmware upgrade jfirmware bmc file ****

  ```
  Validated nfx-3-jfirmware-19.4R1.5.tgz
  BMC firmware upgrade initiated
  Check progress using "request system firmware upgrade jfirmware
  ..<type>.. progress“
  ```

  After the BMC firmware upgrade is completed, BMC reboots automatically.

- To check the status of BMC firmware upgrade:

  `user@host>` **request system firmware upgrade jfirmware bmc progress**

  ```
  BMC upgrade in progress
  ```

After the firmware is successfully loaded, **BMC upgrade successful** message is displayed.

View BMC Firmware Version

To view the BMC firmware version:

`user@host>` **show system inventory firmware | match BMC**

```
BMC Version: 00.06_00.02
```

NFX Product Compatibility

IN THIS SECTION

Hardware Compatibility

To obtain information about the components that are supported on your devices, and special compatibility guidelines with the release, see the Hardware Guide and the Interface Module Reference for the product.

To determine the features supported on NFX Series devices in this release, use the Juniper Networks Feature Explorer, a Web-based application that helps you to explore and compare Junos OS feature information to find the right software release and hardware platform for your network. Find Feature Explorer at: https://pathfinder.juniper.net/feature-explorer/.

Hardware Compatibility Tool

For a hardware compatibility matrix for optical interfaces and transceivers supported across all platforms, see the Hardware Compatibility Tool.

Software Version Compatibility

This section lists the vSRX and Cloud CPE Solution software releases that are compatible with the Junos OS releases on the NFX Series devices.

NOTE:

- Starting in Junos OS Release 18.1R1, NFX Series devices support the same version of platform software and vSRX. For example, see Table 5 on page 29.
- The Linux Bridge mode is supported only up to Junos OS Release 18.4 on NFX250 devices.

NFX250 Software Version Compatibility

This section lists the vSRX and CloudCPE Solution software releases that are compatible with the Junos OS releases on the NFX250 devices:

Table 5: Software Compatibility Details with vSRX and Cloud CPE Solution

NFX250 Junos OS Release	vSRX	Cloud CPE Solution
15.1X53-D40.3	15.1X49-D40.6	Cloud CPE Solution 2.0
15.1X53-D41.6	15.1X49-D40.6	Cloud CPE Solution 2.1
15.1X53-D102.2	15.1X49-D61	Cloud CPE Solution 3.0
15.1X53-D47.4	15.1X49-D100.6	Cloud CPE Solution 3.0.1
15.1X53-D490	15.1X49-D143	Cloud CPE Solution 4.0
15.1X53-D495	15.1X49-D160	Cloud CPE Solution 4.1
15.1X53-D496	15.1X49-D170	Cloud CPE Solution 4.1
15.1X53-D45.3	15.1X49-D61	Not applicable
17.2R1	15.1X49-D78.3	Not applicable
17.3R1	15.1X49-D78.3	Not applicable
17.4R1	15.1X49-D78.3	Not applicable
15.1X53-D471	15.1X49-D143	Not applicable
18.1R1	18.1R1	Not applicable
18.1R2	18.1R2	Not applicable

Table 5: Software Compatibility Details with vSRX and Cloud CPE Solution *(continued)*

NFX250 Junos OS Release	vSRX	Cloud CPE Solution
18.1R3	18.1R3	Not applicable
18.2R1	18.2R1	Not applicable
18.3R1	18.3R1	Not applicable
18.4R1	18.4R1	Not applicable

2

CHAPTER

Initial Configuration

Initial Configuration on NFX350 Devices

IN THIS SECTION

Factory Default Settings

The NFX350 is shipped with the following factory default settings:

Table 6: Security Policies

Source Zone	Destination Zone	Policy Action
trust	trust	permit
trust	untrust	permit

Table 7: Interfaces

Port Label	Interface	Security Zone	DHCP State	IP Address
0/0 to 0/7	ge-0/0/0 to ge-0/0/7	trust	server	192.168.2.1/24
0/8 to 0/15	xe-0/0/8 to xe-0/0/15	untrust	client	ISP assigned
MGMT	fxp0	N/A	N/A	192.168.1.1/24

The device is shipped with the following services enabled in the default security policy: DHCP, HTTP, HTTPS, and SSH.

To provide secure traffic, a basic set of screens are configured on the untrust zone.

Enabling Basic Connectivity

1. Ensure that the device is powered on.

2. Connect to the console port:

   a. Plug one end of the Ethernet cable into the console port on your device.

   b. Connect the other end of the Ethernet cable to the RJ-45 to DB-9 serial port adapter shipped with your device.

   c. Connect the RJ-45 to DB-9 serial port adapter to the serial port on the management device. Use the following values to configure the serial port:

      Bits per second—9600; Parity—None; Data bits—8; Stop bits—1; Flow control—None.

      NOTE: Alternately, you can use the USB cable to connect to the mini-USB console port on the device. To use the mini-USB console port, you must download the USB driver from the following page and install the driver on the management device:

      https://www.juniper.net/support/downloads/junos.html

3. Use any terminal emulation program such as HyperTerminal to connect to the device console. The CLI displays a login prompt.

4. Log in as **root**. If the software completes booting before you connect to the console, you might need to press the Enter key for the prompt to appear.

   ```
   login: root
   ```

5. Start the CLI.

   ```
   root@:~ # cli
   root@>
   ```

6. Enter configuration mode.

   ```
   root@> configure
   [edit]
   root@#
   ```

7. Change the password for the root administration user account.

```
[edit]
root@# set system root-authentication plain-text-password
New password: password
Retype new password: password
```

8. Enable SSH service for the root user.

```
[edit]
root@# set system services ssh root-login allow
```

9. (Optional) Enable Internet connection for the devices connected on LAN by setting the DNS IP.

```
[edit]
root@# set access address-assignment pool junosDHCPPool family inet dhcp-attributes name-server
dns-server-ip
```

10. Commit the configuration.

```
[edit]
root@# commit
```

Establishing the Connection

1. Connect the device to the ISP by connecting one of the WAN ports (0/8 through 0/15) to the ISP. The device is assigned an IP address by the ISP through DHCP.

> **NOTE:** For information about interfaces, see Table 7 on page 32.

2. Connect the laptop to one of the front panel LAN ports (0/0 to 0/7). The laptop is assigned an IP address by the DHCP server running on the device.

3. Open a browser window on your laptop, navigate to https://www.juniper.net, and verify your connectivity.

Zero Touch Provisioning on NFX Series Devices

IN THIS SECTION

Understanding Zero Touch Provisioning

Zero Touch Provisioning (ZTP) allows you to provision and configure an NFX Series device in your network automatically, with minimal manual intervention. ZTP allows you to make configuration changes or software upgrades without logging into the device. NFX Series devices support ZTP with Sky Enterprise, which is a cloud-based network management application. For more information on Sky Enterprise, see Sky Enterprise Documentation.

The initial provisioning process involves the following components:

- NFX Series device—Sends requests to Juniper's Redirect Server.
- Redirect server—Provides authentication and authorization for the devices in a network to access their assigned central servers for the boot images and initial configuration files. The redirect server resides at Juniper Networks.

  Connectivity to the redirect server can be through IPv4 or IPv6 network. Depending on the source address, the redirect server redirects the ZTP to the corresponding Central Server with IPv4 or IPv6 address.

  The NFX Series device is shipped with a factory default configuration. The factory default configuration includes the URL of the redirect server, that is used to connect to the central servers by using a secure encrypted connection.
- Central server—Manages the network and the NFX Series devices located remotely. The central server is located at a central geographical location. Alternately, you can use Contrail Service Orchestration (CSO) along with Sky Enterprise. CSO deploys the network services and Sky Enterprise manages the devices in the network.

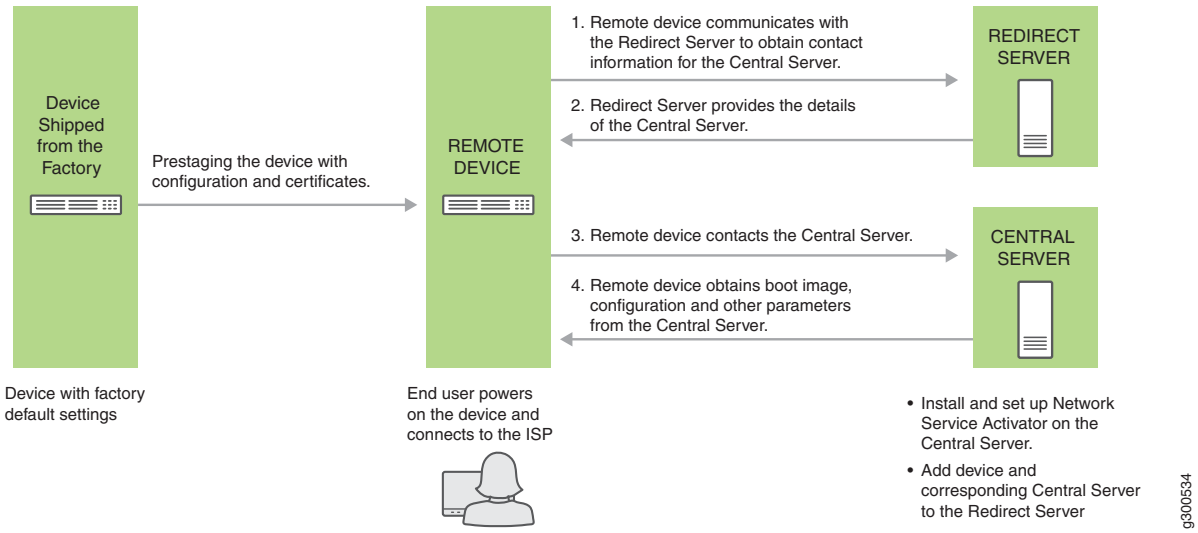
Pre-staging an NFX Series Device

Prestaging is an optional step for the device to by-pass Juniper's Redirect Server and to connect to a customer specific Redirect Server or a Regional Server for authentication and authorization in the network. Prestaging involves copying and applying certificates and customer specific configuration from a specific directory in the device before the device is shipped to the customer site for installation.

The customer specific resources are stored internally. When the device boots up with the factory default configuration, the prestage resources are copied and the configuration is applied on the device.

Figure 4 on page 36 illustrates the workflow of prestaging the NFX Series devices.

Figure 4: Workflow for Prestaging an NFX Series Device



The prestage workflow proceeds as follows:

1. The device is shipped from the factory with the factory default configuration.
2. To prestage the device, the customer specific resources such as certificates and configuration are copied to the device by a user or ISP.

   To add the prestage configuration and certificates, run:

   `user@host>` **request system phone-home pre-stage add configuration *file***
   `user@host>` **request system phone-home pre-stage add certificates *file/files***

3. After the device is prestaged, the device is shipped to the end user.

4. The end user powers on the remote device and connects the device to the ISP by connecting one of the WAN ports (0/12 and 0/13) to the ISP. For more information, see *Initial Configuration on NFX250 NextGen Devices* .
5. The device applies the prestage configuration and uses the certificates to authenticate the customer specific Redirect Server or Regional Server.
6. The Redirect Server or Regional Server sends the corresponding Central Server information to the device.
7. The device sends a provisioning request to the Central Server. The Central Server responds with the boot image and the configuration that is provisioned on the Central Server for that particular device.
8. The device fetches the boot image and configuration file from the Central Server.
9. The device upgrades to the boot image and applies the configuration to start the services and become operational.

To delete the prestage configuration and certificates, run:

```
user@host>request system phone-home pre-stage delete configuration file
```

```
user@host>request system phone-home pre-stage delete certificate all | file
```

```
user@host>request system phone-home pre-stage delete all
```

To verify the prestage configuration and certificates, run:

```
user@host>show system phone-home pre-stage configuration
```

```
user@host>show system phone-home pre-stage certificate
```

```
user@host>show system phone-home pre-stage
```

The prestage resources are not deleted when you upgrade the image by using the **request system software add *image*** command or when you zeroize the device by using the **request system zeroize** command.

The default configuration for phone-home is:

```
user@jdm#  set system phone-home server https://redirect.juniper.net
```

```
user@jdm#  set system phone-home upgrade-image-before-configuration
```

To enable trace operation:

```
user@jdm#  set system phone-home traceoptions file file-name size file-size
```

```
user@jdm#  set system phone-home traceoptions flag [all | config | function | misc | socket | state-machine]
```

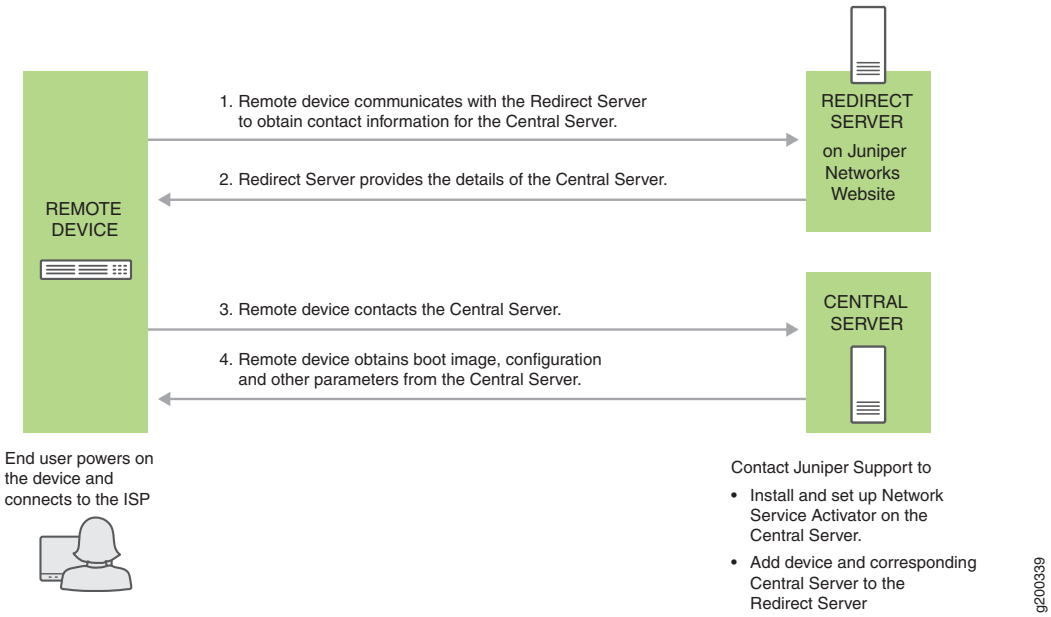
To disable trace operation:

```
user@jdm# set system phone-home traceoptions no-remote-trace
```

Provisioning an NFX Series Device

Figure 5 on page 38 illustrates the workflow of the initial provisioning of NFX Series devices.

Figure 5: Workflow for Initial Provisioning of an NFX Series Device



NOTE: Contact Juniper Support to add the device and the corresponding central server to the redirect server.

The provisioning workflow proceeds as follows:

1. The end user powers on the remote device, and connects the remote device to the ISP through the WAN ports.
2. The remote device transmits its X.509 certificate and fully qualified domain name (FQDN) as a provisioning request to the redirect server.

3. The redirect server searches its data store for the central server that an administrator has specified for the remote device, and confirms that the remote device's request corresponds to the X.509 certificate specified for the server.
4. The redirect server sends contact information for the central server to the remote device.
5. The remote device sends a request to the central server for the URL of the boot image and the location of the initial configuration file. The central server responds with the requested information.
6. The remote device fetches the boot image and configuration file from the central server.
7. The remote device upgrades to the boot image (if the boot image is different from the image running on the NFX Series device), and applies the configuration to start the services and become operational.

Provisioning an NFX Series Device Using Sky Enterprise

Figure 5 on page 38 illustrates the workflow of the initial provisioning of NFX Series devices using Sky Enterprise.

The provisioning workflow proceeds as follows:

1. The end user powers on the remote device, and connects the remote device to the ISP through the WAN ports.
2. The NFX Series device transmits its X.509 certificate and fully qualified domain name (FQDN) as a provisioning request to the Redirect Server.
3. The Redirect Server connects the device to Sky Enterprise.
4. Click the link in the authorization e-mail that you receive from Sky Enterprise. Alternately, you can use the Sky Enterprise application to authorize the device.
5. The NFX Series device registers with Sky Enterprise.
6. The initial configuration of the device begins. The initial configuration process takes about 60 seconds.

3

CHAPTER

Generating YANG Files

YANG files on NFX350 Devices

IN THIS SECTION

Understanding YANG on NFX350 Devices

YANG is a standards-based, extensible data modeling language that is used to model the configuration and operational state data, remote procedure calls (RPCs), and server event notifications of network devices. The NETMOD working group in the IETF originally designed YANG to model network management data and to provide a standard for the content layer of the Network Configuration Protocol (NETCONF) model. However, YANG is protocol independent, and YANG data models can be used independent of the transport or RPC protocol and can be converted into any encoding format supported by the network configuration protocol.

Juniper Networks provides YANG modules that define the Junos OS configuration hierarchy and operational commands and Junos OS YANG extensions. You can generate the modules on the device running Junos OS.

YANG uses a C-like syntax, a hierarchical organization of data, and provides a set of built-in types as well as the capability to define derived types. YANG stresses readability, and it provides modularity and flexibility through the use of modules and submodules and reusable types and node groups.

A YANG module defines a single data model and determines the encoding for that data. A YANG module defines a data model through its data, and the hierarchical organization of and constraints on that data. A module can be a complete, standalone entity, or it can reference definitions in other modules and submodules as well as augment other data models with additional nodes.

A YANG module defines not only the syntax but also the semantics of the data. It explicitly defines relationships between and constraints on the data. This enables you to create syntactically correct configuration data that meets constraint requirements and enables you to validate the data against the model before uploading it and committing it on a device.

YANG uses modules to define configuration and state data, notifications, and RPCs for network operations in a manner similar to how the Structure of Management Information (SMI) uses MIBs to model data for SNMP operations. However, YANG has the benefit of being able to distinguish between operational and configuration data. YANG maintains compatibility with SNMP's SMI version 2 (SMIv2), and you can use

libsmi to translate SMIv2 MIB modules into YANG modules and vice versa. Additionally, when you cannot use a YANG parser, you can translate YANG modules into YANG Independent Notation (YIN), which is an equivalent XML syntax that can be read by XML parsers and XSLT scripts.

For information about YANG, see RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, and related RFCs.

For more information, see YANG Modules Overview, Using Juniper Networks YANG Modules, and show system schema.

Generating YANG Files

You can generate YANG files for JCP on NFX350 devices.

To generate YANG files for JCP:

1. Log in to the NFX device using SSH or console:

   ```
   login: root
   ```

2. Start the CLI:

   ```
   root@:~# cli
   {master:0}
   root>
   ```

3. Create a temporary directory to store the generated YANG files:

   ```
   {master:0}
   root> file make-directory /var/public/yang_files
   {master:0}
   root> file list /var/public/yang_files
    /var/public/yang_files:
   {master:0}
   root>
   ```

4. Generate YANG files for JCP:

   ```
   {master:0}
   root> show system schema module all format yang output-directory /var/public/yang_files
   ```

5. Verify whether YANG files are generated in the specified target directory:

```
{master:0}
root> file list /var/public/yang_files
```

```
/var/public/yang_files:

junos-common-types@2019-01-01.yang
junos-nfx-conf-access-profile@2019-01-01.yang
junos-nfx-conf-access@2019-01-01.yang
junos-nfx-conf-accounting-options@2019-01-01.yang
junos-nfx-conf-applications@2019-01-01.yang
...Output truncated...
```

6. Copy the generated JCP YANG files from the NFX device to the YANG based tools or orchestrators by using the **scp** or **file copy** command.

4

CHAPTER

Configuring Interfaces

Configuring the In-Band Management Interface on NFX350

In in-band management, you configure a network interface as a management interface and connect it to the management device. You can configure any of the ge-1/0/x ports, where x ranges from 0 to 4, as in-band management interfaces.

To configure in-band management:

1. Log in to the JCP CLI and enter configuration mode:

   ```
   root@host% cli
   root@host> configure
   ```

2. Configure VLAN tagging:

   ```
   root@host# set interfaces ge-1/0/x vlan-tagging
   root@host# set interfaces ge-1/0/x unit n vlan-id mgmt-vlan-id
   root@host# set interfaces ge-1/0/x unit n family inet address address/prefix-length
   ```

To configure a LAN port for in-band management:

1. Configure the management VLAN:

   ```
   root@host# set vlans mgmt-vlan vlan-id vlan-id
   ```

2. Add the physical network interface and the service interface as members of the VLAN:

   ```
   root@host# set interfaces ge-0/0/x unit 0 family ethernet-switching vlan members mgmt-vlan
   ```

   Where **x** ranges from 0 to 7.

   ```
   root@host# set interfaces sxe-0/0/x unit 0 family ethernet-switching vlan members mgmt-vlan
   ```

   Where **x** ranges from 0 to 3.

> **NOTE:** If the device is in throughput mode, you must map ge-1/0/*x* to sxe-0/0/*x* by using the **set vmhost virtualization-options interfaces ge-1/0/*x* mapping interface hsx*ex*** command. If the device is in hybrid or compute mode, you must map ge-1/0/*x* to OVS by using the **set vmhost virtualization-options interfaces ge-1/0/*x* mapping interface hsx*ex*** command.

ADSL2 and ADSL2+ Interfaces on NFX350 Devices

IN THIS SECTION

ADSL Interface Overview

Asymmetric digital subscriber line (ADSL) technology is part of the xDSL family of modem technologies that use existing twisted-pair telephone lines to transport high-bandwidth data. ADSL lines connect service provider networks and customer sites over the "last mile" of the network—the loop between the service provider and the customer site.

ADSL transmission is asymmetric because the downstream bandwidth is typically greater than the upstream bandwidth. The typical bandwidths of ADSL2 and ADSL2+ circuits are defined in Table 8 on page 46.

Table 8: Standard Bandwidths of DSL Operating Modes

Operating Modes	Upstream	Downstream
ADSL2	1—1.5 Mbps	12—14 Mbps
ADSL2+	1—1.5 Mbps	24—25 Mbps

ADSL2 and ADSL2+ support the following standards:

- LLCSNAP bridged 802.1q
- VC MUX bridged

Supported security devices with xDSL SFP can use PPP over Ethernet(PPPoE) to connect through ADSL lines only.

ADSL2 and ADSL2+

The ADSL2 and ADSL2+ standards were adopted by the ITU in July 2002. ADSL2 improves the data rate and reach performance, diagnostics, standby mode, and interoperability of ADSL modems.

ADSL2+ doubles the possible downstream data bandwidth, enabling rates of 20 Mbps on telephone lines shorter than 5000 feet (1.5 km).

ADSL2 uses seamless rate adaptation (SRA) to change the data rate of a connection during operation with no interruptions or bit errors. The ADSL2 transceiver detects changes in channel conditions—for example, the failure of another transceiver in a multicarrier link—and sends a message to the transmitter to initiate a data rate change. The message includes data transmission parameters such as the number of bits modulated and the power on each channel. When the transmitter receives the information, it transitions to the new transmission rate.

Example: Configuring ADSL SFP Interface on NFX350 Devices

IN THIS SECTION

Requirements

This example uses the following hardware and software components:

- NFX350 device running the Junos OS Release 19.4R1 version, which supports the reoptimized architecture.

Overview

In this example, you are configuring ADSL SFP interface on an NFX350 device with the following configurations:

- Physical interface - **ge-0/0/11**
- ADSL SFP options - **vpi3, vci34, and encap llcsnap-bridged-802dot1q**

> **NOTE:** Ensure that connectivity to the host is not lost during the configuration process.

Configuration

Step-by-Step Procedure

To configure ADSL SFP interfaces on NFX350 devices:

1. Connect to the host.

```
user@host> configure
[edit]
user@host#
```

2. Allocate hugepages:

```
user@host# run show system visibility memory
user@host# set system memory hugepages size 1024 count 5
```

Reboot the device.

3. Configure virtual interfaces:

```
user@host# set vmhost virtualization-options interfaces ge-1/0/3
user@host# set vmhost virtualization-options interfaces ge-1/0/4
user@host# commit
```

4. Create VLANs using VLAN IDs:

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan101 vlan-id 101
user@host# set vlans vlan200 vlan-id 200
user@host# set vlans vlan50 vlan-id 50
```

5. Configure interfaces:

```
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan50
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan101
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan200
user@host# set interfaces ge-0/0/11 native-vlan-id 50
user@host# set interfaces ge-0/0/11 dsl-sfp-options adsl-options vpi 3
user@host# set interfaces ge-0/0/11 dsl-sfp-options adsl-options vci 32
user@host# set interfaces ge-0/0/11 dsl-sfp-options adsl-options encap llcsnap-bridged-802dot1q
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching interface-mode trunk
```

```
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan50
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan101
user@host# set interfaces ge-1/0/3 vlan-tagging
user@host# set interfaces ge-1/0/3 unit 0 vlan-id 50
user@host# set interfaces ge-1/0/3 unit 0 family inet address 130.1.1.11/24
user@host# set interfaces ge-1/0/3 unit 0 family inet6 address 2001::1/64
```

6. Commit the configuration.

```
user@host# commit and-quit
user@host> exit
```

Results

From configuration mode, verify your configuration by entering the **show interfaces ge-0/0/11** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it:

```
[edit]
```

```
user@host# show interfaces ge-0/0/11
```

```
Physical interface: ge-0/0/11, Enabled, Physical link is Up
  Interface index: 163, SNMP ifIndex: 535
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, ADSL2P mode, Speed: 1000mbps,
 BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online, IEEE 802.3az Energy Efficient
Ethernet: Disabled, Auto-MDIX: Enabled
  ADSL status:
     Modem status  : Showtime (Adsl2plus)
     DSL mode      :     Auto    Annex A
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 08:b2:58:1e:c2:0e, Hardware address: 08:b2:58:1e:c2:0e
  Last flapped   : 2019-03-04 07:25:49 UTC (1w1d 22:55 ago)
  Input rate     : 1272 bps (2 pps)
  Output rate    : 1560 bps (2 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

  Logical interface ge-0/0/11.0 (Index 348) (SNMP ifIndex 536)
    Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
    Input packets : 0
    Output packets: 27874
Protocol eth-switch, MTU: 1514
```

RELATED DOCUMENTATION

VDSL2 Interfaces on NFX350 Devices

IN THIS SECTION

VDSL Interface Overview

IN THIS SECTION

Very-high-bit-rate digital subscriber line (VDSL) technology is part of the *x*DSL family of modem technologies that provide faster data transmission over a single flat untwisted or twisted pair of copper wires. The VDSL lines connect service provider networks and customer sites to provide high bandwidth applications (triple-play services) such as high-speed Internet access, telephone services like VoIP, high-definition TV (HDTV), and interactive gaming services over a single connection.

VDSL2 is an enhancement to G.993.1 (VDSL) and permits the transmission of asymmetric (half-duplex) and symmetric (full-duplex) aggregate data rates up to 100 Mbps on short copper loops using a bandwidth up to 17 MHz. The VDSL2 technology is based on the ITU-T G.993.2 (VDSL2) standard, which is the International Telecommunication Union standard describing a data transmission method for VDSL2 transceivers.

The VDSL2 uses discrete multitone (DMT) modulation. DMT is a method of separating a digital subscriber line signal so that the usable frequency range is separated into 256 frequency bands (or channels) of 4.3125 KHz each. The DMT uses the Fast Fourier Transform (FFT) algorithm for demodulation or modulation for increased speed.

VDSL2 interface supports Packet Transfer Mode (PTM). The PTM mode transports packets (IP, PPP, Ethernet, MPLS, and so on) over DSL links as an alternative to using Asynchronous Transfer Mode (ATM). PTM is based on the Ethernet in the First Mile (EFM) IEEE802.3ah standard.

VDSL2 provides backward compatibility with ADSL2 and ADSL2+ because this technology is based on both the VDSL1-DMT and ADSL2/ADSL2+ recommendations.

VDSL2 Vectoring Overview

Vectoring is a transmission method that employs the coordination of line signals that reduce crosstalk levels and improve performance. It is based on the concept of noise cancellation, like noise-cancelling headphones. The ITU-T G.993.5 standard, "Self-FEXT Cancellation (Vectoring) for Use with VDSL2 Transceivers," also known as G.vector, describes vectoring for VDSL2.

The scope of Recommendation ITU-T G.993.5 is specifically limited to the self-FEXT (far-end crosstalk) cancellation in the downstream and upstream directions. The FEXT generated by a group of near-end transceivers and interfering with the far-end transceivers of that same group is canceled. This cancellation takes place between VDSL2 transceivers, not necessarily of the same profile.

VDSL2 Network Deployment Topology

In standard telephone cables of copper wires, voice signals use only a fraction of the available bandwidth. Like any other DSL technology, the VDSL2 technology utilizes the remaining capacity to carry the data and multimedia on the wire without interrupting the line's ability to carry voice signals.

This example depicts the typical VDSL2 network topology deployed using NFX device.

A VDSL2 link between network devices is set up as follows:

1. Connect an end-user device such as a LAN, hub, or PC through an Ethernet interface to the customer premises equipment (CPE) (for example, an NFX device).

2. Connect the CPE to a DSLAM.

3. The VDSL2 interface uses either Gigabit Ethernet or fiber as second mile to connect to the Broadband Remote Access Server (B-RAS) as shown in Figure 6 on page 54.

4. The ADSL interface uses either Gigabit Ethernet (in case of IP DSLAM] as the “second mile” to connect to the B-RAS or OC3/DS3 ATM as the second mile to connect the B-RAS as shown in Figure 7 on page 54.

> **NOTE:** The VDSL2 technology is backward compatible with ADSL2 and ADSL2+. VDSL2 provides an ADSL2 and ADSL2+ interface in an ATM DSLAM topology and provides a VDSL2 interface in an IP or VDSL DSLAM topology.

The DSLAM accepts connections from many customers and aggregates them to a single, high-capacity connection to the Internet.

Figure 6 on page 54 shows a typical VDSL2 network topology.

Figure 6: Typical VDSL2 End-to-End Connectivity and Topology Diagram

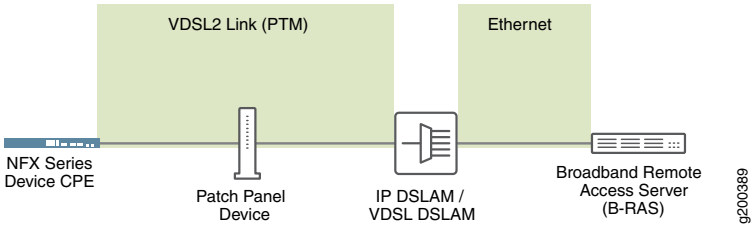
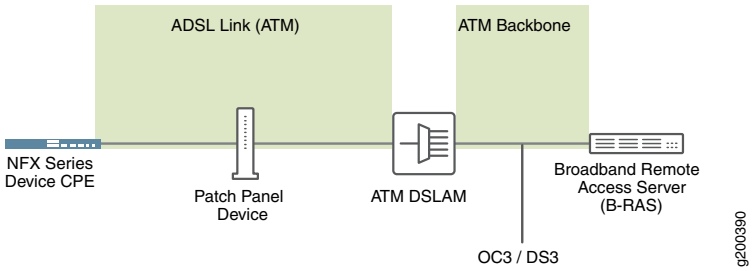


Figure 7 on page 54 shows a backward-compatible ADSL topology using ATM DSLAM.

Figure 7: Backward-Compatible ADSL Topology (ATM DSLAM)



VDSL2 Interface Support on NFX350 Devices

The VDSL2 interface is supported on the NFX Series devices listed in Table 9 on page 55. (Platform support depends on the Junos OS release in your installation.)

Table 9: VDSL2 Annex A and Annex B Features

Features	POTS
Devices	CPE-SFP-VDSL2
Supported annex operating modes	Annex A and Annex B*
Supported Bandplans	Annex A 998 Annex B 997 and 998
Supported standards	ITU-T G.993.2 and ITU-T G.993.5 (VDSL2)
Used in	North American network implementations
ADSL backward compatibility	G 992.3 (ADSL2) G 992.5 (ADSL2+)

NOTE: Only one CPE-SFP-VDSL2 device is supported at a time.

VDSL2 Interface Compatibility with ADSL Interfaces

VDSL2 interfaces on NFX Series devices are backward compatible with most ADSL2 and ADSL2+ interface standards. The VDSL2 interface uses Ethernet in the First Mile (EFM) mode or Packet Transfer Mode (PTM) and uses the named interface ge-0/0/10 and ge-0/0/11.

NOTE:

- The VDSL2 interface has backward compatibility with ADSL2 and ADSL2+.
- It requires around 60 seconds to switch from VDSL2 to ADSL2 and ADSL2+ or from ADSL2 and ADSL2+ to VDSL2 operating modes.

VDSL2 Interfaces Supported Profiles

A profile is a table that contains a list of pre-configured VDSL2 settings. Table 10 on page 56 lists the different profiles supported on the VDSL2 interfaces and their properties.

Table 10: Supported Profiles on the VDSL2 Interfaces

Profiles	Data Rate
8a	50
8b	50
8c	50
8d	50
12a	68
12b	68
17a	100
Auto	Negotiated (based on operating mode)

Example: Configuring VDSL SFP Interface on NFX350 Devices

IN THIS SECTION

Requirements

This example uses the following hardware and software components:

- NFX350 device running Junos OS Release 20.2R1.

Overview

In this example, you are configuring VDSL SFP interface on an NFX350 device with the following configurations:

- Physical interface - **ge-0/0/11**
- VDSL SFP options - **profile auto and carrier auto**

To configure VDSL SFP interface on NFX250 devices, you must configure JDM, vSRX, and vJunos0.

> **NOTE:** Ensure that connectivity to the host is not lost during the configuration process.

Configuration

Step-by-Step Procedure

To configure VDSL SFP interfaces on NFX350 devices:

1. Connect to the host.

```
user@host> configure
[edit]
user@host#
```

2. Configure virtual interfaces:

```
user@host# set vmhost virtualization-options interfaces ge-1/0/3
user@host# set vmhost virtualization-options interfaces ge-1/0/4
user@host# commit
```

3. Create VLANs using VLAN IDs:

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan101 vlan-id 101
user@host# set vlans vlan200 vlan-id 200
user@host# set vlans vlan50 vlan-id 50
```

4. Configure interfaces:

```
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan50
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan101
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan200
user@host# set interfaces ge-0/0/11 native-vlan-id 50
user@host# set interfaces ge-0/0/11 dsl-sfp-options vdsl-options profile auto
user@host# set interfaces ge-0/0/11 dsl-sfp-options vdsl-options carrier auto
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan50
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan101
user@host# set interfaces ge-1/0/3 vlan-tagging
user@host# set interfaces ge-1/0/3 unit 0 vlan-id 50
user@host# set interfaces ge-1/0/3 unit 0 family inet address 130.1.1.11/24
user@host# set interfaces ge-1/0/3 unit 0 family inet6 address 2001::1/64
```

5. Commit the configuration.

```
user@host# commit and-quit
user@host> exit
```

Results

From configuration mode, verify your configuration by entering the **show interfaces ge-0/0/11** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it:

```
[edit]
```

```
user@host# show interfaces ge-0/0/11
```

```
Physical interface: ge-0/0/11, Enabled, Physical link is Up
  Interface index: 171, SNMP ifIndex: 0
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, VDSL2 mode, Speed: 1000mbps,
 BPDU Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,
MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online, IEEE
802.3az Energy Efficient Ethernet: Disabled, Auto-MDIX: Enabled
  VDSL status:
    Modem status  : Showtime (Profile-17a)
    VDSL profile  :     Auto    Annex B
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 08:b2:58:1e:c2:0a, Hardware address: 08:b2:58:1e:c2:0a
  Last flapped   : 2020-06-12 06:12:59 UTC (00:42:09 ago)
  Input rate     : 600 bps (1 pps)
  Output rate    : 600 bps (1 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

  Logical interface ge-0/0/10.0 (Index 77) (SNMP ifIndex 0)
    Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
    Input packets : 0
    Output packets: 0
    Protocol eth-switch, MTU: 1514
```

RELATED DOCUMENTATION

- *NFX250 Overview*
- *JDM Architecture Overview*
- *JDM CLI Overview*

5

CHAPTER

Configuring Solid State Disk

Configuring the Solid State Disk on NFX350 Device

NFX350 devices support two external Solid State Disks (SSDs) for customer data, logging and VNF data. These two SSDs work independently.

You can configure the Solid State Disk (SSD) on the NFX350 device for local persistent storage.

NOTE:

- It is highly recommended not to use third-party external SSDs on NFX350 devices.
- You must plug in the external SSD when the NFX350 device is in powered off state. You can use the SSD only after you initialize and add the SSD to the NFX350 device.
- When an external SSD is initialized for a particular NFX350 device, you can use that SSD with that particular device only.
- If an external SSD is present during the installation (USB/clean-install/zeroize), then the SSD is initialized to be used with the NFX350 current device. If the external SSD is not present during the installation and is inserted later, then use the **request vmhost storage external-disk-1 initialize [force]** commands to initialize the external disk.

To initialize and add an SSD:

1. Log in to the JCP CLI and enter configuration mode:

   ```
   root@host% cli
   root@host> configure
   ```

2. Initialize an SSD:

   ```
   root@host# request vmhost storage external-ssd initialize slot 0 public-dir-name public-disk0
   ```

   There are two slots for SSD, slot 0 and slot 1. The disk name should be either *public-disk0* or *public-disk1* based on the SSD slot.

   NOTE: You can store VNF images in the **/var/public-disk1** folder.

3. Add an SSD:

   ```
   root@host# request vmhost storage external-ssd add slot 0
   ```

4. Verify whether the SSD is added:

```
root@host# run show virtual-network-functions storage
```

```
Filesystem              Size   Avail   Used   Use%
/var/public             19G    7.4G    11G    60%
/var/public-disk1       734G   697G    69M    1%
```

In the output message, **/var/public** shows details of the internal SSD and **/var/public-disk1** shows details of the external SSD that is plugged in slot1.

5. (Optional) Re-initialize the SSD:

   ```
   root@host# request vmhost storage external-ssd initialize slot 1 public-dir-name public-disk1 force
   ```

   NOTE: Upgrade using USB re-initializes and adds the external SSD.

To remove the SSD:

1. Remove the SSD:

   ```
   root@host# request vmhost storage external-ssd remove slot 0
   ```

2. Power off the device.

3. Remove the SSD.

NOTE: SSD is not formatted when you remove it. To erase the data before removing the SSD, re-initialize the SSD.

CHAPTER 6

Configuring Security

IP Security on NFX Devices

IN THIS SECTION

Overview

IPsec provides network-level data integrity, data confidentiality, data origin authentication, and protection from replay. IPsec can protect any protocol running over IP on any medium or a mixture of application protocols running on a complex combination of media. IPsec provides security services at the network layer of the Open Systems Interconnection (OSI) model by enabling a system to select required security protocols, determine the algorithms to use for the security services, and implement any cryptographic keys required to provide the requested services. IPsec is standardized by International Engineering Task Force (IETF).

IPsec protects one or more paths between a pair of hosts or security gateways, or between a security gateway and a host. It achieves this by providing a secure way to authenticate senders/receivers and encrypt IP version 4 (IPv4) and version 6 (IPv6) traffic between network devices.

The key concepts of IPsec include:

- Security associations (SAs)—An SA is a set of IPsec specifications negotiated between devices that are establishing an IPsec relationship. These specifications include preferences for the type of authentication and encryption, and the IPsec protocol that is used to establish the IPsec connection. A security association is uniquely identified by a security parameter index (SPI), an IPv4 or IPv6 destination address, and a security protocol (AH or ESP). IPsec security associations are established either manually through configuration statements, or dynamically by IKE negotiation. For more information about SAs, see Security Associations.
- IPsec key management—VPN tunnels are built using IPsec technology. Virtual private network (VPN) tunnels operate with three kinds of key creation mechanisms such as Manual Key, AutoKey Internet Key Exchange (IKE) , and Diffie-Hellman (DH) Exchange. NFX150 devices support IKEv1 and IKEv2. For more information about IPsec key management, see IPsec Key Management.
- IPsec security protocols—IPsec uses two protocols to secure communications at the IP layer:
  - Authentication Header (AH)—A security protocol for authenticating the source of an IP packet and verifying the integrity of its content.

- Encapsulating Security Payload (ESP)—A security protocol for encrypting the entire IP packet and authenticating its content.

For more information about IPsec security protocols, see IPsec Security Protocols.

- IPsec tunnel negotiation—To establish an IKE IPsec tunnel, two phases of negotiation are required:
  - In Phase 1, the participants establish a secure connection to negotiate the IPsec SAs.
  - In Phase 2, the participants negotiate the IPsec SAs for encrypting and authenticating the ensuing exchanges of user data.

For more information about IPsec tunnel negotiation, see IPsec Tunnel Negotiation.

Table 11 on page 67 lists the IPsec features supported on NFX Series devices.

Table 11: IPsec Features Supported on NFX Series Devices

Features	Reference
AutoVPN Spoke	Understanding Spoke Authentication in AutoVPN Deployments
Auto Discovery VPN (ADVPN) Partner **NOTE:** On NFX150 devices, you cannot configure ADVPN Suggester.	Understanding Auto Discovery VPN
Site-to-Site VPN and Dynamic Endpoints	Understanding IPsec VPNs with Dynamic Endpoints
Route-based VPN **NOTE:** NFX150 devices do not support policy-based VPNs.	Understanding Route-Based IPsec VPNs
NAT-T	Understanding NAT-T
Dead Peer Detection	Understanding VPN Monitoring

Configuring Security

IN THIS SECTION

On NFX150 devices, security is implemented by using IP security (IPsec). The configuration process of IP security (IPsec) includes the following tasks:

Configuring Interfaces

To enable IPsec on a LAN or WAN, you must configure interfaces to provide network connectivity and data flow.

> **NOTE:** To configure IPsec, use the FPC1 interface.

To configure interfaces, complete the following steps:

1. Log in to the JCP CLI and enter configuration mode:

   ```
   root@host% cli
   root@host> configure
   ```

2. Enable VLAN tagging support on the logical interface:

   ```
   root@host# set interfaces interface-name vlan-tagging
   ```

3. Assign a VLAN ID to the logical interface:

   ```
   root@host# set interfaces interface-name unit logical-interface-unit-number vlan-id vlan-id
   ```

4. Assign an IPv4 address to the logical interface:

   ```
   root@host# set interfaces interface-name unit logical-interface-unit-number family inet address
   interface-address
   ```

5. Assign an IPv6 address to the logical interface:

```
root@host# set interfaces interface-name unit interface-logical-unit-number family inet6 address
interface-address
```

Configuring Routing Options

Routing capabilities and features that are not specific to any particular routing protocol are collectively called protocol-independent routing properties. These features often interact with routing protocols. In many cases, you combine protocol-independent properties and routing policy to achieve a goal. For example, you define a static route using protocol-independent properties, and then you use a routing policy to re-distribute the static route into a routing protocol, such as BGP, OSPF, or IS-IS.

Protocol-independent routing properties include:

- Static, aggregate, and generated routes
- Global preference
- Martian routes
- Routing tables and routing information base (RIB) groups

To configure the routing table groups into which the interface routes are imported, complete the following steps:

1. Configure RIB and static route:

   ```
   root@host# set routing-options rib rib-name static route ip-address/prefix-length next-hop ip-address
   ```

2. Configure static route:

   ```
   root@host# set routing-options static route ip-address/prefix-length next-hop ip-address
   ```

Configuring Security IKE

IPsec uses the Internet Key Exchange (IKE) protocol to authenticate the IPsec peers, to negotiate the security association (SA) settings, and to exchange IPsec keys. The IKE configuration defines the algorithms and keys used to establish the secure IKE connection with the peer security gateway.

You can configure IKE traceoptions for debugging and managing the IPsec IKE.

To configure IKE traceoptions, complete the following steps:

1. Specify the maximum size of the trace file:

   ```
   root@host# set security ike traceoptions file size file-size
   ```

2. Specify the parameters to trace information for IKE:

   `root@host#` **set security ike traceoptions flag all**

3. Specify the level of trace information for IKE:

   `root@host#` **set security ike traceoptions level *level 7-15***

You can configure one or more IKE proposals. Each proposal is a list of IKE attributes to protect the IKE connection between the IKE host and its peer.

To configure IKE proposal, complete the following steps:

1. Configure pre-shared-keys as an authentication method for the IPsec IKE proposal:

   > **NOTE:** When you configure IPsec for secure communications in the network, the peer devices in the network must have at least one common authentication method. Only one authentication method can be used between a pair of devices, regardless of the number of authentication methods configured.

   `root@host#` **set security ike proposal *ike-proposal-name* authentication-method pre-shared-keys**

2. Define a Diffie-Hellman group (dh-group) for the IKE proposal:

   `root@host#` **set security ike proposal *ike-proposal-name* dh-group group14**

3. Configure an authentication algorithm for the IKE proposal:

   `root@host#` **set security ike proposal *ike-proposal-name* authentication-algorithm sha-256**

4. Define an encryption algorithm for the IKE proposal:

   `root@host#` **set security ike proposal *ike-proposal-name* encryption-algorithm aes-256-cbc**

5. Set a lifetime for the IKE proposal in seconds:

   `root@host#` **set security ike proposal *ike-proposal-name* lifetime-seconds *180 to 86400 seconds***

After configuring one or more IKE proposals, you must associate these proposals with an IKE policy. An IKE policy defines a combination of security parameters (IKE proposals) to be used during IKE negotiation. It defines a peer address and the proposals needed for that connection. Depending on which authentication method is used, it defines the preshared key for the given peer. During the IKE negotiation, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

To configure IKE policy, complete the following steps:

1. Define an IKE policy with first phase mode:

   `root@host#` **set security ike policy *ike-policy-name* mode aggressive**

2. Define a set of IKE proposals:

   `root@host#` **set security ike policy *ike-policy-name* proposals *proposal-name***

3. Define a pre-shared key for IKE:

   `root@host#` **set security ike policy *ike-policy-name* pre-shared-key ascii-text *text-format***

Configure an IKE gateway to initiate and terminate network connections between a firewall and a security device.

To configure IKE gateway, complete the following steps:

1. Configure an IKE gateway with an IKE policy:

   `root@host#` **set security ike gateway *gateway-name* ike-policy *ike-policy-name***

2. Configure an IKE gateway with an address or hostname of the peer:

   `root@host#` **set security ike gateway *gateway-name* address *address-or-hostname-of-peer***

3. Enable dead peer detection (DPD) feature to send DPD messages periodically:

   `root@host#` **set security ike gateway *gateway-name* dead-peer-detection always-send**

4. Configure the local IKE identity:

   `root@host#` **set security ike gateway *gateway-name* local-identity <inet | inet6 | key-id | hostname | user-at-hostname | distinguished-name>**

5. Configure the remote IKE identity:

   `root@host#` **set security ike gateway *gateway-name* remote-identity <inet | inet6 | key-id | hostname | user-at-hostname | distinguished-name>**

6. Configure an external interface for IKE negotiations:

   `root@host#` **set security ike gateway *gateway-name* external-interface ge-1/0/1.0**

7. Configure username of the client:

   `root@host#` **set security ike gateway *gateway-name* client username *client-username***

8. Configure password of the client:

   `root@host#` **set security ike gateway *gateway-name* client password *client-password***

Configuring Security IPsec

IPsec is a suite of related protocols that provides network-level data integrity, data confidentiality, data origin authentication, and protection from replay. IPsec can protect any protocol running over IP on any medium or a mixture of application protocols running on a complex combination of media.

Configure an IPsec proposal, which lists protocols and algorithms or security services to be negotiated with the remote IPsec peer.

To configure an IPsec proposal, complete the following steps:

1. Define an IPsec proposal and protocol for the proposal:

   `root@host#` **set security ipsec proposal *ipsec-proposal-name* protocol esp**

2. Define an authentication algorithm for the IPsec proposal:

   `root@host#` **set security ipsec proposal *ipsec-proposal-name* authentication-algorithm hmac-sha-256-128**

3. Define an encryption algorithm for the IPsec proposal:

   `root@host#` **set security ipsec proposal *ipsec-proposal-name* encryption-algorithm aes-256-cbc**

4. Set a lifetime for the IPsec proposal in seconds:

   `root@host#` **set security ipsec proposal *ipsec-proposal-name* lifetime-seconds *180..86400 seconds***

After configuring one or more IPsec proposals, you must associate these proposals with an IPsec policy. An IPsec policy defines a combination of security parameters (IPsec proposals) used during IPsec negotiation. It defines Perfect Forward Secrecy (PFS) and the proposals needed for the connection. During the IPsec negotiation, IPsec searches for a proposal that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match.

To configure IPsec policies, complete the following steps:

1. Define an IPsec policy, a perfect forward secrecy, and a Diffie-Hellman group for the policy:

   `root@host#` **set security ipsec policy *ipsec-policy-name* perfect-forward-secrecy keys group14**

2. Define a set of IPsec proposals for the policy:

   `root@host#` **set security ipsec policy *ipsec-policy-name* proposals *proposal-name***

Configure an IPsec virtual private network (VPN) to provide a means for securely communicating among remote computers across a public WAN such as the Internet. A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication while passing through the WAN, the two participants create an IPsec tunnel. For more information, see IPsec VPN Overview.

To configure IPsec VPN, complete the following steps:

1. Define an IKE gateway for the IPsec VPN:

   `root@host#` **set security ipsec vpn *vpn-name* ike gateway *remote-gateway-name***

2. Define an IPsec policy for the IPsec VPN:

   `root@host#` **set security ipsec vpn *vpn-name* ike ipsec-policy *ipsec-policy-name***

3. Define a local traffic selector for the IPsec VPN:

   `root@host#` **set security ipsec vpn *vpn-name* traffic-selector *traffic-selector-name* local-ip *local-traffic-selector-ip-address***

4. Define a remote traffic selector for the IPsec VPN:

   `root@host#` **set security ipsec vpn *vpn-name* traffic-selector *traffic-selector-name* remote-ip *remote-traffic-selector-ip-address***

5. Define a criteria to establish IPsec VPN tunnels:

   `root@host#` **set security ipsec vpn *vpn-name* establish-tunnels on-traffic**

Configuring Security Policies

A security policy controls the traffic flow from one zone to another zone by defining the kind of traffic permitted from specified IP sources to specified IP destinations at scheduled times. Policies allow you to deny, permit, reject, encrypt and decrypt, authenticate, prioritize, schedule, filter, and monitor the traffic attempting to cross from one security zone to another. You can decide which users and what data can enter and exit, and when and where they can go.

To configure security policies, complete the following steps:

1. Configure security policy match criteria for the source address:

   `root@host#` **set security policies from-zone *from-zone-name* to-zone *to-zone-name* policy *policy-name* match source-address any**

2. Configure security policy match criteria for the destination address:

   `root@host#` **set security policies from-zone *from-zone-name* to-zone *to-zone-name* policy *policy-name* match destination-address any**

3. Configure security policy application:

   `root@host#` **set security policies from-zone *from-zone-name* to-zone *to-zone-name* policy *policy-name* match application any**

4. Set security policy match criteria:

   `root@host#` **set security policies from-zone *from-zone-name* to-zone *to-zone-name* policy *policy-name* match then permit**

Configuring Security Zones

Security zones are the building blocks for policies. They are logical entities to which one or more interfaces are bound. Security zones provide a means of distinguishing groups of hosts (user systems and other hosts, such as servers) and their resources from one another in order to apply different security measures to them. For information, see *Understanding Security Zones*.

To configure security zones, complete the following steps:

1. Configure security zones with system services:

   `root@host#` **set security zones security-zone *zone-name* host-inbound-traffic system-services all**

2. Define protocols for security zones:

```
root@host# set security zones security-zone zone-name host-inbound-traffic protocols all
```

3. Configure interfaces for security zones:

```
root@host# set security zones security-zone zone-name interfaces interface-name
```

CHAPTER 7

Configuring Virtual Network Functions

Prerequisites to Onboard Virtual Network Functions on NFX350 Devices

You can onboard and manage Juniper Virtual Network Functions (VNFs) and third-party VNFs on NFX devices through the Junos Control Plane (JCP).

> **NOTE:** This topic provides general guidelines to qualify VNFs on NFX350 devices. Before onboarding a VNF, you must test the VNF according to your use case scenario.

NFX350 Device Prerequisites to Onboard a VNF

To onboard VNFs on NFX350, the device must be on either Hybrid mode or Compute mode. The number of VNFs that you can onboard on the device depends on the system resources such as CPUs and system memory that are available on the mode that the device is operating. For more information about the performance modes, see "NFX350 Overview" on page 17.

Before you onboard the VNF, check the following NFX350 device capabilities:

- Check the current performance mode of the device by using the **show vmhost mode** command. The NFX350 device must be in either Compute or Hybrid mode when you run the **show vmhost mode** command.
- Check the available system memory by using the **show system visibility memory** command.

  Table 12 on page 77 lists the possible memory availability for VNF usage for the NFX350 models.

  Table 12: Memory Availability for VNF Usage (Junos OS 19.4R1 Release)

Model	Memory Availability for VNF Usage	Hugepages Availability for VNF Usage
NFX350-S1	32 GB	7 1G hugepages
NFX350-S2	64 GB	23 1G hugepages
NFX350-S3	128 GB	62 1G hugepages

- Check the available CPUs and its status by using the **show system visibility cpu** command. Use the **show vmhost mode** command to check the available CPUs in the current performance mode of the device.

  Table 13 on page 78 lists the CPUs available for VNF usage for the NFX350 models.

Table 13: CPUs Available for VNF Usage (Junos OS 19.4R1 Release)

Model	CPUs Available for VNF Usage		
	Throughput Mode	Hybrid Mode	Compute Mode
NFX350-S1	0	8	10
NFX350-S2	0	10	14
NFX350-S3	0	14	20

NOTE: When you change the performance mode of the device, it is recommended to check the availability of the CPUs for VNFs.

NOTE: On NFX350 devices, it is recommended to use external SSD for storing VNF images or files.

For more information, see "Configuring VNFs on NFX350 Devices" on page 84.

VNF Prerequisites to Onboard on an NFX350 Device

To onboard a VNF on an NFX350 device, the following VNF properties should be met:

NOTE: For VNF production deployment, it is recommended to use external hard disk.

- KVM based hypervisor deployment
- OVS or Virtio interface drivers
- raw or qcow2 VNF file types
- Support of up to a maximum of 8 user interfaces

Following are the optional prerequisites to onboard a VNF:

- (Optional) SR-IOV
- (Optional) CD-ROM and USB configuration drives

- (Optional) Hugepages for memory requirements if VNF wants to access OVS.

Validate the VNFs

To validate and qualify the VNFs, you must ensure the following:

- The configuration commit succeeds for the VNF.
- The **show virtual-network-functions** command output displays the VNF entry.
- The **show system visibility vnf** command output displays the VNF properties and interfaces that are configured.
- The **show vmhost network nfv-back-plane** command displays all interfaces that are connected to the OVS bridges with the state **up/up**. The **show system visibility network** command displays all the VNF interfaces.
- Connection to the console of the VNF succeeds and VNF boot up or login prompt is displayed.
- When you are logged into the VNF, use the **request virtual-network-function console** command for the VNF to display all the interfaces that are configured.
- The **show virtual-network-functions** command lists the VNF that are alive when the internal management interface is configured with DHCP client inside the VNF.
- VNF interfaces on the OVS bridge show **tx/rx** statistics when the traffic is ingressed or egressed from the VNF.
- VNF should restart successfully when a restart is initiated from within the VNF or by using the **request virtual-network-functions restart *vnf-name*** command.

For sample configuration of third-party VNFs, see "Example: Configuring Service Chaining for LAN to WAN Routing through Third-party VNFs on NFX350 Devices" on page 161.

Sample Output

- show virtual-network-functions

```
root@host> show virtual-network-functions

ID       Name                                              State      Liveliness
-
5        vsrx                                               Running    down
1        vjunos0                                            Running    alive
```

The **Liveliness** is alive when there is a management connectivity to the VNF. The **State** should be **Running** to show that the VNF is up.

- show system visibility vnf

```
root@host> show system visibility vnf
```

```
List of VNFs
-
ID   Name                                    State
- - -
5    vsrx                                    Running

VNF Memory Usage
-
Name                                  Maximum Memory (KiB)  Used Memory (KiB)
 Used 1G Hugepages  Used 2M Hugepages
- - - - -
vsrx                                  4194304               49715
 4                 0

VNF CPU Statistics (Time in ms)
-
Name                                 CPU Time           System Time  User Time
- - - -
vsrx                                 164425446          3214840      197880

VNF MAC Addresses
-
VNF                                     MAC
- -
centos1_ethdef0                         9C:CC:83:BD:8C:40
centos1_ethdef1                         9C:CC:83:BD:8C:46
centos1_eth2                            9C:CC:83:BD:8C:41
vsrx_ethdef0                            9C:CC:83:BD:8C:42
vsrx_ethdef1                            9C:CC:83:BD:8C:43
vsrx_eth2                               9C:CC:83:BD:8C:45
vsrx_eth3                               9C:CC:83:BD:8C:44

VNF Internal IP Addresses
-
VNF                                     IP
- -
vsrx                                    192.0.2.100

VNF Interfaces
```

```
-
VNF                  Interface Type      Source       Model      MAC
    IPv4-address
- - - - - - -
vsrx               vnet6     network   default      virtio     9c:cc:83:bd:8c:42
 -
vsrx               vnet7     bridge    eth0br       virtio     9c:cc:83:bd:8c:43
 -
vsrx                vsrx_eth2 vhostuser -          virtio     9c:cc:83:bd:8c:45
 -

VNF Disk Information
-
VNF                           Disk        File
- - -
vsrx                          vda
/var/public/junos-vsrx3-x86-64-19.4R1.12.qcow2

VNF Disk Usage
-
VNF                  Disk      Read Req   Read Bytes   Write Req  Write Bytes
- - - - - -
vsrx                 vda       220376     1951876096   24927      185393152

VNF Port Statistics
-
VNF                  Port      Rcvd Bytes   Rcvd Packets Rcvd Error Rcvd Drop
Trxd Bytes   Trxd Packets Trxd Error Trxd Drop
- - - - - - - - - -
vsrx                 vnet6     4113582      79122        0          0         0
            0            0          0
vsrx                 vnet7     3399770129   47653525     0          34631     0
            0            0          0
vsrx                 vsrx_eth2 3724         65           0          0
4372         73           0          0
```

- request virtual-network-functions vsrx console

root@host> **request virtual-network-functions vsrx console**

```
Internal instance: vsrx
Connected to domain vsrx
Escape character is ^]
```

```
FreeBSD/amd64 (Amnesiac) (ttyu0)

login: root
Password:
Last login: Tue Mar 17 16:10:40 on ttyu0

- JUNOS 19.4R1.12 Kernel 64-bit XEN JNPR-11.0-20191115.14c2ad5_buil
root@:~ #
root@:~ # cli
hroot> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
gr-0/0/0                up    up
ip-0/0/0                up    up
lsq-0/0/0               up    up
lt-0/0/0                up    up
mt-0/0/0                up    up
sp-0/0/0                up    up
sp-0/0/0.0              up    up   inet
                                   inet6
sp-0/0/0.16383          up    up   inet
ge-0/0/1                up    up
ge-0/0/1.0              up    up   inet     10.10.10.1/24

root> show configuration | display set |match fxp0
set system services web-management http interface fxp0.0
set system services web-management https interface fxp0.0
set interfaces fxp0 unit 0 family inet dhcp

root> show interfaces terse | match fxp0
fxp0                    up    up
fxp0.0                  up    up   inet     192.0.2.100/24
```

- show system visibility memory

`root@host>` **show system visibility memory**

```
Memory Information
------------------

Virtual Memory:
---------------
Total       (KiB): 131042784
```

```
Used        (KiB): 64842740
Available   (KiB): 66595824
Free        (KiB): 66200044
Percent Used     : 49.2

Huge Pages:
------------
Total 1GiB Huge Pages:      18
Free 1GiB Huge Pages:       0
Configured 1GiB Huge Pages: 0
Total 2MiB Huge Pages:      20481
Free 2MiB Huge Pages:       0
Configured 2MiB Huge Pages: 0

Hugepages Usage:
-----------------------------------------------------------------------------------------------------
Name                              Type                                 Used 1G
Hugepages  Used 2M Hugepages
---------------------------------- ----------------------------------
------------------ ------------------
ovs-vswitchd                      other process                        18
         0
srxpfe                            other process                        6
         20481
```

In the output message, check **Free** and **Configured** fields under **Virtual Memory** and **Huge Pages** sections for the memory availability.

- show vmhost mode

```
root@host> show vmhost mode
```

```
Mode:
--------
Current Mode: hybrid

CPU Allocations:
Name                                Configured
Used
-----------------------------------------------------------------------------------------------------
Junos Control Plane                 16
16,9
Juniper Device Manager              16                                   16
LTE                                 16                                    -
```

```
NFV Backplane Control Path          16                                          16
NFV Backplane Data Path             1,2,3,4
1,2,3,4
Layer 2 Control Path                -                                           -
Layer 2 Data Path                   -                                           -
Layer 3 Control Path                0                                           0
Layer 3 Data Path                   5,6,7,8
5,6,7,8
CPUs available for VNFs             9,10,11,12,13,14,15,25,26,27,28,29,30,31 -
CPUs turned off                     17,18,19,20,21,22,23,24                     -

Memory Allocations:
Name                                Configured
Used
-------------------------------------------------------------------------------------------------------------
Junos Control Plane (mB)            2048
2009
NFV Backplane 1G hugepages          12                                          18
NFV Backplane 2M hugepages          -                                           0
Layer 2 1G hugepages                -                                           -
Layer 2 2M hugepages                -                                           -
Layer 3 1G hugepages                6                                           6
Layer 3 2M hugepages                20481
20481
```

In the output message, check the **Current Mode** field under the **Mode** section for the current performance mode of the device. Check the **CPUs available for VNFs** field under the **CPU Allocations** section for the CPU availability.

Configuring VNFs on NFX350 Devices

IN THIS SECTION

The NFX350 devices enable you to instantiate and manage virtualized network functions (VNFs) from the Junos Control Plane (JCP). The JCP supports the creation and management of third-party VNFs.

Load a VNF Image

To configure a VNF, you must log in to the JCP:

user@host:~ # cli

user@host>

To load a VNF image on the device from a remote location, use the **file-copy** command.

> **NOTE:** You can save the VNF image in the **/var/public** directory if you are using up to two VNFs. If you are using more than two VNFs, save the files on an external SSD. If you are using an external SSD for VNFs, make sure to initialize and add the SSD to the device. For more information, see "Configuring the Solid State Disk on NFX350 Device" on page 63.

user@host> file copy *source-address* /var/public

For example:

user@host> file copy scp://192.0.2.0//tftpboot/centos.img /var/public

Alternatively, you can load a VNF image by using the NETCONF command, **file-put**.

Prepare the Bootstrap Configuration

You can bootstrap a VNF by attaching a CD-ROM, a USB storage device, or a config drive that contains a bootstrap-config ISO file.

For an example of creating an ISO file, see the procedure in Creating a vSRX Bootstrap ISO Image. The procedure might differ based on the operating system (for example, Linux, Ubuntu) that you use to create the ISO file.

A bootstrap configuration file must contain an initial configuration that allows the VNF to be accessible from an external controller, and accepts SSH, HTTP, or HTTPS connections from an external controller for further runtime configurations.

NOTE:

- The system saves the bootstrap-config ISO file in the **/var/public** folder. The file is saved only if the available space in the folder is more than double the total size of the contents in the file. If the available space in the folder is not sufficient, an error message is displayed when you commit the configuration.
- When you reboot the system, the system generates a new bootstrap-config ISO file and replaces the existing ISO file with the new ISO file on the VNF.

Allocate CPUs for a VNF

Table 13 on page 78 lists the CPUs available for VNF usage for the NFX350 models.

Table 14: CPUs Available for VNF Usage (Junos OS 19.4R1 Release)

Model	CPUs Available for VNF Usage		
	Throughput Mode	Hybrid Mode	Compute Mode
NFX350-S1	0	8	10
NFX350-S2	0	10	14
NFX350-S3	0	14	20

NOTE: When you change the performance mode of the device, it is recommended to check the availability of the CPUs for VNFs.

To check the CPU availability and its status:

`user@host>` **show system visibility cpu**

```
CPU Statistics (Time in sec)
--------------------------------------------------------------------------------
CPU Id User Time System Time Idle Time Nice Time IOWait Time Intr. Service Time
------ --------- ----------- --------- --------- ----------- ------------------
0      7762      1475        60539     0         84          0
1      191       511         70218     0         10          0
2      102       32          70841     0         12          0
3      0         0           70999     0         0           0
4      0         0           70999     0         0           0
5      0         0           70999     0         0           0
6      70949     0           50        0         0           0
7      9005      532         59602     0         0           0
8      23        7           70966     0         0           0
9      21        7           70969     0         0           0
10     20        6           70969     0         0           0
11     18        6           70970     0         0           0

CPU Usages
----------------
CPU Id CPU Usage
------ ---------
0      17.899999999999999
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      100.0
7      15.199999999999999
8      0.0
9      0.0
10     0.0
11     0.0

CPU Pinning Information
```

```
-------------------------------------
Virtual Machine             vCPU CPU
--------------------------- ---- ---
vjunos0                     0    0


System Component                CPUs
------------------------------- --------
ovs-vswitchd                    0, 6
```

user@host> **show vmhost mode**

```
Starting network management services: snmpd libvirtMib_subagent.
Synchronizing UEFI key-store:
Failed to get revocation list: 2
Juniper Dev keys are not revoked. Doing nothing
cp: cannot stat '/var/platform/lte_vm_xml_params': No such file or directory
rm: cannot remove '/lib/udev/rules.d/lte_usb.rules': No such file or directory

Mode:
--------
Current Mode: compute

CPU Allocations:
Name                                Configured                                Used
-------------------------------------------------------------------------------------------------------------
Junos Control Plane                  8                                         3,8
Juniper Device Manager               8                                         8
LTE                                  8                                         -
NFV Backplane Control Path           8                                         8
NFV Backplane Data Path              1                                         1
Layer 2 Control Path                 -                                         -
Layer 2 Data Path                    -                                         -
Layer 3 Control Path                 0                                         0
Layer 3 Data Path                    2                                         2
CPUs available for VNFs              3,4,5,6,7,11,12,13,14,15                  -
CPUs turned off                      9,10                                      -

Memory Allocations:
Name                                Configured                                Used
-------------------------------------------------------------------------------------------------------------
Junos Control Plane (mB)             2048                                      2011
NFV Backplane 1G hugepages           4                                         8
```

```
NFV Backplane 2M hugepages          -                                    0
Layer 2 1G hugepages                -                                    -
Layer 2 2M hugepages                -                                    -
Layer 3 1G hugepages                4                                    4
Layer 3 2M hugepages                5633                                 5377
```

The **CPUs available for VNFs** section in the output message shows the CPUs that are available to onboard VNFs.

> **NOTE:** vjunos0 is a system VNF, you cannot modify the CPU allocation for the vjunos0.

To specify the number of virtual CPUs that are required for a VNF:

1. Specify the number of CPUs required for the VNF:

   `user@host#` **set virtual-network-functions *vnf-name* virtual-cpu count *number***

2. Connect a virtual CPU to a physical CPU:

   `user@host#` **set virtual-network-functions *vnf-name* virtual-cpu *vcpu-number* physical-cpu *pcpu-number***

3. Commit the configuration:

   `user@host#` **commit**

The physical CPU number can be either a number or a number range. By default, a VNF is allocated one virtual CPU that is not connected to any physical CPU.

> **NOTE:** You cannot change the CPU configuration of a VNF while the VNF is running. You must restart the VNF for the changes to take effect.

To enable hardware virtualization or hardware acceleration for VNF CPUs:

`user@host#` **set virtual-network-functions *vnf-name* virtual-cpu *features* hardware-virtualization**

Allocate Memory for a VNF

By default, a certain amount of memory is allocated for VNFs. Table 12 on page 77 lists the possible memory availability for VNF usage for the NFX350 models.

Table 15: Memory Availability for VNF Usage

Model	Total System Memory (Junos OS 19.4R1 Release)	Hugepages Availability for VNF Usage (Junos OS 19.4R1 Release)
NFX350-S1	32 GB	7 1G hugepages
NFX350-S2	64 GB	23 1G hugepages
NFX350-S3	128 GB	62 1G hugepages

To check the available memory:

user@host> show system visibility memory

```
Memory Information
------------------

Virtual Memory:
---------------
Total       (KiB): 15914364
Used        (KiB): 13179424
Available   (KiB): 3087076
Free        (KiB): 2734940
Percent Used     : 80.6

Huge Pages:
------------
Total 1GiB Huge Pages:      7
Free 1GiB Huge Pages:       5
Configured 1GiB Huge Pages: 5
Total 2MiB Huge Pages:      1376
Free 2MiB Huge Pages:       1
Configured 2MiB Huge Pages: 0

Hugepages Usage:
-----------------------------------------------------------------------------------------------------
Name                              Type                                  Used 1G
Hugepages  Used 2M Hugepages
```

```
------------------------------- ----------------------------------
----------------- ------------------
srxpfe                          other process                     1
       1375
ovs-vswitchd                    other process                     2
       0
```

NOTE: vjunos0 is a system VNF, you cannot modify the memory allocation for the vjunos0.

To specify the maximum primary memory that the VNF can use:

`user@host#` **set virtual-network-functions *vnf-name* memory size *size***

NOTE: You cannot change the memory configuration of a VNF while the VNF is running. You must restart the VNF for the changes to take effect.

Configure Interfaces and VLANs for a VNF

You can configure a VNF interface and attach the interface to a physical NIC port, a management interface, or VLANs.

To attach a VNF interface to a physical NIC port by using the SR-IOV virtual function:

`user@host#` **set virtual-network-functions *vnf-name* interfaces *interface-name* mapping *physical-interface-name* virtual-function [vlan-id *vlan-id*]**

vlan-id is the VLAN ID of the port and is an optional value.

To attach a VNF interface to a VLAN:

- Create a VLAN:

  `user@host#` **set vmhost vlan *vlan-name***

- Attach a VNF interface to a VLAN:

`user@host#` **set virtual-network-functions *vnf-name* interfaces *interface-name* mapping vlan members *list-of-vlans* [mode trunk|access]**

NOTE:

- The interfaces attached to a VNF are persistent across VNF restarts.
- If the VNF supports hot-plugging, you can attach the interfaces while the VNF is running. Otherwise, you must add the interfaces, and then restart the VNF.
- You cannot change the mapping of a VNF interface while the VNF is running.

NOTE: You can prevent the VNF interface from sending or receiving traffic by using the **deny-forwarding** CLI option.

If the **deny-forwarding** option is enabled on an interface that is a part of cross-connect, then the cross-connect status goes down and drops all traffic.

set virtual-network-options *vnf-name* interface *interface-name* forwarding-options deny-forwarding

To specify the target PCI address for a VNF interface:

`user@host#` **set virtual-network-functions *vnf-name* interfaces *interface-name* pci-address *target-pci-address***

You can use the target PCI address to rename or reorganize interfaces within the VNF.

For example, a Linux-based VNF can use udev rules within the VNF to name the interface based on the PCI address.

NOTE:

- The target PCI address string should be in the following format:

  0000:00:<slot:>:0, which are the values for domain:bus:slot:function. The value for slot should be different for each VNF interface. The values for domain, bus, and function should be zero.
- You cannot change the target PCI address of VNF interface while the VNF is running.

To delete a VNF interface:

```
user@host# delete virtual-network-functions vnf-name interfaces interface-name
user@host# commit
```

NOTE:

- To delete a VNF interface, you must stop the VNF, delete the interface, and then restart the VNF.
- After attaching or detaching a virtual function, you must restart the VNF for the changes to take effect.
- eth0 and eth1 are reserved for the default VNF interfaces that are connected to the internal network and the out-of-band management network. Therefore, the configurable VNF interface names start from eth2.
- Within a VNF, the interface names can be different, based on guest OS naming conventions. VNF interfaces that are configured in the JCP might not appear in the same order within the VNF.
- You must use the target PCI addresses to map to the VNF interfaces that are configured in the JCP and you must name them accordingly.

Configure Storage Devices for VNFs

An NFX350 device supports the following storage options for VNFs:

- CD-ROM
- Disk
- USB

To add a virtual CD or to update the source file of a virtual CD:

`user@host#` **set virtual-network-functions *vnf-name* storage *device-name* type cdrom source file *file-name***

You can specify a valid device name in the format hd*x*, sd*x*, or vd*x*—for example, hdb, sdc, vdb, and so on.

To add a virtual USB storage device:

`user@host#` **set virtual-network-functions *vnf-name* storage *device-name* type usb source file *file-name***

To attach an additional hard disk:

`user@host#` **set virtual-network-functions *vnf-name* storage *device-name* type disk [bus-type virtio | ide] [file-type raw | qcow2] source file *file-name***

To delete a virtual CD, USB storage device, or hard disk from the VNF:

`user@host#` **delete virtual-network-functions *vnf-name* storage *device-name***

NOTE:

- After attaching or detaching a CD from a VNF, you must restart the device for the changes to take effect. The CD detach operation fails if the device is in use within the VNF.
- A VNF supports one virtual CD, one virtual USB storage device, and multiple virtual hard disks.
- You can update the source file in a CD or USB storage device while the VNF is running.
- You must save the source file in the **/var/public** directory, and the file must have read and write permission for all users.

Instantiate a VNF

You can instantiate a VNF by configuring the VNF name, and by specifying the path of an image.

While instantiating a VNF with an image, two VNF interfaces are added by default. These interfaces are required for management and for the internal network.

> **NOTE:** Only QCOW2, IMG, and RAW image types are supported.

To instantiate a VNF by using an image:

```
user@host# set virtual-network-functions vnf-name image file-path
user@host# set virtual-network-functions vnf-name image image-type image-type
user@host# commit
```

> **NOTE:** When you configure VNFs, do not use VNF names in the format vnf*n*—for example, vnf1, vnf2, and so on. Configurations that contain such names fail to commit.

(Optional) To specify a UUID for the VNF:

```
user@host# set virtual-network-functions vnf-name [uuid vnf-uuid]
```

uuid is an optional parameter. We recommend that you allow the system to allocate a UUID for the VNF.

> **NOTE:** You cannot change the image configuration for a VNF after saving and committing the configuration. To change the image for a VNF, you must delete the VNF and create a VNF again.

Verify the VNF Instantiation

To verify that the VNF is instantiated successfully:

```
user@host> show virtual-network-functions
```

```
ID        Name                                                  State      Liveliness
------------------------------------------------------------------------------------
1         vjunos0                                               Running    alive
2         centos1                                               Running    alive
3         centos2                                               Running    alive
```

The output in the **Liveliness** field of a VNF indicates whether the IP address of the VNF is reachable over the internal management network. The default IP address of the liveliness bridge is 192.0.2.1/24. Note that this IP address is internal to the device and is used for VNF management.

Managing VNFs on NFX350 Devices

IN THIS SECTION

Managing VNF States

By default, a VNF automatically starts when the VNF configuration is committed.

- To disable autostart of a VNF when the VNF configuration is committed:

  `user@host#` **set virtual-network-functions *vnf-name* no-autostart**

- To manually start a VNF:

  `user@host>` **request virtual-network-functions *vnf-name* start**

- To stop a VNF:

  `user@host>` **request virtual-network-functions *vnf-name* stop**

- To restart a VNF:

  `user@host>` **request virtual-network-functions *vnf-name* restart**

- To access the console of an active VNF:

  `user@host>` **request virtual-network-functions *vnf-name* console**

  > **NOTE:** The **request virtual-network-functions *vnf-name* console** command is supported only for root login over ssh.

- To access a VNF through SSH:

  `user@host>` **request virtual-network-functions ssh *vnf-name***

- To access a VNF through Telnet:

  `user@host>` **request virtual-network-functions telnet *vnf-name***

Managing VNF MAC Addresses

VNF interfaces that are defined, either using the CLI, are assigned a globally unique and persistent MAC address. A common pool of 176 MAC addresses is used to assign MAC addresses to VNF interfaces. These MAC addresses are automatically allocated when a VNF is instantiated. You can configure a MAC address other than what is available in the common pool, and this address will not be overwritten.

- To configure a specific MAC address for a VNF interface:

  `user@host#` **set virtual-network-functions *vnf-name* interfaces *interface-name* mac-address *mac-address***

- To delete the MAC address configuration of a VNF interface:

  `user@host#` **delete virtual-network-functions *vnf-name* interfaces *interface-name* mac-address *mac-address***

NOTE:

- To delete or modify the MAC address of a VNF interface, you must stop the VNF, make the necessary changes, and then restart the VNF.
- The MAC address specified for a VNF interface can be either a system MAC address or a user-defined MAC address.
- The MAC address specified from the system MAC address pool must be unique for the VNF interfaces.

Managing the MTU of a VNF Interface

The maximum transmission unit (MTU) is the largest data unit that can be forwarded without fragmentation. You can configure either 1500 bytes or 9216 bytes as the MTU size. The default MTU value is 1500 bytes, and the maximum MTU size for both VNF and L3 interface is 9216 bytes.

NOTE: MTU configuration is supported only on VLAN interfaces.

1. To configure the MTU on a VNF interface:

   `user@host# set` **virtual-network-functions *vnf-name* interfaces *interface-name* mtu *size***

   NOTE: You must restart the VNF after configuring the MTU, if the VNF does not support hot-plugging functionality.

2. To delete the MTU of a VNF interface:

   `user@host#` **delete virtual-network-functions *vnf-name* interfaces *interface-name* mtu**

   NOTE: After the MTU is deleted, the MTU of the VNF interface is reset to 1500 bytes.

NOTE:

- The maximum number of VLAN interfaces on the OVS that are supported in the system is 64.

Accessing a VNF from the JCP

You can access a VNF from the JCP through SSH or by using the console.

To access a VNF from the JCP through SSH:

user@host> request virtual-network-functions ssh *vnf-name*

To access a VNF from the JCP by using the console:

user@host> request virtual-network-functions console *vnf-name*

Viewing the List of VNFs

To view the list of VNFs:

`user@host>` **show virtual-network-functions**

```
ID        Name                                                  State      Liveliness
------------------------------------------------------------------------------------
1         vjunos0                                               Running    alive
2         centos1                                               Running    alive
3         centos2                                               Running    alive
```

The **Liveliness** field of a VNF indicates whether the IP address of the VNF is reachable from the JCP. The default IP address of the liveliness bridge is 192.0.2.1/24.

Displaying the Details of a VNF

To display the details of a VNF:

`user@host>` **show virtual-network-functions *vnf-name* detail**

```
user@host>show virtual-network-functions centos1 detail
Virtual Network Function Information
------------------------------------

Id:                 2
Name:               centos1
State:              Running
Liveliness:         Up
IP Address:         192.0.2.101
VCPUs:              1
Maximum Memory:     1048576 KiB
Used Memory:        1048576 KiB
Used 1G Hugepages:  0
Used 2M Hugepages:  0
Error:              None
```

Deleting a VNF

To delete a VNF:

`user@host#` **delete virtual-network-functions *vnf-name***

> **NOTE:** The VNF image remains in the disk even after you delete a VNF.

Configuring Analyzer VNF and Port-mirroring

The **Port-mirroring** feature allows you to monitor network traffic. If the feature is enabled on a VNF interface, the OVS system bridge sends a copy of all network packets of that VNF interface to the analyzer VNF for analysis. You can use the port-mirroring or analyzer commands for analyzing the network traffic.

NOTE:

- Port-mirroring is supported only on VNF interfaces that are connected to an OVS system bridge.
- VNF interfaces must be configured before configuring port-mirroring options.
- If the analyzer VNF is active after you configure, you must restart the VNF for changes to take effect.
- You can configure up to four input ports and only one output port for an analyzer rule.
- Output ports must be unique in all analyzer rules.
- After changing the configuration of the input VNF interfaces, you must de-activate and activate the analyzer rules referencing to it along with the analyzer VNF restart.

To configure the analyzer VNF and enable port-mirroring:

1. Configure the analyzer VNF:

```
[edit]
user@host#set virtual-network-functions analyzer-vnf-name image file-path
user@host#set virtual-network-functions analyzer-vnf-name interfaces interface-name analyzer
```

2. Enable port-mirroring of the network traffic in the input and output ports of the VNF interface and analyzer VNF:

```
user@host# set vmhost forwarding-options analyzer analyzer-instance-name input [ingress | egress]
   virtual-network-function vnf-name interface interface-name
user@host# set vmhost forwarding-options analyzer analyzer-rule-name output virtual-network-function
   analyzer-vnf-name interface interface-name
```

8

CHAPTER

Configuring Mapping of Address and Port with Encapsulation (MAP-E)

Mapping of Address and Port with Encapsulation on NFX Series Devices

IN THIS SECTION

Overview

Mapping of Address and Port with Encapsulation (MAP-E) is an IPv6 transition technique that encapsulates an IPv4 packet in an IPv6 address and carries it over an IPv4-over-IPv6 tunnel from MAP-E customer edge (CE) devices to MAP-E provider edge (PE) devices (also called as border relay [BR] devices) through an IPv6 routing topology, where the packets are detunneled for further processing.

MAP-E uses Network Address Port Translation (NAPT) features for restricting transport protocol ports, Internet Control Message Protocol (ICMP) identifiers, and fragment identifiers to the configured port sets. The existing NAPT features are enhanced to add MAP-E capability.

Benefits of MAP-E

In most cases, during IPv4 to IPv6 migration, only the IPv6 network is available. However, an IPv4 network is required for all residual IPv4 deployment. In scenarios where service providers have an IPv6 network and the LAN subscribers are not IPv6-capable, MAP-E supports IPv4 to IPv6 migration and deployment. MAP-E transports IPv4 packets across an IPv6 network using IP encapsulation. Encapsulation is done based on the mapping of IPv6 addresses to IPv4 addresses and to transport layer ports. Typically, during IPv6 transition, service providers might have a limited pool of public IPv4 addresses. MAP-E enables the sharing of public IPv4 addresses among multiple CE devices.

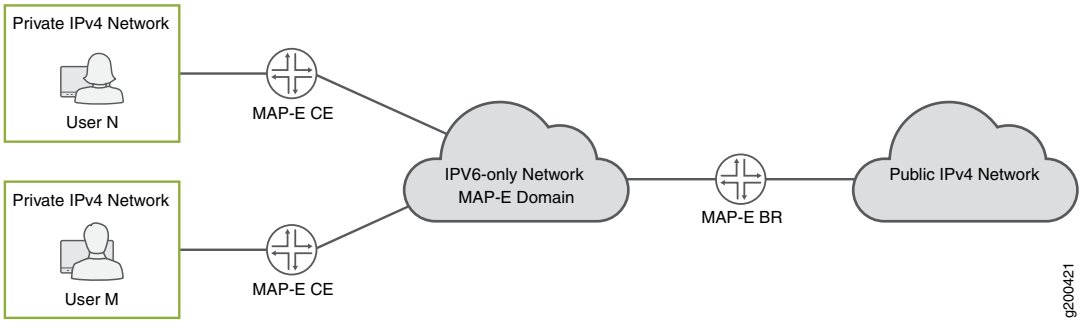
MAP-E Terminology

Terminology	Description
Border relay (BR)	The MAP-E-enabled provider edge device in a MAP domain. A BR device has at least one IPv6-enabled interface and one IPv4 interface connected to the native IPv4 network.
Embedded address (EA) bits	The EA bits in the IPv6 address identify an IPv4 prefix, IPv4 address, or a shared IPv4 address and a PSID.
MAP domain	One or more MAP-E customer edge devices and BR devices connected to the same virtual link.
MAP rule	A set of parameters that describe the mapping of an IPv4 prefix, IPv4 address, or a shared IPv4 address with an IPv6 prefix or IPv6 address. Each domain uses a different mapping rule set. Every MAP node must be provisioned with a basic mapping rule, which is used by the node to configure its IPv4 address, IPv4 prefix, or shared IPv4 address. The basic mapping rule is a forwarding mapping rule that is used for forwarding, where an IPv4 destination address and optionally a destination port is mapped to an IPv6 address.
MAP-E Customer Edge (CE)	The MAP-E-enabled customer edge device in a MAP deployment.
Port set ID (PSID)	Separate part of the transport layer port space that is denoted as the port set ID.
Softwire	Tunnel between two IPv6 endpoints to carry IPv4 packets or between two IPv4 endpoints to carry IPv6 packets.

MAP-E Functionality

Figure 8 on page 105 illustrates a simple MAP-E deployment scenario.

Figure 8: MAP-E Deployment



In a MAP-E network topology, there are two MAP-E CE devices, each connected to a private IPv4 host. The MAP-E CE devices are dual stack and are capable of NAPT. The MAP-E CE devices connect to a MAP-E BR device through an IPv6-only MAP-E network domain. The MAP-E BR device is dual stack and is connected to both a public IPv4 network and an IPv6 MAP-E network.

The MAP-E functionality is as follows:

1. The MAP-E CE devices are capable of NAPT. On receiving an IPv4 packet from the host, the MAP-E CE device performs NAT on the incoming IPv4 packets.
2. After NAT is performed, the IPv4 packets are then encapsulated into IPv6 packets by the MAP-E CE device, and are sent to the MAP-E BR device.
3. The IPv6 packets are transported through the IPv6-only service provider network and reach the MAP-E BR device.
4. The incoming IPv6 packets are decapsulated by the MAP-E BR and are routed to the IPv4 public network.

In the reverse path, the incoming IPv4 packets are encapsulated into IPv6 packets by the MAP-E BR device, and are routed to the MAP-E CE devices.

Configuring MAP-E on NFX Series Devices

IN THIS SECTION

Overview

This example describes how to configure Mapping of Address and Port with Encapsulation (MAP-E) functionality on NFX Series devices. For more information about MAP-E, see "Mapping of Address and Port with Encapsulation on NFX Series Devices" on page 103.

Requirements

This example uses the following hardware and software components:

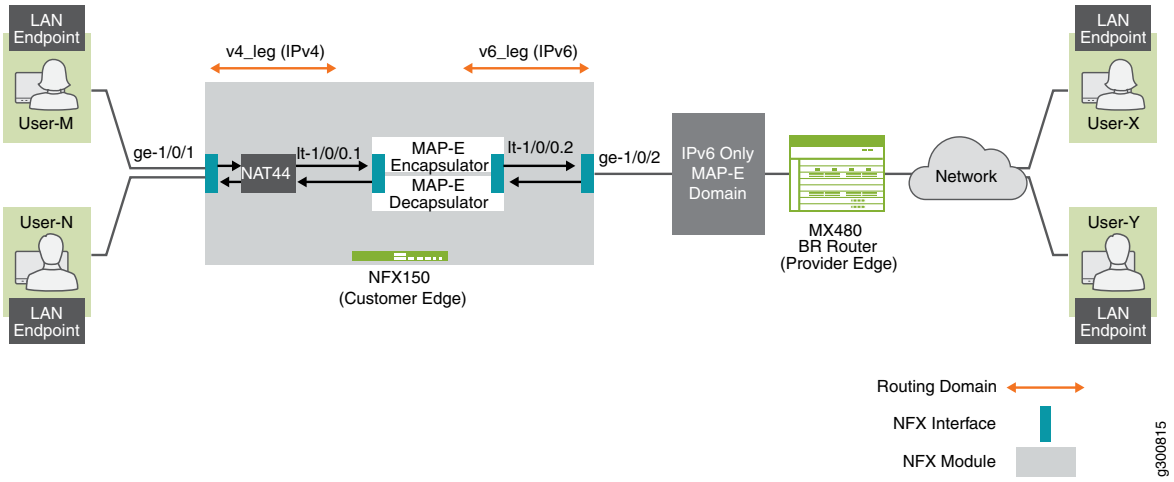
- NFX150 device running Junos OS Release 19.4R1, deployed as a customer edge (CE) device.
- MX480 device, deployed as a border relay (BR) device.
- Map physical interfaces to virtual interfaces. For more information, see Mapping Interfaces on NFX150 Devices.

Topology Overview

This topology shows how to configure MAP-E CE functionality on NFX Series devices. This topology also shows how the IPv4 packets from MAP-E CE devices are encapsulated and transported through an IPv4-over-IPv6 tunnel to MAP-E provider edge (PE) devices (also known as border relay [BR] devices) through an IPv6 routing topology, where the packets are detunneled for further processing. An MX Series device is used as the MAP-E BR device, which is a dual-stack device connected to both a public IPv4 network and an IPv6 MAP-E network.

Figure 9 on page 107 shows the MAP-E deployment on NFX Series devices.

Figure 9: MAP-E Deployment on NFX Series Device



Configure an NFX Series Device as a MAP-E CE Device

To configure an NFX Series device as a MAP-E customer edge device:

1. Configure the security policies and zones for applying different security measures on IPv4-facing interfaces and IPv6-facing interfaces. The following configuration adds LAN interface (ge-1/0/1) and WAN interface on the service provider end (ge-1/0/2) into relevant security zones and configures a policy to permit all traffic between these zones. The configuration also adds corresponding internal logical tunnel (lt) interface units into security zones.

```
user@host# set security policies global policy my_ce match source-address any
user@host# set security policies global policy my_ce match destination-address any
user@host# set security policies global policy my_ce match application any
user@host# set security policies global policy my_ce then permit
user@host# set security policies default-policy permit-all
user@host# set security zones security-zone v4zone host-inbound-traffic system-services all
user@host# set security zones security-zone v4zone host-inbound-traffic protocols all
user@host# set security zones security-zone v4zone interfaces ge-1/0/1.0
user@host# set security zones security-zone v4zone interfaces lt-1/0/0.1
user@host# set security zones security-zone v6zone host-inbound-traffic system-services all
user@host# set security zones security-zone v6zone host-inbound-traffic protocols all
user@host# set security zones security-zone v6zone interfaces ge-1/0/2.0
user@host# set security zones security-zone v6zone interfaces lt-1/0/0.2
```

2. Configure the interfaces to provide network connectivity and data flow. The following configuration assigns IPv4 address on LAN side and IPv6 on WAN side. The MTU on the IPv6 side must support maximum MTU.

```
user@host# set interfaces ge-1/0/1 unit 0 family inet address 10.10.10.1/24
user@host# set interfaces ge-1/0/2 mtu 9192
user@host# set interfaces ge-1/0/2 unit 0 family inet6 address 2001:db8:ffff::1/64
```

3. Configure both the logical tunnel interfaces. The logical tunnel interfaces act as internal endpoints to MAP-E encapsulator or decapsulator block in NFX series box. This separates the network traffic for IPv4 and IPv6. Here, lt-1/0/0 unit 1 terminates IPv4 traffic that is received on ge-1/0/1 and lt-1/0/0 unit 2 initiates IPv6 traffic to be sent out through ge-1/0/2. lt-1/0/0 unit 2 terminates IPv6 traffic that is received on ge-1/0/2 and lt-1/0/0 unit 1 initiates IPv4 traffuc to be sent out through ge-1/0/1.

```
user@host# set interfaces lt-1/0/0 mtu 9192
user@host# set interfaces lt-1/0/0 unit 1 encapsulation ethernet
user@host# set interfaces lt-1/0/0 unit 1 peer-unit 2
user@host# set interfaces lt-1/0/0 unit 1 family inet address 172.16.100.1/24
user@host# set interfaces lt-1/0/0 unit 1 family inet6 address 2001:db8:fffe::1/64

user@host# set interfaces lt-1/0/0 unit 2 encapsulation ethernet
```

```
user@host# set interfaces lt-1/0/0 unit 2 peer-unit 1
user@host# set interfaces lt-1/0/0 unit 2 family inet address 172.16.100.2/24
user@host# set interfaces lt-1/0/0 unit 2 family inet6 address 2001:db8:fffe::2/64
```

4. Configure the routing instances for the IPv4 and IPv6 network traffic domains inside NFX:

```
user@host# set routing-instances v4_leg routing-options rib v4_leg.inet.0 static route
198.51.100.0/24 next-hop 172.16.100.2
user@host# set routing-instances v4_leg routing-options rib v4_leg.inet.0 static route 203.0.113.0/24
next-hop 172.16.100.2
user@host# set routing-instances v4_leg routing-options rib v4_leg.inet.0 static route 192.0.2.0/24
next-hop 172.16.100.2
user@host# set routing-instances v4_leg instance-type virtual-router
user@host# set routing-instances v4_leg interface lt-1/0/0.1

user@host# set routing-instances v4_leg interface ge-1/0/1.0
user@host# set routing-instances v6_leg routing-options rib v6_leg.inet.0 static route 10.10.10.0/24
next-hop 172.16.100.1
user@host# set routing-instances v6_leg routing-options rib v6_leg.inet6.0 static route
2001:db8::a/128 next-hop 2001:db8:ffff::9
user@host# set routing-instances v6_leg routing-options rib v6_leg.inet6.0 static route
2001:db8:0012:3500::/56 next-hop 2001:db8:ffff::2
user@host# set routing-instances v6_leg routing-options rib v6_leg.inet6.0 static route
2001:db8:0012:3400::/56 next-hop 2001:db8:fffe::1
user@host# set routing-instances v6_leg instance-type virtual-router
user@host# set routing-instances v6_leg interface lt-1/0/0.2
user@host# set routing-instances v6_leg interface ge-1/0/2.0
```

5. Configure the MAP-E BMR and FMR rules to provide mapping between the IPv4 network and IPv6 network:

```
user@host# set security softwires map-e mapce1 br-address 2001:db8::a/128
user@host# set security softwires map-e mapce1 end-user-prefix 2001:db8:0012:3400::/56
user@host# set security softwires map-e mapce1 rule bmr rule-type BMR
user@host# set security softwires map-e mapce1 rule bmr ipv4-prefix 192.0.2.0/24
user@host# set security softwires map-e mapce1 rule bmr ipv6-prefix 2001:db8::/40
user@host# set security softwires map-e mapce1 rule bmr ea-bits-length 16
user@host# set security softwires map-e mapce1 rule bmr psid-offset 6
user@host# set security softwires map-e mapce1 role CE
```

```
user@host# set security softwires map-e mapce1 version 3
```

6. Configure source NAT rule and NAT pool:

```
user@host# set security nat source pool my_mape allocation-domain mapce1
user@host# set security nat source pool my_mape allocation-domain allocation-rule bmr
user@host# set security nat source rule-set mape from zone v4zone
user@host# set security nat source rule-set mape to interface lt-1/0/0.1
user@host# set security nat source rule-set mape to interface ge-1/0/1.0
user@host# set security nat source rule-set mape rule r1 match source-address 10.10.10.0/24
user@host# set security nat source rule-set mape rule r1 match destination-address 10.10.10.0/24
user@host# set security nat source rule-set mape rule r1 match destination-address 198.51.100.0/24
user@host# set security nat source rule-set mape rule r1 match destination-address 203.0.113.0/24
user@host# set security nat source rule-set mape rule r1 match destination-address 192.0.2.0/24
user@host# set security nat source rule-set mape rule r1 then source-nat pool my_mape
user@host# set security nat source rule-set mape rule r1 then source-nat pool persistent-nat permit
any-remote-host
```

7. Commit the configuration:

```
user@host# commit
```

Configure an MX Series Device as a BR Device

To configure an MX Series device as a border relay device:

1. Configure the service set for MAP-E on the MX Series device:

```
user@host# set services service-set ss1 softwire-rules sw-rule1
user@host# set services service-set ss1 next-hop-service inside-service-interface si-1/0/0.1
user@host# set services service-set ss1 next-hop-service outside-service-interface si-1/0/0.2
```

2. Configure the MAP-E softwire concentrator and associated parameters. This creates a tunnel between two IPv6 endpoints to carry IPv4 packets or between two IPv4 endpoints to carry IPv6 packets.

```
user@host# set services softwire softwire-concentrator map-e mape-domain-1 softwire-address
2001:db8::a
user@host# set services softwire softwire-concentrator map-e mape-domain-1 ipv4-prefix
192.0.2.0/24
```

```
user@host# set services softwire softwire-concentrator map-e mape-domain-1 mape-prefix
2001:db8::/40
user@host# set services softwire softwire-concentrator map-e mape-domain-1 ea-bits-len 16
user@host# set services softwire softwire-concentrator map-e mape-domain-1 psid-offset 6
user@host# set services softwire softwire-concentrator map-e mape-domain-1 psid-length 8
user@host# set services softwire softwire-concentrator map-e mape-domain-1 mtu-v6 9192
user@host# set services softwire softwire-concentrator map-e mape-domain-1 version-03
user@host# set services softwire softwire-concentrator map-e mape-domain-1 v4-reassembly
user@host# set services softwire softwire-concentrator map-e mape-domain-1 v6-reassembly
user@host# set services softwire softwire-concentrator map-e mape-domain-1 disable-auto-route
```

3. Configure a softwire rule to specify the direction of traffic to be tunneled and the MAP-E softwire concentrator to be used:

```
user@host# set services softwire rule sw-rule1 match-direction input
user@host# set services softwire rule sw-rule1 term t1 then map-e mape-domain-1
```

4. Configure a service interface inside the dual-stack domain:

```
user@host# set interfaces si-1/0/0 unit 1 family inet6
user@host# set interfaces si-1/0/0 unit 1 service-domain inside
```

5. Configure a service interface outside the dual-stack domain:

```
user@host# set interfaces si-1/0/0 unit 2 family inet
user@host# set interfaces si-1/0/0 unit 2 service-domain outside
```

6. Configure the maximum transmission unit (MTU) on the BR interface:

```
user@host# set interfaces ge-1/1/2 mtu 9192
```

7. Configure the logical interfaces and assign the IPv4 and IPv6 addresses:

```
user@host# set interfaces ge-1/1/2 unit 0 family inet6 address 2001:db8:ffff::9/64
user@host# set interfaces ge-1/1/3 unit 0 family inet address 203.0.113.1/24
```

8. Configure the routing instances:

```
user@host# set routing-options rib inet6.0 static route 2001:db8::/40 next-hop si-1/0/0.1
user@host# set routing-options rib inet6.0 static route 2001:db8:0012:3400::/56 next-hop
2001:db8:ffff::1
```

```
user@host# set routing-options rib inet6.0 static route 2001:db8:0012:3500::/56 next-hop
2001:db8:ffff::2
user@host# set routing-options static route 192.0.2.0/24 next-hop si-1/0/0.2
user@host# set routing-options static route 198.51.100.0/24 next-hop si-1/0/0.2
user@host# set routing-options static route 203.0.113.0/24 next-hop si-1/0/0.2
```

9. Commit the configuration:

```
user@host# commit
```

Verify the MAP-E Configuration

Purpose

After completing the MAP-E configuration on an NFX Series device, you can verify the status of the MAP-E configuration.

Action

- Verify the status of the packet flow:

```
user@host> show security flow session
```

```
Session ID: 134218806, Policy name: my_ce/4, Timeout: 1800, Valid
  In: 10.10.10.2/57630 --> 203.0.113.2/22;tcp, Conn Tag: 0x0, If: ge-1/0/1.0,
Pkts: 50, Bytes: 5797,
  Out: 203.0.113.2/22 --> 192.0.2.18/20691;tcp, Conn Tag: 0x0, If: lt-1/0/0.1,
Pkts: 33, Bytes: 5697,

Session ID: 134218807, Policy name: my_ce/4, Timeout: 1800, Valid
  In: 2001:db8:12:3400:c0:2:1200:3400/1 --> 2001:db8::a/1;ipip, Conn Tag: 0x0,
If: lt-1/0/0.2, Pkts: 50, Bytes: 7797,
  Out: 2001:db8::a/1 --> 2001:db8:12:3400:c0:2:1200:3400/1;ipip, Conn Tag: 0x0,
 If: ge-1/0/2.0, Pkts: 33, Bytes: 7017,
Total sessions: 2
```

- Verify whether the IPv4 and IPv6 addresses are configured correctly:

```
user@host> show security softwires map-e domain mapce1
```

```
Role                  : CE
Version               : 3
Domain Name           : mapce1
```

```
BR Address            : 2001:db8::a/128
End User Ipv6 prefix  : 2001:db8:12:3400::/56
BMR Mapping Rule :
    Rule Name            : bmr
    Rule Ipv4 Prefix     : 192.0.2.0/24
    Rule Ipv6 Prefix     : 2001:db8::/40
    PSID offset          : 6
    PSID length          : 8
    EA bit length        : 16
    Port SetID           : 0x34
    MAP-E Ipv4 address   : 192.0.2.18/32
    MAP-E Ipv6 address   : 2001:db8:12:3400:c0:2:1200:3400
```

- Verify the map rule statistics:

  `user@host>` **show security softwires map-e domain mapce1 statistics rule bmr**

```
BMR Rule Name             :bmr
Encapsulated packets       :289
Decapsulated packets       :269
Encapsulation errors       :0
Decapsulation errors       :0
Encapsulated fragmentation :0
Decapsulated fragmentation :0
Invalid port set           :0
IPv6 address mismatch      :0
```

- View the details of the NAT source rule:

  `user@host>` **show security nat source rule all**

```
Total rules: 1
Total referenced IPv4/IPv6 ip-prefixes: 5/0
source NAT rule: r1                     Rule-set: mape
  Rule-Id                   : 1
  Rule position             : 1
  From zone                 : v4zone
  To interface              : lt-1/0/0.1
                            : ge-1/0/1.0
  Match
    Source addresses        : 10.10.10.0      - 10.10.10.255
    Destination addresses   : 10.10.10.0      - 10.10.10.255
                              198.51.100.0    - 198.51.100.255
```

```
                              203.0.113.0     - 203.0.113.255
                              192.0.2.0       - 192.0.2.255
  Action                       : my_mape
    Persistent NAT type        : any-remote-host
    Persistent NAT mapping type : address-port-mapping
    Inactivity timeout         : 300
    Max session number         : 30
  Translation hits           : 1
    Successful sessions      : 1
    Failed sessions          : 0
  Number of sessions         : 1
```

- View the details of the NAT source pool:

  `user@host>` **show security nat source pool all**

```
Total pools: 1
Pool name          : my_mape
Pool id            : 4
Routing instance   : default
Host address base  : 0.0.0.0
Map-e domain name  : mapce1
Map-e rule name    : bmr
PSID offset        : 6
PSID length        : 8
PSID               : 0x34
Port overloading   : 1
Address assignment : no-paired
Total addresses    : 1
Translation hits   : 1
Address range                        Single Ports   Twin Ports
         192.0.2.18 - 192.0.2.18         1              0
Total used ports   :                       1              0
```

- View the NAT source summary:

  `user@host>` **show security nat source summary**

```
show security nat source summary
Total port number usage for port translation pool: 252
Maximum port number for port translation pool: 33554432
Total pools: 1
Pool                Address                 Routing             PAT  Total
Name                Range                   Instance                 Address
```

```
my_mape              192.0.2.18-192.0.2.18    default              yes  1

Total rules: 1
Rule name          Rule set       From              To                   Action
r1                 mape           v4zone            lt-1/0/0.1           my_mape
r1                                                  ge-1/0/1.0
```

- View the persistent NAT table:

  `user@host>` **show security nat source persistent-nat-table all**

```
Internal                        Reflective                  Source     Type
         Left_time/  Curr_Sess_Num/  Source
In_IP          In_Port I_Proto Ref_IP          Ref_Port R_Proto NAT Pool
             Conf_time   Max_Sess_Num    NAT Rule
10.10.10.2      57630   tcp    192.0.2.18      20691     tcp      my_mape
any-remote-host     -/300      1/30          r1
```

- View the softwire statistics on the MX Series device:

  `user@host>` **show services inline softwire statistics mape**

```
Service PIC Name                                    si-1/0/0

Control Plane Statistics
     MAPE ICMPv6 echo requests to softwire concentrator          0
     MAPE ICMPv6 echo responses from softwire concentrator       0
     MAPE Dropped ICMPv6 packets to softwire concentrator        0

Data Plane Statistics (v6-to-v4)      Packets                  Bytes
     MAPE decaps                        15034                    1388760
     MAPE ICMP decap errors             0                        0
     MAPE decap spoof errors            0                        0
     MAPE v6 reassembled                0                        0
     MAPE dropped v6 fragments          0                        0
     MAPE v6 unsupp protocol drops      0                        0

Data Plane Statistics (v4-to-v6)      Packets                  Bytes
     MAPE encaps                        149544                   223527457
     MAPE ICMP encap errors             0                        0
     MAPE v6 mtu errors                 0                        0
     MAPE v4 reassembled                0                        0
     MAPE dropped v4 fragments          0                        0
```

Meaning

This section describes the output fields for the MAP-E configuration on NFX Series devices.

Role—MAP-E is deployed on a CE device. Currently, only the CE role is supported.

Version—MAP-E version: MAP-E draft-3.

BR address—Border router address to be used as the destination address in the absence of a matching FMR rule.

Rule name—Name of the BMR or FMR rule configured.

Rule IPv4 prefix—IPv4 prefix in the BMR or FMR rule.

Rule IPv6 prefix—IPv6 prefix in the BMR or FMR rule.

Port set ID—Port set identifier, used to algorithmically identify a set of ports exclusively assigned to a CE device.

PSID offset—Port set identifier offset, used to specify the range of excluded ports.

PSID length—Port set identifier length, used to specify the sharing ratio.

EA bit length—Embedded address bit length, used to specify part of the IPv4 address or the PSID.

CHAPTER 9

Configuring Cross-Connect

Configuring Cross-Connect on NFX Series Devices

You can configure cross-connect feature on NFX Series devices when there is a requirement to switch the traffic from one interface to the other interface without using the MAC-based forwarding rule. The cross-connect feature allows you to connect any two NFV backplane interfaces, which can be either two VNF interfaces or one VNF and one NIC interface to switch traffic between the interfaces. You can either switch all traffic or traffic belonging to a particular VLAN, unidirectionally or bidirectionally.

NOTE:

- On NFX150 and NFX250 NextGen devices, virtual interfaces can be hsxe0, hsxe1, or ge-1/0/x connected to NFV backplane.
- On NFX350 devices, virtual interfaces can be hsxe0, hsxe1, hsxe2, hsxe3, or ge-1/0/x connected to NFV backplane.
- On NFX250 devices, virtual interfaces can be hsxe0 or hsxe1.

NOTE: Cross-connect feature is not supported on VNF interfaces that are SR-IOV interfaces.

You can configure the following types of cross-connect on your device:

- Port cross-connect—You can configure port cross-connect between two OVS interfaces. All traffic is switched between two OVS interfaces in the cross-connect configuration.
- VLAN cross-connect—You can specify a VLAN ID in the cross-connect rule to redirect the traffic from a particular VLAN on an OVS interface to a different interface. All VLAN cross-connects have higher precedence over port cross-connects.
- Untagged cross-connect—You can choose to redirect the untagged frames out of a trunk port to a different destination port by specifying the VLAN ID as *none* in the cross-connect CLI. During this process, you can additionally add a VLAN tag by mentioning the VLAN ID in the other entry of the cross connect rule.

Single leg cross-connect feature allows configuration of single entry on either VNF interface or virtual interface and configure other entry at any later point of time. Single leg cross-connect status is down until the other entry is configured and the interface status of both the entries is up. In a single leg cross-connect configuration, traffic flow is not present until the other entry of the cross-connect is configured.

Unidirectional cross-connect feature allows the traffic to be forwarded conditionally or unconditionally in a single direction. Traffic flow in the opposite direction follows the MAC-based forwarding rule.

The cross-connect feature supports the following:

- Unconditional and conditional cross-connect between two interfaces on the NFV backplane.

- VLAN-based traffic forwarding between two interfaces on the NFV backplane support the following functions:
  - Allows to switch traffic based on a VLAN ID.
  - Allows to switch traffic flow from trunk to access interfaces.
  - Allows to switch traffic flow from access to trunk interfaces.
  - Allows to add a VLAN tag to the traffic, remove the VLAN tag from the traffic, and rewrite the existing VLAN tag to a different tag while switching the traffic between the interfaces.

NOTE:
- When two VNF interfaces are part of cross-connect configuration, and if one of the VNF interfaces is disabled, then all traffic from the VNF interface which is up and is part of cross-connect, is dropped.

Example: Configuring Cross-Connect on NFX350 Devices

IN THIS SECTION

This example shows how to configure the cross-connect feature on NFX350 devices.

Requirements

This example uses an NFX350 device running Junos OS Release 19.4R1.

Overview

The cross-connect feature enables traffic switching between any two VNF interfaces. You can bidirectionally switch either all traffic or traffic belonging to a particular VLAN between any two VNF interfaces.

> **NOTE:** This feature does not support unidirectional traffic flow.

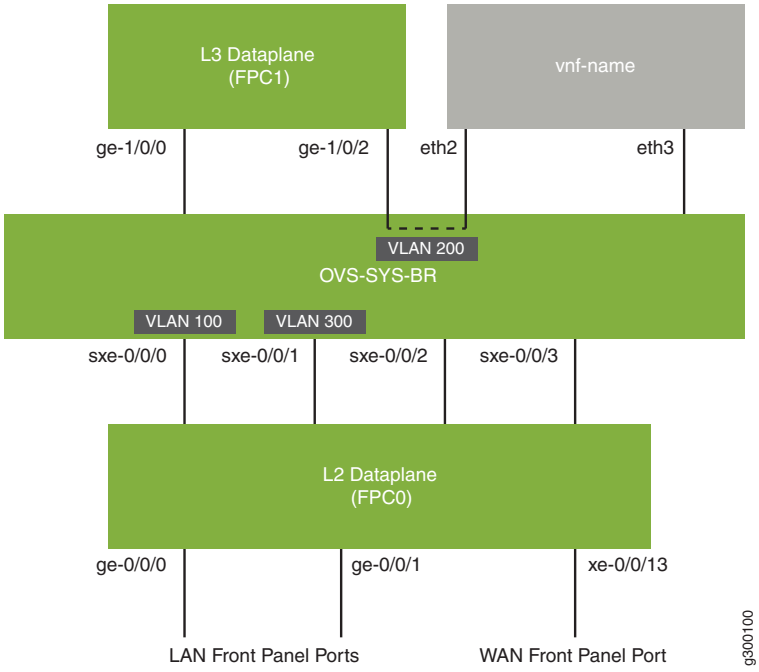
The cross-connect feature supports the following:

- Port cross-connect between two VNF interfaces for all network traffic.
- VLAN-based traffic forwarding between VNF interfaces that support the following functions:
  - Provides an option to switch traffic based on a VLAN ID.
  - Supports VLAN PUSH, POP, and SWAP operations.
  - Supports network traffic flow from trunk to access port through the POP operation.
  - Supports network traffic flow from access to trunk ports through the PUSH operation.

Topology

This example uses the topology shown in Figure 10 on page 122.

Figure 10: Configuring Cross-Connect



Configuration

IN THIS SECTION

Configure VLANs

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

```
user@host# set vlans vlan100 vlan-id 100
```

2. Configure a VLAN for the WAN-side interface.

```
user@host# set vlans vlan300 vlan-id 300
```

Configure the Layer 2 Datapath

Step-by-Step Procedure

1. Configure the LAN-side front panel ports and add them to the LAN-side VLAN.

```
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
```

2. Configure the internal-facing interfaces as trunk ports and add them to the WAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

```
user@host# set interfaces xe-0/0/13 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members vlan300
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching vlan members vlan300
```

Configure the Layer 3 Datapath

Step-by-Step Procedure

1. Configure VLAN tagging on ge-1/0/0:

```
user@host# set interfaces ge-1/0/0 vlan-tagging
user@host# set interfaces ge-1/0/0 unit 0 vlan-id 100
user@host# set interfaces ge-1/0/0 unit 0 family inet address 192.0.2.1/24
```

2. Configure VLAN tagging on ge-1/0/2:

```
user@host# set interfaces ge-1/0/2 vlan-tagging
user@host# set interfaces ge-1/0/2 unit 0 vlan-id 200
```

```
user@host# set interfaces ge-1/0/2 unit 0 family inet address 203.0.113.2/24
```

Configure the VNF

Step-by-Step Procedure

1. Launch the VNF:

```
user@host# set virtual-network-functions vnf-name image /var/public/centos-updated_1.img
user@host# set virtual-network-functions vnf-name image image-type raw
```

2. Specify the number of CPUs required for the VNF:

```
user@host# set virtual-network-functions vnf-name virtual-cpu count 1
```

3. Pin a virtual CPU to a physical CPU:

```
user@host# set virtual-network-functions vnf-name virtual-cpu 0 physical-cpu 2
```

4. Create host VLANs:

```
user@host# set vmhost vlans vlan200 vlan-id 200
user@host# set vmhost vlans vlan300 vlan-id 300
```

5. Configure the VNF interfaces as trunk ports and add them to the LAN-side VLAN:

```
user@host# set virtual-network-functions vnf-name interfaces eth2 mapping vlan mode trunk
user@host# set virtual-network-functions vnf-name interfaces eth2 mapping vlan members vlan200
user@host# set virtual-network-functions vnf-name interfaces eth3 mapping vlan members vlan300
```

6. Specify the memory allocation for the VNF:

```
user@host# set virtual-network-functions vnf-name memory size 1048576
```

Configure Cross-Connect

Step-by-Step Procedure

1. Configure cross-connect:

```
user@host# set vmhost cross-connect c1 virtual-interface ge-1/0/2
user@host# set vmhost cross-connect c1 virtual-network-function vnf-name interface eth2
```

Verify the Configuration

IN THIS SECTION

Verifying the Control Plane Configuration

Purpose

Verify the control plane configuration:

Action

- Verify the VLANs configured.

```
user@host > show vlans
```

```
Routing instance        VLAN name             Tag         Interfaces
default-switch          default               1

default-switch          vlan100               100
                                                          ge-0/0/0.0*
                                                          ge-0/0/1.0*
                                                          sxe-0/0/0.0*
default-switch          vlan200               200
                                                          sxe-0/0/1.0*
                                                          xe-0/0/12.0*
default-switch          vlan300               300
                                                          sxe-0/0/1.0*
                                                          xe-0/0/13.0*
```

- Verify that the VLANs and VLAN memberships are correct by using the **show vmhost vlans** command.

```
user@host> show vmhost vlans
```

```
Routing instance        VLAN name             Tag          Interfaces
vmhost                  vlan200               200
                                                           vnf-name_eth2.0
vmhost                  vlan300               300
                                                           vnf-name_eth3.0
```

- Verify that the VNF is operational. The **State** field shows **Running** for VNFs that are up.

```
user@host> show virtual-network-functions vnf-name
```

```
ID       Name                                                State      Liveliness
------------------------------------------------------------------------------
3        vnf-name                                             Running    alive
```

The **Liveliness** field of the VNF indicates whether the internal management IP address of the VNF is accessible from the Junos Control Plane (JCP).

To view more details of the VNF:

```
user@host> show virtual-network-functions vnf-name detail
```

```
Virtual Network Function Information
------------------------------------

Id:                 3
Name:               vnf-name
State:              Running
Liveliness:         alive
IP Address:         192.0.2.100
VCPUs:              1
Maximum Memory:     1048576 KiB
Used Memory:        1048576 KiB
Used 1G Hugepages:  0
Used 2M Hugepages:  0
Error:              None
```

Verifying the Data Plane Configuration

Purpose

Verify the data plane configuration.

Action

- Verify the status of the Layer 2 (ge-0/0/x) and Layer 3 (ge-1/0/x) interfaces.

  user@host> **show interfaces *interface-name* statistics**

  For example:

  user@host> **show interfaces ge-0/0/0 statistics**

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 149, SNMP ifIndex: 517
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, Duplex: Full-Duplex, BPDU Error: None,
  Loop Detect PDU Error: None, Ethernet-Switching Error: None, MAC-REWRITE Error:
 None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online, IEEE
802.3az Energy Efficient Ethernet: Disabled, Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 30:7c:5e:4c:78:03, Hardware address: 30:7c:5e:4c:78:03
  Last flapped   : 2018-11-26 11:03:32 UTC (04:15:32 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

  Logical interface ge-0/0/0.0 (Index 330) (SNMP ifIndex 519)
    Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
    Input packets : 0
    Output packets: 0
    Protocol eth-switch, MTU: 1514
      Flags: Trunk-Mode
```

  user@host> **show interfaces ge-1/0/2 statistics**

```
Physical interface: ge-1/0/2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 547
  Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Link-mode: Half-duplex,
Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 30:7c:5e:4c:78:1d, Hardware address: 30:7c:5e:4c:78:1d
  Last flapped   : 2018-11-26 11:03:45 UTC (04:19:57 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

  Logical interface ge-1/0/2.0 (Index 334) (SNMP ifIndex 550)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.200 ]  Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Security: Zone: Null
    Protocol inet, MTU: 1500
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 0, Curr new hold
 cnt: 0, NH drop cnt: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 203.0.113/24, Local: 203.0.113.2, Broadcast: 203.0.113.255

  Logical interface ge-1/0/2.32767 (Index 335) (SNMP ifIndex 551)
    Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ]  Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Security: Zone: Null
```

- Verify the status of the OVS interfaces.

  user@host> **show vmhost network nfv-back-plane**

```
Network Name : ovs-sys-br

   Interface : ovs-sys-br
   Type : internal, Link type : Full-Duplex, MAC : 52:86:3c:df:9c:44
   MTU : [], Link State :down, Admin State : down
   IPV4 : None, Netmask : None
   IPV6 : None, IPV6 netmask : None
       Rx-packets :       0
       Rx-drops   :       0
       Rx-errors  :       0
       Tx-packets :       1
       Tx-drops   :       1
       Tx-errors  :       0

   Interface : dpdk0
   Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:e2:b9:08
   MTU : [], Link State :up, Admin State : up
   IPV4 : None, Netmask : None
   IPV6 : None, IPV6 netmask : None
       Rx-packets :       0
       Rx-drops   :       0
       Rx-errors  :       0
       Tx-packets :       1
       Tx-drops   :       0
       Tx-errors  :       0

   Interface : dpdk1
   Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:83:39:72
   MTU : [], Link State :up, Admin State : up
   IPV4 : None, Netmask : None
   IPV6 : None, IPV6 netmask : None
       Rx-packets :       0
       Rx-drops   :       0
       Rx-errors  :       0
       Tx-packets :       0
       Tx-drops   :       0
       Tx-errors  :       0

   Interface : l3_h_ge_1_0_0
   Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
   MTU : [], Link State :up, Admin State : up
```

```
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0

Interface : l3_h_ge_1_0_2
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0

Interface : vnf-name_eth2
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : 1500, Link State :down, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0

Interface : vnf-name_eth3
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : 1500, Link State :down, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0
```

10
CHAPTER

Configuring Service Chaining

Example: Configuring Service Chaining Using VLANs on NFX350 Devices

IN THIS SECTION

This example shows how to configure service chaining using VLANs on the host bridge.

Requirements

This example uses an NFX350 device running Junos OS Release 19.4R1.

Before you configure service chaining, ensure that you have installed and instantiated the relevant virtual network functions (VNFs), assigned the corresponding interfaces, and configured the resources.

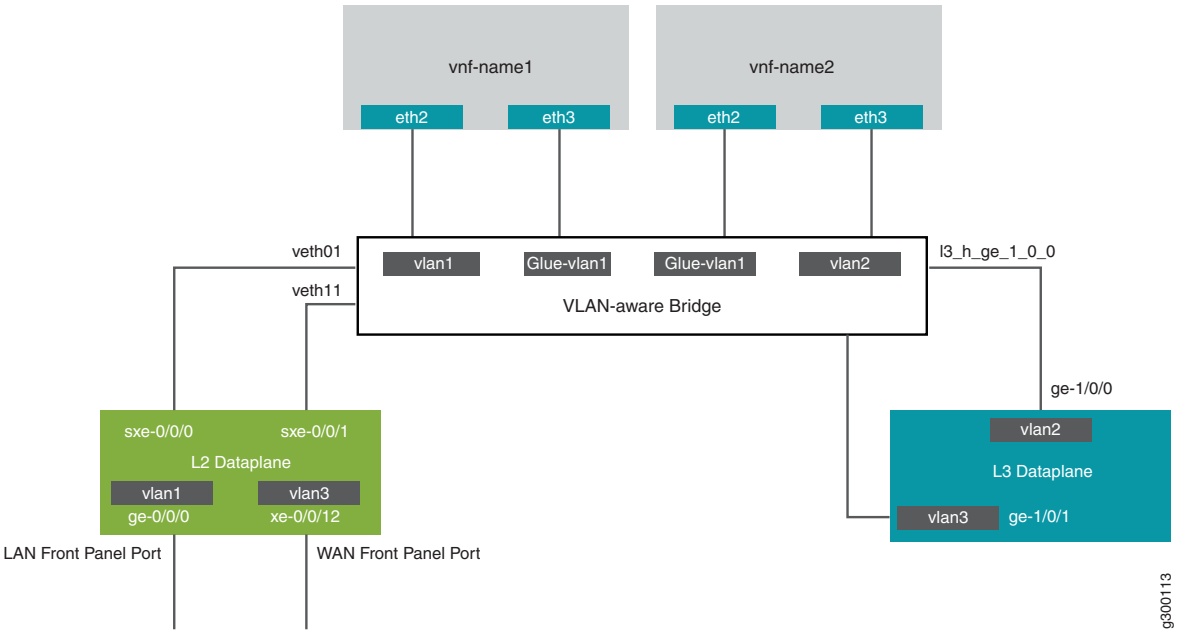
Overview

Service chaining on a device enables multiple services or VNFs on the traffic that flows through the device. This example explains how to configure the various layers of the device to enable traffic to enter the device, flow through two service VNFs, and exit the device.

Topology

This example uses a single NFX350 device running Junos OS, as shown in Figure 11 on page 133.

Figure 11: Configuring Service Chaining Using VLANs



This example is configured using the Junos Control Plane (JCP). The key configuration elements include:

- Front panel ports
- Internal-facing ports
- VNF interfaces, which use the naming format eth# (where # ranges from 0 through 9)
- VLANs to provide bridging between the static interfaces (sxe) and VNF interfaces

Configuration

IN THIS SECTION

Configuring the JCP Interfaces

Step-by-Step Procedure

To configure the interfaces:

1. Connect to the JCP.

```
user@host:~ # cli
user@host>
user@host> configure
[edit]
user@host#
```

2. Map the Layer 3 interface to the Open vSwitch (OVS).

```
user@host# set vmhost virtualization-options interfaces ge-1/0/1
```

3. Configure a VLAN for the LAN-side interfaces.

```
user@host# set vlans vlan1 vlan-id 77
```

4. Configure the LAN-side front panel port and add it to the LAN-side VLAN.

   The LAN-side port is typically an access port, but can be a trunk port if required.

```
user@host# set interfaces ge-0/0/0.0 family ethernet-switching vlan members vlan1
```

5. Configure the LAN-side internal-facing interface as a trunk port and add it to the LAN-side VLAN.

   The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

```
user@host# set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0.0 family ethernet-switching vlan members vlan1
```

6. Configure the WAN-side internal-facing interface as a trunk port and add it to the WAN-side VLAN.

```
user@host# set interfaces sxe-0/0/1.0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/1.0 family ethernet-switching vlan members vlan3
```

7. Configure the WAN-side front panel port and add it to the WAN-side VLAN.

```
user@host# set interfaces xe-0/0/12.0 family ethernet-switching interface-mode access
user@host#  set interfaces xe-0/0/12.0 family ethernet-switching vlan members vlan3
```

8. Configure a VLAN for the WAN-side interface.

```
user@host# set interfaces xe-0/0/12.0 family ethernet-switching interface-mode access
user@host#  set interfaces xe-0/0/12.0 family ethernet-switching vlan members vlan3
```

9. Configure VLAN tagging on the WAN-side front panel port and assign an IP address.

   ```
   user@host# set vlans vlan3 vlan-id 1178
   ```

10. Configure the WAN-side internal-facing interface as a VLAN-tagged interface and assign an IP address to it.

    ```
    user@host# set interfaces ge-1/0/0 vlan-tagging
    user@host# set interfaces ge-1/0/0.0 vlan-id 1177
    user@host#  set interfaces ge-1/0/0.0 family inet address 203.0.113.2/24
    ```

11. Commit the configuration.

    ```
    user@host# commit
    ```

Results

From configuration mode, check the results of your configuration by entering the following **show** commands:

```
[edit]
```

```
user@host# show interfaces ge-0/0/0
```

```
mtu 9192;
unit 0 {
family ethernet-switching {
vlan {
members [ vlan1 ];
}
}
}
```

```
[edit]
```

```
user@host# show interfaces ge-1/0/0
```

```
vlan-tagging;
unit 0 {
    vlan-id 1177;
    family inet {
        address 203.0.113.2/24;
```

```
    }
}
```

[edit]

user@host# **show interfaces ge-1/0/1**

```
vlan-tagging;
unit 0 {
    vlan-id 1178;
    family inet {
          address 192.0.2.1/24;
    }
}
```

[edit]

user@host# **show interfaces sxe-0/0/0**

```
mtu 9192;
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members [ default vlan1 ];
        }
    }
}
```

[edit]

user@host# **show interfaces sxe-0/0/1**

```
mtu 9192;
unit 0 {
family ethernet-switching {
interface-mode trunk;
vlan {
members [ vlan3 ];
}
```

```
}
}
```

```
[edit]
```

```
user@host# show interfaces xe-0/0/12
```

```
mtu 9192;
unit 0 {
family ethernet-switching {
vlan {
members [ vlan3 ];
}
}
}
```

```
[edit]
```

```
user@host# show vlans
```

```
default {
vlan-id 1;
}
vlan1 {
vlan-id 77;
}
Vlan3 {
vlan-id 1178;
}
```

Configuring the VNF Interfaces and Creating the Service Chain

Step-by-Step Procedure

Configure the VNF interfaces.

1. Configure the vmhost instance with the LAN, WAN, or the glue VLANs to be used for service chaining:

   ```
   user@host# set vmhost vlans vlan1 vlan-id 77
   user@host# set vmhost vlans vlan2 vlan-id 1177
   user@host# set vmhost vlans glue-vlan1 vlan-id 123
   ```

2. Instantiate the VNF (vnf-name1) with one virtio interface mapped to the VLAN vlan1 and the other virtio interface mapped to the VLAN glue-vlan1.

   ```
   user@host# set virtual-network-functions vnf-name1 interfaces eth2 mapping vlan members vlan1
   user@host# set virtual-network-functions vnf-name1 interfaces eth3 mapping vlan members
   glue-vlan1
   ```

3. Instantiate the second VNF (vnf-name2) with one interface mapped to the VLAN vlan2 and the second interface mapped to the same glue-vlan1.

   ```
   user@host# set virtual-network-functions vnf-name2 interfaces eth2 mapping vlan members
   glue-vlan1
   user@host# set virtual-network-functions vnf-name2 interfaces eth3 mapping vlan members vlan2
   ```

4. Configure the IP addresses and static routes for each interface of the VNFs as shown in Figure 11 on page 133.

Example: Configuring Service Chaining Using SR-IOV on NFX350 Devices

IN THIS SECTION

This example shows how to configure service chaining using single-root I/O virtualization (SR-IOV). For information about SR-IOV, see *Understanding SR-IOV Usage*.

Requirements

This example uses an NFX350 device running Junos OS Release 19.4R1.

Before you configure service chaining, ensure that you have installed and started the relevant VNFs.

Overview

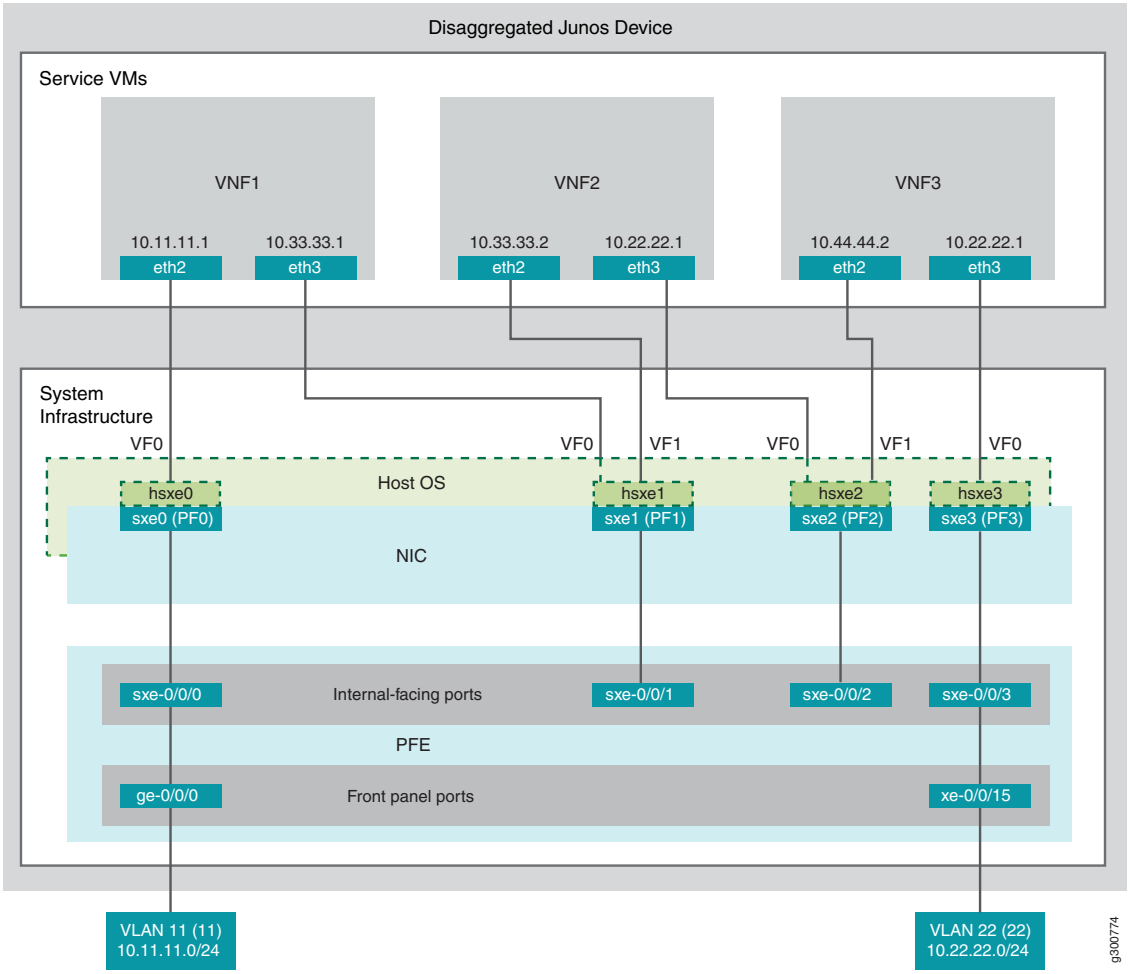
This example uses the front panel ports ge-0/0/0 and xe-0/0/15 associated with the PFE, and its internal-facing ports, sxe-0/0/0 and sxe-0/0/3. The internal NIC ports, sxe0 and sxe3, are not configured directly; instead, they are abstracted at the host OS layer and configured as interfaces hsxe0 and hsxe3. The VNFs use two interfaces, eth2 and eth3. These elements are generally separated into a LAN side and a WAN side. For information on configuring VNFs, see “Configuring VNFs on NFX350 Devices” on page 84.

As this example uses SR-IOV, the virtual functions (VFs) of the NIC ports are used to bypass the host OS and provide direct NIC-to-VM connectivity.

Topology

Figure 12 on page 140 shows the topology for this example.

Figure 12: Service Chaining Using SR-IOV



This example is configured using the Junos Control Plane (JCP). The key configuration elements include:

- Front panel ports associated with the Packet Forwarding Engine
- Internal-facing ports associated with the Packet Forwarding Engine
- NIC ports

> **NOTE:** You must use the host OS interface (hsxe) for these ports because the NIC interfaces (sxe ports) cannot be configured directly.

- VNF interfaces, which use the format eth# (where # ranges from 2 to 9)
- Virtual function settings, which indicate that SR-IOV is being used to provide direct access between the hsxe and VNF interfaces

Configuration

IN THIS SECTION

Configuring the Packet Forwarding Engine Interfaces

CLI Quick Configuration

To quickly configure the Packet Forwarding Engine interfaces, enter the following configuration statements from the JCP:

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

```
set vlans Vlan11 vlan-id 11
set interfaces ge-0/0/0.0 family ethernet-switching vlan member Vlan11
set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
set interfaces sxe-0/0/0.0 family ethernet-switching vlan member Vlan11
set vlans Vlan22 vlan-id 22
set interfaces xe-0/0/15.0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/15.0 family ethernet-switching vlan member Vlan22
set interfaces sxe-0/0/3.0 family ethernet-switching interface-mode trunk
set interfaces sxe-0/0/3.0 family ethernet-switching vlan member Vlan22
```

Step-by-Step Procedure

To configure the Packet Forwarding Engine interfaces:

1. Configure a VLAN for the LAN-side interfaces.

   ```
   user@host# set vlans Vlan11 vlan-id 11
   ```

2. Configure the PFE LAN-side front panel port and add it to the LAN-side VLAN.

   The LAN-side port is typically an access port, but can be a trunk port if required.

   ```
   user@host# set interfaces ge-0/0/0.0 family ethernet-switching vlan members Vlan11
   ```

3. Configure the PFE LAN-side internal-facing interface as a trunk port and add it to the LAN-side VLAN.

   The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

   ```
   user@host# set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
   user@host# set interfaces sxe-0/0/0.0 family ethernet-switching vlan member Vlan11
   ```

4. Configure a VLAN for the WAN-side interfaces.

   ```
   user@host# set vlans Vlan22 vlan-id 22
   ```

5. Configure the PFE WAN-side front panel port as a trunk port and add it to the WAN-side VLAN.

   The WAN-side front panel port is typically a trunk port as it might be required to support multiple VLANs.

   ```
   user@host# set interfaces xe-0/0/15.0 family ethernet-switching interface-mode trunk
   user@host# set interfaces xe-0/0/15.0 family ethernet-switching vlan members Vlan22
   ```

6. Configure the PFE WAN-side internal-facing interface as a trunk port and add it to the WAN-side VLAN.

   ```
   user@host# set interfaces sxe-0/0/3.0 family ethernet-switching interface-mode trunk
   user@host# set interfaces sxe-0/0/3.0 family ethernet-switching vlan members Vlan22
   ```

7. Commit the configuration.

   ```
   user@host# commit
   ```

Results

From configuration mode, check the results of your configuration by entering the following **show** commands:

```
user@host> show interfaces ge-0/0/0
unit 0 {
    family ethernet-switching {
        vlan {
            members Vlan11;
        }
    }
}

user@host> show interfaces xe-0/0/15
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members Vlan22;
        }
    }
}

user@host> show interfaces sxe-0/0/0
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members Vlan11;
        }
    }
}

user@host> show interfaces sxe-0/0/3
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members Vlan22;
        }
    }
}

user@host> show vlans
Vlan11 {
    vlan-id 11;
}
Vlan22 {
```

```
    vlan-id 22;
}
```

Configuring the VNF Interfaces and Creating the Service Chain

Step-by-Step Procedure

To configure the VNF interfaces and create the service chain:

1. Configure VNF1's LAN-side interface as a Layer 3 interface, and map it to the LAN-side NIC interface. Include the virtual function (VF) setting to specify direct NIC-to-VM connectivity. VNFs must use the interfaces from eth2 through eth9.

   The hsxe interface is the configurable representation of the related NIC (sxe) interface.

   ```
   user@host> configure
   [edit]
   user@host# set virtual-network-functions vm1 interfaces eth2 mapping hsxe0 virtual-function
   ```

2. Configure VNF1's WAN-side interface from sxe1.

   ```
   user@host# set virtual-network-functions vm1 interfaces eth3 mapping hsxe1 virtual-function
   ```

3. Instantiate VNF2 with the interfaces eth2 on sxe1 and eth3 on sxe2.

   ```
   user@host# set virtual-network-functions vm2 interfaces eth2 mapping hsxe1 virtual-function
   user@host# set virtual-network-functions vm2 interfaces eth3 mapping hsxe2 virtual-function
   ```

4. Instantiate VNF3 with the interfaces eth2 on sxe2 and eth3 on sxe3.

   ```
   user@host# set virtual-network-functions vm2 interfaces eth2 mapping hsxe2 virtual-function
   user@host# set virtual-network-functions vm2 interfaces eth3 mapping hsxe3 virtual-function
   ```

5. Configure the IP addresses and static routes for each interface of the VNFs, and add routes to achieve a complete bidirectional path for the service chain.

RELATED DOCUMENTATION

- *Understanding Service Chaining on Disaggregated Junos OS Platforms*
- *Disaggregated Junos OS VMs*
- *Understanding SR-IOV Usage*

Example: Configuring Service Chaining Using a Custom Bridge on NFX350 Devices

IN THIS SECTION

This example shows how to configure service chaining using a custom bridge.

Requirements

This example uses an NFX350 device running Junos OS Release 19.4R1.

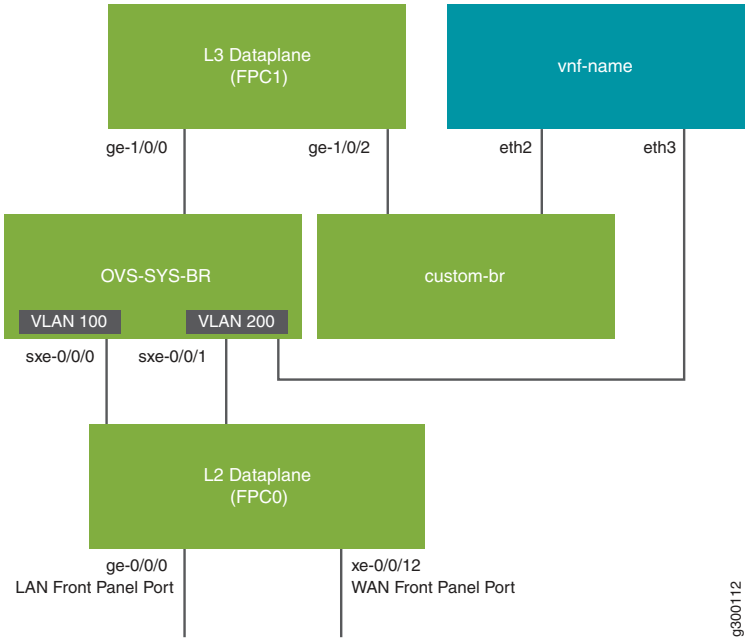
Overview

The default system bridge is Open vSwitch (OVS). The OVS bridge is a VLAN-aware system bridge, which acts as the Network Functions Virtualization (NFV) backplane to which the VNFs and FPCs connect. However, you can choose to create a custom bridge based on your requirement. This example explains how to configure service chaining using a custom bridge.

Topology

This example uses the topology shown in Figure 13 on page 146.

Figure 13: Service Chaining Using a Custom Bridge



Configuration

IN THIS SECTION

Configuring VLANs and Creating the Custom Bridge

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces:

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan200 vlan-id 200
```

2. Create a custom bridge:

```
user@host# set vmhost vlans custom-br vlan-id none
```

3. Map the Layer 3 interface to the custom bridge:

```
user@host# set vmhost virtualization-options interfaces ge-1/0/2 mapping vlan custom-br
```

Configuring the Layer 2 Datapath

Step-by-Step Procedure

1. Configure the LAN-side front panel ports and add them to the LAN-side VLAN.

```
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members vlan200
```

2. Configure the internal-facing interfaces as trunk ports and add them to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

```
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching vlan members vlan200
```

Configuring the Layer 3 Datapath

Step-by-Step Procedure

1. Configure VLAN tagging on ge-1/0/0:

```
user@host# set interfaces ge-1/0/0 vlan-tagging
user@host# set interfaces ge-1/0/0 unit 0 vlan-id 100
user@host# set interfaces ge-1/0/0 unit 0 family inet address 192.0.2.1/24
```

2. Configure VLAN tagging on ge-1/0/2:

```
user@host# set interfaces ge-1/0/2 vlan-tagging
user@host# set interfaces ge-1/0/2 unit 0 vlan-id 200
user@host# set interfaces ge-1/0/2 unit 0 family inet address 203.0.113.2/24
```

Configuring the VNF

Step-by-Step Procedure

> **NOTE:** This example uses a Layer 2 VNF.

1. Launch the VNF:

   ```
   user@host# set virtual-network-functions vnf-name image /var/public/centos-updated1.img
   user@host# set virtual-network-functions vnf-name image image-type raw
   ```

2. Specify the number of CPUs required for the VNF:

   ```
   user@host# set virtual-network-functions vnf-name virtual-cpu count 1
   ```

3. Pin a virtual CPU to a physical CPU:

   ```
   user@host# set virtual-network-functions vnf-name virtual-cpu 0 physical-cpu 2
   ```

4. Configure the vmhost instance:

   ```
   user@host# set vmhost vlans vlan200 vlan-id 200
   ```

5. Create a VNF interface on the custom OVS bridge:

   ```
   user@host# set virtual-network-functions vnf-name interfaces eth2 mapping vlan members
   custom-br
   ```

6. Create a VNF interface on the OVS bridge:

   ```
   user@host# set virtual-network-functions vnf-name interfaces eth3 mapping vlan members vlan200
   ```

7. Specify the memory allocation for the VNF:

   ```
   user@host# set virtual-network-functions vnf-name memory size 1048576
   ```

Verifying the Configuration

IN THIS SECTION

Verify the Control Plane Configuration

Purpose

Verify the control plane configuration:

Action

- Verify that the VLANs are configured:

```
user@host > show vlans
```

```
Routing instance        VLAN name             Tag          Interfaces
default-switch          default               1

default-switch          vlan100               100
                                                           ge-0/0/0.0*
                                                           sxe-0/0/0.0*
default-switch          vlan200               200
                                                           sxe-0/0/1.0*
                                                           xe-0/0/12.0*
```

- Verify the vmhost VLANs:

```
user@host> show vmhost vlans
```

```
Routing instance        VLAN name             Tag          Interfaces
vmhost                  custom-br                            vnf-name_eth2.0
vmhost                  vlan200               200            vnf-name_eth3.0
```

- Verify that the VNF is operational. The **State** field shows **Running** for VNFs that are up.

```
user@host> show virtual-network-functions
```

```
ID       Name                                                State      Liveliness
------------------------------------------------------------------------------
4        vnf-name                                             Running    alive
1        vjunos0                                              Running    alive
```

The **Liveliness** field of the VNF indicates whether the internal management IP address of the VNF is reachable from the Junos Control Plane (JCP).

To view more details of the VNF:

`user@host>` **show virtual-network-functions vnf-name detail**

```
Virtual Network Function Information
------------------------------------

Id:                 4
Name:               vnf-name
State:              Running
Liveliness:         alive
IP Address:         192.0.2.100
VCPUs:              1
Maximum Memory:     1048576 KiB
Used Memory:        1048576 KiB
Used 1G Hugepages:  0
Used 2M Hugepages:  0
Error:              None
```

Verifying the Data Plane Configuration

Purpose

Verify the data plane configuration.

Action

- Verify the status of the Layer 2 (ge-0/0/x) and Layer 3 (ge-1/0/x) interfaces.

  `user@host >` **show interfaces *interface-name* statistics**

  For example:

  `user@host >` **show interfaces ge-0/0/0 statistics**

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 149, SNMP ifIndex: 517
```

```
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, Duplex: Full-Duplex, BPDU Error: None,
  Loop Detect PDU Error: None, Ethernet-Switching Error: None, MAC-REWRITE Error:
 None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online, IEEE
802.3az Energy Efficient Ethernet: Disabled, Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 30:7c:5e:4c:78:03, Hardware address: 30:7c:5e:4c:78:03
  Last flapped   : 2018-11-26 11:03:32 UTC (04:25:39 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

  Logical interface ge-0/0/0.0 (Index 330) (SNMP ifIndex 519)
    Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
    Input packets : 0
    Output packets: 0
    Protocol eth-switch, MTU: 1514
      Flags: Trunk-Mode
```

- Verify the status of the interfaces on the OVS and the custom bridge:

  `user@host >` **show vmhost network nfv-back-plane**

```
Network Name : custom-br

   Interface : custom-br
   Type : internal, Link type : Full-Duplex, MAC : 2e:8e:a3:e3:e5:40
   MTU : [], Link State :down, Admin State : down
   IPV4 : None, Netmask : None
```

```
   IPV6 : None, IPV6 netmask : None
       Rx-packets :       0
       Rx-drops   :       0
       Rx-errors  :       0
       Tx-packets :       0
       Tx-drops   :       0
       Tx-errors  :       0

   Interface : vnf-name_eth2
   Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
   MTU : 1500, Link State :down, Admin State : up
   IPV4 : None, Netmask : None
   IPV6 : None, IPV6 netmask : None
       Rx-packets :       0
       Rx-drops   :       0
       Rx-errors  :       0
       Tx-packets :       0
       Tx-drops   :       0
       Tx-errors  :       0


Network Name : ovs-sys-br

   Interface : ovs-sys-br
   Type : internal, Link type : Full-Duplex, MAC : 66:9c:3f:25:04:40
   MTU : [], Link State :down, Admin State : down
   IPV4 : None, Netmask : None
   IPV6 : None, IPV6 netmask : None
       Rx-packets :       0
       Rx-drops   :       0
       Rx-errors  :       0
       Tx-packets :       0
       Tx-drops   :       0
       Tx-errors  :       0

   Interface : dpdk0
   Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:1a:c6:ee
   MTU : [], Link State :up, Admin State : up
   IPV4 : None, Netmask : None
   IPV6 : None, IPV6 netmask : None
       Rx-packets :       0
       Rx-drops   :       0
       Rx-errors  :       0
       Tx-packets :       0
```

```
    Tx-drops   :       0
    Tx-errors  :       0

Interface : dpdk1
Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:7b:6c:47
MTU : [], Link State :up, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0

Interface : l3_h_ge_1_0_0
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0

Interface : l3_h_ge_1_0_1
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0

Interface : l3_h_ge_1_0_2
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
```

```
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0

Interface : vnf-name_eth3
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : 1500, Link State :down, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
    Rx-packets :       0
    Rx-drops   :       0
    Rx-errors  :       0
    Tx-packets :       0
    Tx-drops   :       0
    Tx-errors  :       0
```

Example: Configuring Service Chaining for LAN Routing on NFX350 Devices

IN THIS SECTION

This example shows how to configure service chaining for LAN routing.

Requirements

This example uses an NFX350 device running Junos OS Release 19.4R1.

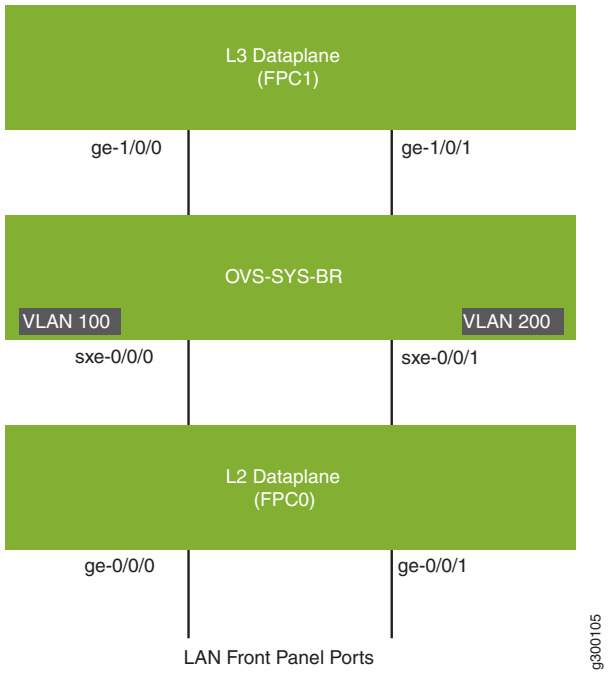
Overview

This example explains how to configure the various layers of the device to enable traffic flow within a LAN network.

Topology

This example uses the topology shown in Figure 14 on page 155.

Figure 14: Service Chaining for LAN Routing



Configuration

IN THIS SECTION

Configuring the Layer 2 Datapath

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan200 vlan-id 200
```

2. Configure the LAN-side front panel ports and add them to the LAN-side VLAN.

```
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@jcp# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan200
```

3. Configure the internal-facing interfaces as trunk ports and add them to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

```
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching vlan members vlan200
```

Configuring the Layer 3 Datapath

Step-by-Step Procedure

1. Configure VLAN tagging on ge-1/0/0:

```
user@host# set interfaces ge-1/0/0 vlan-tagging
```

```
user@host# set interfaces ge-1/0/0 unit 0 vlan-id 100
user@host# set interfaces ge-1/0/0 unit 0 family inet address 192.0.2.1/24
```

2. Configure VLAN tagging on ge-1/0/1:

```
user@host# set interfaces ge-1/0/1 vlan-tagging
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 200
user@host# set interfaces ge-1/0/1 unit 0 family inet address 203.0.113.2/24
```

RELATED DOCUMENTATION

Example: Configuring Service Chaining for LAN-WAN Routing

Example: Configuring Service Chaining for LAN to WAN Routing on NFX350 Devices

IN THIS SECTION

This example shows how to configure service chaining for LAN to WAN routing.

Requirements

This example uses an NFX350 device running Junos OS Release 19.4R1.

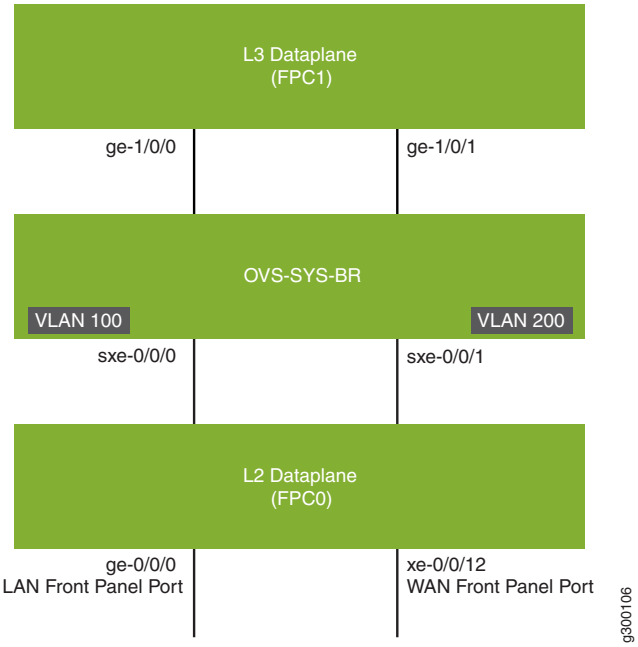
Overview

This example explains how to configure the various layers of the device to enable traffic from the LAN network to enter the device, flow through the OVS, exit the device, and enter the WAN network.

Topology

This example uses the topology shown in Figure 15 on page 158.

Figure 15: Service Chaining for LAN to WAN Routing



Configuration

IN THIS SECTION

Configuring the Layer 2 Datapath

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan200 vlan-id 200
```

2. Configure the LAN-side front panel ports and add them to the LAN-side and WAN-side VLANs.

```
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members vlan200
```

3. Configure the internal-facing interface, sxe-0/0/0, as a trunk port and add it to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

```
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
```

4. Configure the internal-facing interface, sxe-0/0/1, as a trunk port and add it to the WAN-side VLAN.

```
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/1 unit 0 family ethernet-switching vlan members vlan200
```

Configuring the Layer 3 Datapath

Step-by-Step Procedure

1. Configure VLAN tagging on ge-1/0/0:

```
user@host# set interfaces ge-1/0/0 vlan-tagging
user@host# set interfaces ge-1/0/0 unit 0 vlan-id 100
user@host# set interfaces ge-1/0/0 unit 0 family inet address 192.0.2.1/24
```

2. Configure VLAN tagging on ge-1/0/1:

```
user@host# set interfaces ge-1/0/1 vlan-tagging
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 200
user@host# set interfaces ge-1/0/1 unit 0 family inet address 203.0.113.2/24
```

Verification

IN THIS SECTION

Verifying the Status of the Interfaces

Purpose

Verify the status of the Layer 2 and Layer 3 interfaces.

Action

- Verify the status of the Layer 2 (ge-0/0/x) and Layer 3 (ge-1/0/x) interfaces.

  `user@host>` **show interfaces *interface-name* statistics**

  For example:

  `user@host>` **show interfaces ge-0/0/0 statistics**

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 144, SNMP ifIndex: 518
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 1000mbps,
  BPDU Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,

  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 00:00:5e:00:53:43, Hardware address: 00:00:5e:00:53:43
  Last flapped   : 2018-04-18 05:38:22 UTC (2d 10:07 ago)
  Statistics last cleared: Never
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Input errors: 0, Output errors: 0
  Active alarms  : None
  Active defects : None
```

```
PCS statistics                      Seconds
  Bit errors                             0
  Errored blocks                         0
Ethernet FEC statistics              Errors
  FEC Corrected Errors                    0
  FEC Uncorrected Errors                  0
  FEC Corrected Errors Rate               0
  FEC Uncorrected Errors Rate             0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled

Logical interface ge-0/0/0.0 (Index 333) (SNMP ifIndex 524)
  Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
  Input packets : 147888
  Output packets: 22
  Protocol eth-switch, MTU: 9192
    Flags: Is-Primary
```

Example: Configuring Service Chaining for LAN to WAN Routing through Third-party VNFs on NFX350 Devices

IN THIS SECTION

This example shows how to configure service chaining for LAN to WAN routing through third-party VNFs on NFX350 devices.

Requirements

This example uses an NFX350 device running Junos OS Release 19.4R1.

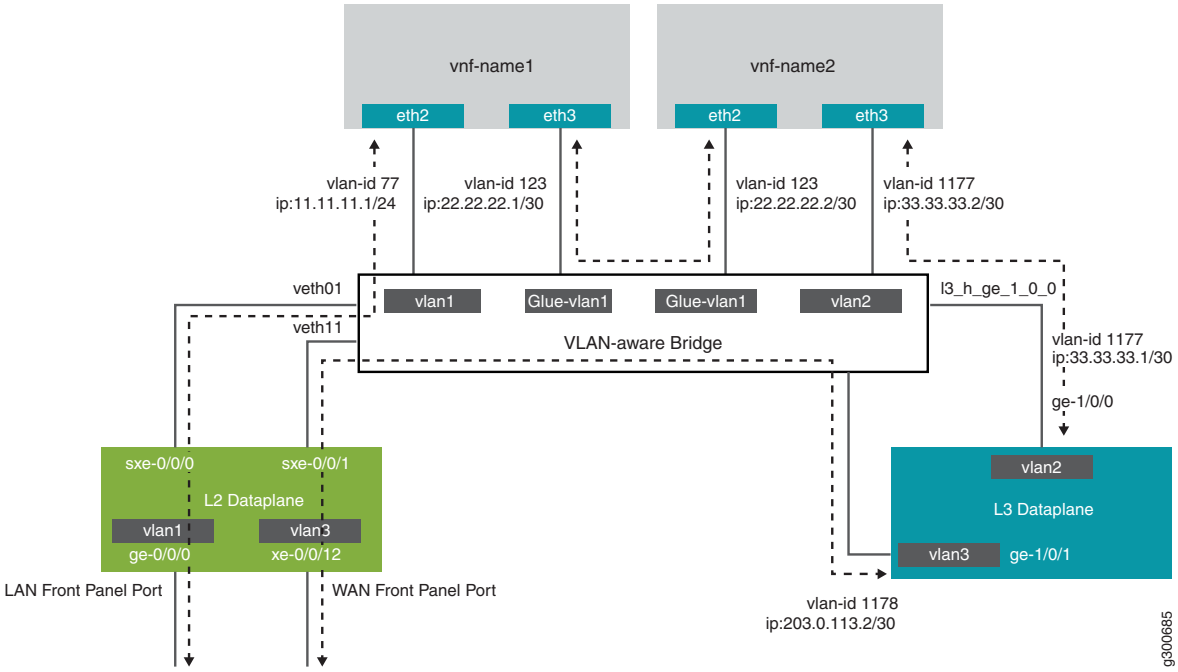
Overview

This example explains how to configure the various layers of the device to enable traffic from the LAN network to enter the device, flow through the OVS bridge and third-party VNFs, exit the device, and enter the WAN network.

Topology

This example uses the topology shown in Figure 16 on page 162.

Figure 16: Service Chaining for LAN to WAN Routing through Third-party VNFs



Configuration

IN THIS SECTION

Configuring the Layer 2 Datapath (JCP LAN Interfaces)

Step-by-Step Procedure

1. Connect to the JCP.

```
user@host:~ # cli
user@host>
user@host> configure
[edit]
user@host#
```

2. Configure VLANs for the LAN-side interfaces.

```
user@host# set vlans vlan1 vlan-id 77
```

3. Configure the LAN-side front panel ports and add them to the LAN-side VLANs. The LAN-side port is typically an access port, and can be a trunk port if required

```
user@host# set interfaces ge-0/0/0.0 family ethernet-switching vlan members vlan1
```

4. Configure the internal-facing interface, sxe-0/0/0, as a trunk port and add it to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

```
user@host# set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
```

```
user@host# set interfaces sxe-0/0/0.0 family ethernet-switching vlan members vlan1
```

Verifying the Performance Mode of the NFX350 Device

Purpose

Verify the performance mode of the NFX350 device and check the CPU availability. If the NFX350 device is operating in throughput mode, you must change it to either compute or hybrid mode by using the **request vmhost mode** command.

For more information about the device performance modes, see "NFX350 Overview" on page 17.

Action

```
user@host> show vmhost mode | no-more
```

```
Mode:
--------
Current Mode: compute

CPU Allocations:
Name                                Configured                                Used
-------------------------------------------------------------------------------------------------------------
Junos Control Plane                 16                                        16,6
Juniper Device Manager               16                                        16
LTE                                 16                                        -
NFV Backplane Control Path           16                                        16
NFV Backplane Data Path             1,2,3                                     1,2,3
Layer 2 Control Path                 -                                         -
Layer 2 Data Path                    -                                         -
Layer 3 Control Path                 0                                         0
Layer 3 Data Path                    4,5                                       4,5
CPUs available for VNFs
6,7,8,9,10,11,12,13,14,15,22,23,24,25,26,27,28,29,30,31 -
CPUs turned off                      17,18,19,20,21                            -

Memory Allocations:
Name                                Configured                                Used
-------------------------------------------------------------------------------------------------------------
Junos Control Plane (mB)            2048                                      2002
NFV Backplane 1G hugepages           12                                        18
NFV Backplane 2M hugepages           -                                         0
Layer 2 1G hugepages                 -                                         -
Layer 2 2M hugepages                 -                                         -
```

```
Layer 3 1G hugepages                6                                    6
Layer 3 2M hugepages                20481                                20481
```

Configuring the Hugepages for VNF

Step-by-Step Procedure

> **NOTE:** It is recommended to reboot the device if the configured number of hugepages are not allocated.

1. Check the memory availability:

   ```
   user@host> show system visibility memory | no-more
   ```

   ```
   Memory Information
   ------------------

   Virtual Memory:
   ---------------
   Total       (KiB): 131042784
   Used        (KiB): 67141828
   Available   (KiB): 66151972
   Free        (KiB): 63900956
   Percent Used     : 49.5

   Huge Pages:
   ------------
   Total 1GiB Huge Pages:      18
   Free 1GiB Huge Pages:       0
   Configured 1GiB Huge Pages: 0
   Total 2MiB Huge Pages:      20481
   Free 2MiB Huge Pages:       0
   Configured 2MiB Huge Pages: 0
   ```

2. Configure hugepages:

   ```
   user@host> configure
   [edit]
   user@host#
   user@host# set system memory hugepages page-size 1024 page-count 10
   user@host# commit
   ```

3. Verify whether hugepages is configured:

```
user@host# run show system visibility memory | no-more
```

```
Memory Information
------------------

Virtual Memory:
---------------
Total       (KiB): 131042784
Used        (KiB): 77624220
Available   (KiB): 55670868
Free        (KiB): 53418564
Percent Used     : 57.5

Huge Pages:
------------
Total 1GiB Huge Pages:      28
Free 1GiB Huge Pages:       10
Configured 1GiB Huge Pages: 10
Total 2MiB Huge Pages:      20481
Free 2MiB Huge Pages:       0
Configured 2MiB Huge Pages: 0

Hugepages Usage:
-----------------------------------------------------------------------------------------------------
Name                              Type                                Used 1G
Hugepages  Used 2M Hugepages
--------------------------------- ----------------------------------
------------------ ------------------
ovs-vswitchd                      other process                       18
          0
srxpfe                            other process                       6
          20481
```

Configuring VNFs

Step-by-Step Procedure

Configure VNF-1:

1. Load the VNF image on the device from the remote location:

   > **NOTE:** You can save the VNF image in the **/var/public** directory if you are using up to two VNFs. If you are using more than two VNFs, save the files on an external SSD. If you are using an external SSD for VNFs, make sure to initialize and add the SSD to the device. For more information, see "Configuring the Solid State Disk on NFX350 Device" on page 63.

   ```
   user@host> file copy source-address /var/public/vnf-1_junos-vsrx3-x86-64-19.1R1-S1.3.qcow2
   ```

2. Launch the VNF:

   ```
   user@host> set virtual-network-functions VNF-1 image
   /var/public/vnf-1_junos-vsrx3-x86-64-19.1R1-S1.3.qcow2
   ```

3. Connect a virtual CPUs to physical CPUs:

   ```
   user@host> set virtual-network-functions VNF-1 virtual-cpu 0 physical-cpu 6
   user@host> set virtual-network-functions VNF-1 virtual-cpu 1 physical-cpu 7
   ```

4. Specify the number of CPUs required for the VNF:

   ```
   user@host> set virtual-network-functions VNF-1 virtual-cpu count 2
   ```

5. Enable hardware virtualization or hardware acceleration for VNF CPUs:

   ```
   user@host> set virtual-network-functions VNF-1 virtual-cpu features hardware-virtualization
   ```

6. Configure the VNF interfaces as trunk ports and add them to the LAN-side VLAN:

   ```
   user@host> set virtual-network-functions VNF-1 interfaces eth2 mapping vlan mode trunk
   user@host> set virtual-network-functions VNF-1 interfaces eth2 mapping vlan members vlan1
   user@host> set virtual-network-functions VNF-1 interfaces eth3 mapping vlan mode trunk
   user@host> set virtual-network-functions VNF-1 interfaces eth3 mapping vlan members glue-vlan1
   ```

7. Specify the memory allocation for the VNF:

   ```
   user@host> set virtual-network-functions VNF-1 memory size 4194304
   user@host> set virtual-network-functions VNF-1 memory features hugepages
   ```

Step-by-Step Procedure

Configure VNF-2:

1. Load the VNF image on the device from the remote location:

   ```
   user@host> file copy source-address /var/public/vnf-2-junos-vsrx3-x86-64-19.1R1-S1.3.qcow2
   ```

2. Launch the VNF:

   ```
   user@host> set virtual-network-functions VNF-2 image
   /var/public/vnf-2-junos-vsrx3-x86-64-19.1R1-S1.3.qcow2
   ```

3. Connect a virtual CPUs to physical CPUs:

   ```
   user@host> set virtual-network-functions VNF-2 virtual-cpu 0 physical-cpu 8
   user@host> set virtual-network-functions VNF-2 virtual-cpu 1 physical-cpu 9
   ```

4. Specify the number of CPUs required for the VNF:

   ```
   user@host> set virtual-network-functions VNF-2 virtual-cpu count 2
   ```

5. Enable hardware virtualization or hardware acceleration for VNF CPUs:

   ```
   user@host> set virtual-network-functions VNF-2 virtual-cpu features hardware-virtualization
   ```

6. Configure the VNF interfaces as trunk ports and add them to the LAN-side VLAN:

   ```
   user@host> set virtual-network-functions VNF-2 interfaces eth2 mapping vlan mode trunk
   user@host> set virtual-network-functions VNF-2 interfaces eth2 mapping vlan members glue-vlan1
   user@host> set virtual-network-functions VNF-2 interfaces eth3 mapping vlan mode trunk
   user@host> set virtual-network-functions VNF-2 interfaces eth3 mapping vlan members vlan2
   ```

7. Specify the memory allocation for the VNF:

   ```
   user@host> set virtual-network-functions VNF-2 memory size 4194304
   user@host> set virtual-network-functions VNF-2 memory features hugepages
   ```

Configuring the Layer 3 Datapath (WAN Interfaces)

Step-by-Step Procedure

1. Configure the internal-facing L3 Dataplane interface as a VLAN-tagged interface and assign an IP address to it:

   ```
   user@host# set interfaces ge-1/0/0 vlan-tagging
   user@host# set interfaces ge-1/0/0.0 vlan-id 1177
   user@host# set interfaces ge-1/0/0.0 family inet address 33.33.33.1/30
   ```

2. Map the Layer 3 interface to the Open vSwitch (OVS) and commit the configuration:

   ```
   user@host# set vmhost virtualization-options interfaces ge-1/0/1
   user@host# commit
   ```

3. Configure the external-facing L3 Dataplane interface as a VLAN-tagged interface and assign an IP address to it:

   ```
   user@host# set interfaces ge-1/0/1 vlan-tagging
   user@host# set interfaces ge-1/0/1.0 vlan-id 1178
   user@host# set interfaces ge-1/0/1.0 family inet address 203.0.113.2/30
   ```

4. Configure a VLAN for the WAN-side JCP interfaces:

   ```
   user@host# set vlans vlan3 vlan-id 1178
   ```

5. Configure the WAN-side internal-facing interface as a trunk port and add it to the WAN-side VLAN:

   ```
   user@host# set interfaces sxe-0/0/1.0 family ethernet-switching interface-mode trunk
   user@host# set interfaces sxe-0/0/1.0 family ethernet-switching vlan members vlan3
   ```

6. Configure the WAN-side front panel port and add it to the WAN-side VLAN:

   ```
   user@host# set interfaces xe-0/0/12.0 family ethernet-switching interface-mode access
   user@host# set interfaces xe-0/0/12.0 family ethernet-switching vlan members vlan3
   ```

7. Commit the configuration:

   ```
   user@host# commit
   ```

Configuring the VNF Interfaces for Creating the Service Chain

Step-by-Step Procedure

1. Check the MAC addresses of the VNF interfaces:

   `user@host#` **run show system visibility network**

```
VNF MAC Addresses
------------------------------------------------------------
VNF                                     MAC
--------------------------------------- -----------------
VNF-1_ethdef0                           D0:DD:49:E8:B6:CA
VNF-1_ethdef1                           D0:DD:49:E8:B6:CB
VNF-1_eth2                              D0:DD:49:E8:B6:CC
VNF-1_eth3                              D0:DD:49:E8:B6:C7
VNF-2_ethdef0                           D0:DD:49:E8:B6:C8
VNF-2_ethdef1                           D0:DD:49:E8:B6:C9
VNF-2_eth2                              D0:DD:49:E8:B6:CD
VNF-2_eth3                              D0:DD:49:E8:B6:CE

VNF Internal IP Addresses
------------------------------------------------------------
VNF                                     IP
--------------------------------------- ---------------
VNF-1                                   192.0.2.100
VNF-2                                   192.0.2.101

Free Virtual Functions
----------------------
PF        VF
--------- ------------
hsxe0     0000:b7:03.6
hsxe0     0000:b7:03.4
hsxe0     0000:b7:03.5
hsxe0     0000:b7:02.3
hsxe0     0000:b7:02.2
hsxe0     0000:b7:02.1
hsxe0     0000:b7:02.7
hsxe0     0000:b7:02.6
hsxe0     0000:b7:02.5
hsxe0     0000:b7:02.4
hsxe1     0000:b7:07.4
hsxe1     0000:b7:06.7
hsxe1     0000:b7:06.6
hsxe1     0000:b7:06.5
hsxe1     0000:b7:06.4
hsxe1     0000:b7:06.3
hsxe1     0000:b7:06.2
```

```
hsxe1     0000:b7:06.1
hsxe1     0000:b7:07.5
hsxe1     0000:b7:07.6
hsxe2     0000:b7:0b.6
hsxe2     0000:b7:0b.5
hsxe2     0000:b7:0b.4
hsxe2     0000:b7:0a.4
hsxe2     0000:b7:0a.5
hsxe2     0000:b7:0a.6
hsxe2     0000:b7:0a.7
hsxe2     0000:b7:0a.1
hsxe2     0000:b7:0a.2
hsxe2     0000:b7:0a.3
hsxe3     0000:b7:0f.6
hsxe3     0000:b7:0f.5
hsxe3     0000:b7:0f.4
hsxe3     0000:b7:0e.1
hsxe3     0000:b7:0e.2
hsxe3     0000:b7:0e.3
hsxe3     0000:b7:0e.4
hsxe3     0000:b7:0e.5
hsxe3     0000:b7:0e.6
hsxe3     0000:b7:0e.7

VNF Interfaces
-------------------------------------------------------------------------------------------
VNF                  Interface Type      Source       Model      MAC
        VLAN-ID
-------------------- --------- --------- ------------ ----------
-------------------- -------
VNF-1              vnet4     network   default      virtio     d0:dd:49:e8:b6:ca
    --
VNF-1              vnet5     bridge    eth0br       virtio     d0:dd:49:e8:b6:cb
    --
VNF-1              VNF-1_eth2 vhostuser --         virtio     d0:dd:49:e8:b6:cc
    --
VNF-1              VNF-1_eth3 vhostuser --         virtio     d0:dd:49:e8:b6:c7
    --
VNF-2              vnet6     network   default      virtio     d0:dd:49:e8:b6:c8
    --
VNF-2              vnet7     bridge    eth0br       virtio     d0:dd:49:e8:b6:c9
    --
VNF-2              VNF-2_eth2 vhostuser --         virtio     d0:dd:49:e8:b6:cd
    --
```

```
VNF-2              VNF-2_eth3 vhostuser --         virtio     d0:dd:49:e8:b6:ce
    --

OVS Interfaces
------------------------
NAME              MTU
----------------- ------
ovs-sys-br        1500
dpdk2             9216
xdsl_eth0         9192
l3_h_ge_1_0_1     9216
l3_h_ge_1_0_0     1500
dpdk0             9216
VNF-2_eth2        1500
dpdk1             9216
VNF-1_eth3        1500
dpdk3             9216
VNF-1_eth2        1500
VNF-2_eth3        1500
```

2. Access the VNF (VNF-1) from the JCP through the console:

   `user@host>` **request virtual-network-functions console VNF-1**

```
Internal instance: VNF-1
Connected to domain VNF-1
```

3. Log in to the console:

   `user@host:~ #` **cli**
   `user@host>`

4. Check the status of the interfaces:

   - `user@host#` **show interfaces terse | no-more**

```
Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
gr-0/0/0                up    up
ip-0/0/0                up    up
lsq-0/0/0               up    up
lt-0/0/0                up    up
mt-0/0/0                up    up
sp-0/0/0                up    up
```

```
sp-0/0/0.0              up    up   inet
                                   inet6
sp-0/0/0.16383          up    up   inet
ge-0/0/1                up    up
ge-0/0/2                up    up
dsc                     up    up
fti0                    up    up
fxp0                    up    up
fxp0.0                  up    up
gre                     up    up
ipip                    up    up
irb                     up    up
lo0                     up    up
lo0.16384               up    up   inet     127.0.0.1           --> 0/0
lo0.16385               up    up   inet     10.0.0.1            --> 0/0
                                            10.0.0.16           --> 0/0
                                            128.0.0.1           --> 0/0
                                            128.0.0.4           --> 0/0
                                            128.0.1.16          --> 0/0
lo0.32768               up    up
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
pp0                     up    up
ppd0                    up    up
ppe0                    up    up
st0                     up    up
tap                     up    up
vlan                    up    down
```

- `user@host>` **show interfaces ge-0/0/0 | no-more**

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 135, SNMP ifIndex: 508
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Half-duplex,

  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
 Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,

  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

  Remote fault: Online
  Device flags   : Present Running
```

```
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: d0:dd:49:e8:b6:cb, Hardware address: d0:dd:49:e8:b6:cb
  Last flapped   : 2020-05-11 10:22:06 UTC (00:46:40 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
 PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  Interface transmit statistics: Disabled
```

- user@host> **show interfaces fxp0 | no-more**

```
Physical interface: fxp0, Enabled, Physical link is Up
  Interface index: 65, SNMP ifIndex: 1
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Current address: d0:dd:49:e8:b6:ca, Hardware address: d0:dd:49:e8:b6:ca
  Last flapped   : 2020-05-11 10:21:26 UTC (00:47:53 ago)
    Input packets : 1484
    Output packets: 0

  Logical interface fxp0.0 (Index 3) (SNMP ifIndex 13)
    Flags: Up SNMP-Traps Encapsulation: ENET2
    Input packets : 1452
    Output packets: 0
```

- user@host> **show interfaces ge-0/0/1 | no-more**

```
Physical interface: ge-0/0/1, Enabled, Physical link is Up
  Interface index: 136, SNMP ifIndex: 517
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Half-duplex,
```

```
  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
 Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,

 Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: d0:dd:49:e8:b6:cc, Hardware address: d0:dd:49:e8:b6:cc
  Last flapped   : 2020-05-11 10:22:06 UTC (00:47:39 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  Interface transmit statistics: Disabled
```

- user@host> **show interfaces ge-0/0/2 | no-more**

```
Physical interface: ge-0/0/2, Enabled, Physical link is Up
  Interface index: 137, SNMP ifIndex: 518
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Half-duplex,

  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
 Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,

 Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: d0:dd:49:e8:b6:c7, Hardware address: d0:dd:49:e8:b6:c7
  Last flapped   : 2020-05-11 10:22:06 UTC (00:47:52 ago)
```

```
Input rate     : 0 bps (0 pps)
Output rate    : 0 bps (0 pps)
Active alarms  : None
Active defects : None
PCS statistics                      Seconds
  Bit errors                             0
  Errored blocks                         0
Ethernet FEC statistics              Errors
  FEC Corrected Errors                    0
  FEC Uncorrected Errors                  0
  FEC Corrected Errors Rate               0
  FEC Uncorrected Errors Rate             0
Interface transmit statistics: Disabled
```

5. Set the root password:

   ```
   user@host# set system root-authentication plain-text-password
   ```

6. At the first prompt, enter the new root password. At the second prompt, reenter the new root password:

   ```
   New password:
   Retype new password:
   ```

7. After you have finished configuring the password, commit the configuration:

   ```
   user@host# commit
   commit complete
   ```

8. Configure the WAN-side internal-facing interface (ge-0/0/1) as a VLAN-tagged interface and assign an IP address to it:

   ```
   user@host# set interfaces ge-0/0/1 vlan-tagging
   user@host# set interfaces ge-0/0/1 unit 0 vlan-id 77
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 11.11.11.1/24
   user@host# commit
   commit complete
   ```

9. Configure the WAN-side internal-facing interface (ge-0/0/2) as a VLAN-tagged interface and assign an IP address to it:

   ```
   user@host# set interfaces ge-0/0/2 vlan-tagging
   ```

```
user@host# set interfaces ge-0/0/2 unit 0 vlan-id 123
user@host# set interfaces ge-0/0/2 unit 0 family inet address 22.22.22.1/30
user@host# commit
commit complete
```

10. Access the VNF (VNF-2) from the JCP through the console:

```
user@host> request virtual-network-functions console VNF-2
Internal instance: VNF-2
Connected to domain VNF-2
```

11. Log in to the console:

```
user@host:~ # cli
user@host>
```

12. Check the status of the interfaces:

- `user@host# show interfaces terse | no-more`

```
Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
gr-0/0/0                up    up
ip-0/0/0                up    up
lsq-0/0/0               up    up
lt-0/0/0                up    up
mt-0/0/0                up    up
sp-0/0/0                up    up
sp-0/0/0.0              up    up   inet
                                   inet6
sp-0/0/0.16383          up    up   inet
ge-0/0/1                up    up
ge-0/0/2                up    up
dsc                     up    up
fti0                    up    up
fxp0                    up    up
fxp0.0                  up    up
gre                     up    up
ipip                    up    up
irb                     up    up
lo0                     up    up
lo0.16384               up    up   inet     127.0.0.1           --> 0/0
lo0.16385               up    up   inet     10.0.0.1            --> 0/0
```

```
                                          10.0.0.16           --> 0/0
                                          128.0.0.1           --> 0/0
                                          128.0.0.4           --> 0/0
                                          128.0.1.16          --> 0/0
lo0.32768               up    up
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
pp0                     up    up
ppd0                    up    up
ppe0                    up    up
st0                     up    up
tap                     up    up
vlan                    up    down
```

- `user@host>` **show interfaces ge-0/0/0 | no-more**

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 135, SNMP ifIndex: 508
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Half-duplex,

  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
 Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,

  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: d0:dd:49:e8:b6:c9, Hardware address: d0:dd:49:e8:b6:c9
  Last flapped   : 2020-05-11 10:26:20 UTC (22:53:57 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
```

```
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  Interface transmit statistics: Disabled
```

- user@host> **show interfaces fxp0 | no-more**

```
Physical interface: fxp0, Enabled, Physical link is Up
  Interface index: 65, SNMP ifIndex: 1
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Current address: d0:dd:49:e8:b6:c8, Hardware address: d0:dd:49:e8:b6:c8
  Last flapped   : 2020-05-11 10:25:39 UTC (22:54:38 ago)
    Input packets : 41363
    Output packets: 0

  Logical interface fxp0.0 (Index 3) (SNMP ifIndex 13)
    Flags: Up SNMP-Traps Encapsulation: ENET2
    Input packets : 41320
    Output packets: 0
```

- user@host> **show interfaces ge-0/0/1 | no-more**

```
Physical interface: ge-0/0/1, Enabled, Physical link is Up
  Interface index: 136, SNMP ifIndex: 509
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Half-duplex,

  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
 Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,

  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: d0:dd:49:e8:b6:cd, Hardware address: d0:dd:49:e8:b6:cd
  Last flapped   : 2020-05-11 10:26:20 UTC (22:53:57 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
```

```
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  Interface transmit statistics: Disabled
```

- `user@host>` **show interfaces ge-0/0/2 | no-more**

```
Physical interface: ge-0/0/2, Enabled, Physical link is Up
  Interface index: 137, SNMP ifIndex: 510
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Half-duplex,

  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
 Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,

 Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: d0:dd:49:e8:b6:ce, Hardware address: d0:dd:49:e8:b6:ce
  Last flapped   : 2020-05-11 10:26:20 UTC (22:53:57 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  Interface transmit statistics: Disabled
```

13. Set the root password:

```
user@host# set system root-authentication plain-text-password
```

14. At the first prompt, enter the new root password. At the second prompt, reenter the new root password:

```
New password:
Retype new password:
```

15. After you have finished configuring the password, commit the configuration:

```
user@host# commit
commit complete
```

16. Configure the WAN-side internal-facing interface (ge-0/0/1) as a VLAN-tagged interface and assign an IP address to it:

```
user@host# set interfaces ge-0/0/1 vlan-tagging
user@host# set interfaces ge-0/0/1 unit 0 vlan-id 123
user@host# set interfaces ge-0/0/1 unit 0 family inet address 22.22.22.2/30
user@host# commit
commit complete
```

17. Configure the WAN-side internal-facing interface (ge-0/0/2) as a VLAN-tagged interface and assign an IP address to it:

```
user@host# set interfaces ge-0/0/2 vlan-tagging
user@host# set interfaces ge-0/0/2 unit 0 vlan-id 1177
user@host# set interfaces ge-0/0/2 unit 0 family inet address 33.33.33.2/30
user@host# commit
commit complete
```

Configuring Security in NFX350

Step-by-Step Procedure

1. Clear the current security settings:

```
user@host# delete security
```

2. Configure security options:

```
user@host# set security forwarding-options family inet6 mode flow-based
```

3. Configure security policies:

```
user@host# set security policies default-policy permit-all
```

4. Configure security zones:

```
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces all
```

Configuring Security in vSRX VNFs

Step-by-Step Procedure

1. Clear the current security settings:

```
user@host# delete security
```

2. Configure security options:

```
user@host# set security forwarding-options family inet6 mode flow-based
```

3. Configure security policies:

```
user@host# set security policies default-policy permit-all
```

4. Configure security zones:

```
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces all
```

11

CHAPTER

Troubleshooting

Recovering the Root Password for NFX150, NFX250 NextGen, and NFX350 Devices

The root password on your Junos OS-enabled device helps to prevent unauthorized users from making changes to your network.

If you forget the root password, you can use the password recovery procedure to reset the root password.

> **NOTE:** You need console access to the device to recover the root password.

To recover the root password:

1. Power off the device by switching off the AC power outlet of the device or, if necessary, by pulling the power cords out of the device's power supplies.

2. Turn off the power to the management device, such as a PC or laptop computer, that you want to use to access the CLI.

3. Plug one end of the Ethernet rollover cable supplied with the device into the RJ-45 to DB-9 serial port adapter supplied with the device.

4. Plug the RJ-45 to DB-9 serial port adapter into the serial port on the management device.

5. Connect the other end of the Ethernet rollover cable to the console port on the device.

6. Turn on the power to the management device.

7. On the management device, start any asynchronous terminal emulation application (such as Microsoft Windows HyperTerminal), and select the port to be used.

8. Configure the port settings as follows:
   - Bits per second—9600
   - Data bits—8
   - Parity—None
   - Stop bits—1
   - Flow control—None

9. Power on the device by plugging the power cords into the device's power supply (if necessary), or by turning on the power to the device by switching on the AC power outlet that the device is plugged into.

   The terminal emulation screen on your management device displays the device's boot sequence.

```
i2cset -y 5 0x19 0xff 0x05
i2cset -y 5 0x19 0x2d 0x81
i2cset -y 5 0x19 0x15 0x12
i2cset -y 5 0x18 0xff 0x05
i2cset -y 5 0x18 0x2d 0x82
i2cset -y 5 0x18 0x15 0x12
 * Stopping virtualization library daemon: libvirtd
```

   [This message is truncated...]

```
Checking Prerequisites
jdm docker container is in Exit state, required to cleanup, please wait...
9dba6935234b
[  OK  ]
Launching jdm container 'jdm'...
```

10. When the prompt shows **Launching jdm container 'jdm'**, press **Ctrl+C**. The **Main Menu** appears.

```
Main Menu

1.  Boot [J]unos volume
2.  Boot Junos volume in [S]afe mode
3.  [R]eboot
4.  [B]oot menu
5.  [M]ore options
```

11. From the **Main Menu**, select **5. [M]ore options**. The **Options Menu** appears.

```
Options Menu

1.  Recover [J]unos volume
2.  Recovery mode - [C]LI
3.  Check [F]ile system
4.  Enable [V]erbose boot
5.  [B]oot prompt
6.  [M]ain menu
```

12. From the **Options Menu**, select **2. Recovery mode - [C]LI**. The device reboots into CLI recovery mode.

```
Booting Junos in CLI recovery mode ...

it will boot in recovery mode and will get MGD cli

/packages/sets/active/boot/os-kernel/kernel text=0x444c38 data=0x82348+0x2909a0
 syms=[0x8+0x94c50+0x8+0x8165b]
/packages/sets/active/boot/os-kernel/contents.izo size=0x84d200
/packages/sets/active/boot/os-kernel/miibus.ko size 0x40778 at 0x14bc000
loading required module 'netstack'
/packages/sets/active/boot/netstack/netstack.ko size 0x1386b08 at 0x14fd000
loading required module 'crypto'
```

[This message is truncated...]

```
Starting MGD
mgd: error: could not open database: /var/run/db/schema.db: No such file or
directory
mgd: error: could not open database schema: /var/run/db/schema.db
mgd: error: could not open database schema
mgd: error: database schema is out of date, rebuilding it
mgd: error: could not open database: /var/run/db/juniper.data: No such file or
 directory
mgd: error: Cannot read configuration: Could not open configuration database
mgd: warning: schema: dbs_remap_daemon_index: could not find daemon name 'isdnd'

 Starting CLI ...
```

13. Enter configuration mode in the CLI.

```
root> configure
```

```
Entering configuration mode
```

14. Set the root password.

```
[edit]
root# set system root-authentication plain-text-password
```

15. At the first prompt, enter the new root password:

```
New password:
```

16. At the second prompt, reenter the new root password.

```
Retype new password:
```

17. After you have finished configuring the password, commit the configuration.

```
[edit]
root# commit
commit complete
```

18. Exit configuration mode in the CLI.

```
[edit]
root@host# exit
root@host>
```

19. Exit operational mode in the CLI.

```
root@host> exit
root@host%
```

20. At the shell prompt, type **exit** to reboot the device.

```
root@host% exit
```

RELATED DOCUMENTATION

Configuring the Root Password

Troubleshooting Interfaces on NFX Devices

IN THIS SECTION

Monitoring Interface Status and Traffic on NFX Series Devices

Purpose

View the interface status to monitor bandwidth utilization and traffic statistics of an interface.

Action

To view the status of an interface:

`user@host>` **show interfaces *interface-name***

For example:

- To view the status of an interface for an NFX350 device:

  `user@host>` **show interfaces ge-0/0/0 | no-more**

```
Physical interface: ge-0/0/0, Enabled, Physical link is Down
  Interface index: 150, SNMP ifIndex: 514
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Unknown,
  Speed: 1000mbps, Duplex: Full-Duplex, BPDU Error: None,
  Loop Detect PDU Error: None, Ethernet-Switching Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online,
  IEEE 802.3az Energy Efficient Ethernet: Disabled, Auto-MDIX: Enabled
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: d0:dd:49:e8:6e:7d, Hardware address: d0:dd:49:e8:6e:7d
  Last flapped   : 2020-02-19 06:17:42 UTC (00:25:17 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : LINK
  Active defects : LINK
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled
```

```
  Logical interface ge-0/0/0.0 (Index 74) (SNMP ifIndex 523)
    Flags: Device-Down SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
    Input packets : 0
    Output packets: 0
    Protocol eth-switch, MTU: 1514
```

user@host> **show interfaces xe-0/0/15 | no-more**

```
Physical interface: xe-0/0/15, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 557
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps,
  BPDU Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,

  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: d0:dd:49:e8:6e:8c, Hardware address: d0:dd:49:e8:6e:8c
  Last flapped   : 2020-02-19 06:17:43 UTC (00:25:32 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 232 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled
  Logical interface xe-0/0/15.0 (Index 72) (SNMP ifIndex 558)
    Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
    Input packets : 0
    Output packets: 57
    Protocol eth-switch, MTU: 1514
      Flags: Is-Primary
```

user@host> **show interfaces ge-1/0/1 | no-more**

```
Physical interface: ge-1/0/1, Enabled, Physical link is Up
  Interface index: 168, SNMP ifIndex: 538
  Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Link-mode: Half-duplex,
  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: d0:dd:49:e8:6e:96, Hardware address: d0:dd:49:e8:6e:96
  Last flapped   : 2020-02-19 06:18:30 UTC (00:24:55 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 208 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled
 Logical interface ge-1/0/1.2 (Index 85) (SNMP ifIndex 544)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.2 ]  Encapsulation: ENET2
    Input packets : 0
    Output packets: 19
    Security: Zone: Null
    Protocol inet, MTU: 1500
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 0,
    Curr new hold cnt: 0, NH drop cnt: 0
      Flags: Sendbcast-pkt-to-re
    Protocol inet6, MTU: 1500
    Max nh c
```

- To view the status of an interface for an NFX150 device:

  `user@host>` **show interfaces heth-0-1**

```
Physical interface: heth-0-1, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Copper, MTU: 9192, Speed: 1Gbps, Duplex:
```

```
 Full-duplex, Auto-negotiation: Enabled
  Device flags   : Present Running
  Current address: 00:00:5e:00:53:8e, Hardware address: 00:00:5e:00:53:8e
```

- To view the status of the interface for an NFX250 device:

  user@host> **show interfaces xe-0/0/12**

```
Physical interface: xe-0/0/12, Enabled, Physical link is Up
Interface index: 145, SNMP ifIndex: 509
Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, BPDU Error:
None, Loop Detect PDU Error: None,
Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled
Device flags : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
Link flags : None
CoS queues : 12 supported, 12 maximum usable queues
Current address: 30:7c:5e:4c:78:0f, Hardware address: 30:7c:5e:4c:78:0f
Last flapped : 2018-12-10 19:53:35 UTC (2d 03:08 ago)
Input rate : 0 bps (0 pps)
Output rate : 0 bps (0 pps)
Active alarms : None
Active defects : None
PCS statistics Seconds
Bit errors 0
Errored blocks 0
Ethernet FEC statistics Errors
FEC Corrected Errors 0
FEC Uncorrected Errors 0
FEC Corrected Errors Rate 0
FEC Uncorrected Errors Rate 0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled
```

12
CHAPTER

Operational Commands

request chassis cluster failover node

Syntax

```
request chassis cluster failover node  node-number
redundancy-group  group-number
```

Release Information

Command introduced in Junos OS Release 9.0.

Description

For chassis cluster configurations, initiate manual failover in a redundancy group from one node to the other, which becomes the primary node, and automatically reset the priority of the group to 255. The failover stays in effect until the new primary node becomes unavailable, the threshold of the redundancy group reaches 0, or you use the **request chassis cluster failover reset** command.

After a manual failover, you must use the **request chassis cluster failover reset** command before initiating another failover.

Options

- **node** ***node-number***—Number of the chassis cluster node to which the redundancy group fails over.
- **Range:** 0 through 1
- **redundancy-group** ***group-number***—Number of the redundancy group on which to initiate manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

  Range: 0 through 255

Required Privilege Level

maintenance

RELATED DOCUMENTATION

clear chassis cluster failover-count

request chassis cluster failover reset | **198**

show chassis cluster status | **245**

List of Sample Output

request chassis cluster failover node on page 195

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster failover node

`user@host>` **request chassis cluster failover node 0 redundancy-group 1**

```
Initiated manual failover for redundancy group 1
```

request chassis cluster failover redundancy-group

Syntax

```
request chassis cluster failover node node-number redundancy-group redundancy-group-number
```

Release Information

Command introduced in Junos OS Release 9.0.

Description

For chassis cluster configurations, initiate manual failover in a redundancy group from one node to the other, which becomes the primary node, and automatically reset the priority of the group to 255. The failover stays in effect until the new primary node becomes unavailable, the threshold of the redundancy group reaches 0, or you use the **request chassis cluster failover reset** command.

After a manual failover, you must use the **request chassis cluster failover reset** command before initiating another failover.

Options

- **node** ***node-number***—Number of the chassis cluster node to which the redundancy group fails over.
- **Range:** 0 or 1
- **redundancy-group** ***group-number***—Number of the redundancy group on which to initiate manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

  Range: 0 through 255

Required Privilege Level

maintenance

RELATED DOCUMENTATION

> *Initiating a Chassis Cluster Manual Redundancy Group Failover*
>
> *Verifying Chassis Cluster Failover Status*

List of Sample Output

request chassis cluster failover redundancy-group on page 197

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster failover redundancy-group

```
user@host> request chassis cluster failover redundancy-group 0 node 1
```

```
{primary:node0}
user@host> request chassis cluster failover redundancy-group 0 node 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Initiated manual failover for redundancy group 0
```

request chassis cluster failover reset

Syntax

```
request chassis cluster failover reset
redundancy-group group-number
```

Release Information
Command introduced in Junos OS Release 9.0.

Description
In chassis cluster configurations, undo the previous manual failover and return the redundancy group to its original settings.

Options
redundancy-group ***group-number***—Number of the redundancy group on which to reset manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

Range: 0 through 255

Required Privilege Level
maintenance

RELATED DOCUMENTATION

- *clear chassis cluster failover-count*
- request chassis cluster failover node | **194**
- show chassis cluster status | **245**

List of Sample Output
request chassis cluster failover reset on page 198

Output Fields
When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster failover reset

```
user@host> request chassis cluster failover reset redundancy-group 0
```

request chassis fpc

Syntax

```
request chassis fpc (offline | online | restart) slot slot-number
```

Release Information

Command modified in Junos OS Release 9.2.
Command introduced in Junos OS Release 17.2 for PTX10008 Routers.

Description

Control the operation of the Flexible PIC Concentrator (FPC).

> **NOTE:** The SRX5K-SPC-2-10-40 (SPC1) and SRX5K-SPC-4-15-320 (SPC2) does not support the **request chassis fpc** command. SRX5K-SPC3 card supports **request chassis fpc** command.

Options

offline—Take the FPC offline.

online—Bring the FPC online.

restart—Restart the FPC.

slot *slot-number*—Specify the FPC slot number.

Required Privilege Level

maintenance

RELATED DOCUMENTATION

- *show chassis fpc (View)*

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis fpc (SRX Series)

```
user@host> request chassis fpc online slot 0
```

```
FPC 0 already online
```

request chassis fpc (PTX10008 Router)

```
user@host> request chassis fpc online slot 1
```

```
FPC 0 already online
```

request vmhost cleanup

Syntax

```
request vmhost cleanup
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Clean up temporary files, crash generated files, and log files located in the **/var/tmp**, **/var/crash**, and **/var/log** directories respectively on the host OS.

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

request vmhost file-copy

Syntax

request vmhost file-copy (crash|log) from-jnode *host file-name* to-vjunos *host file-name*

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Copy crash files or log files from the host OS to Junos OS. You can use these files for analysis and debugging purposes.

Options

- crash—Files in /var/crash on the host.
- from-jnode *filename*—Name of the host file to be copied.
- log—Files in **/var/log** on the host.
- to-vjunos *filename*—Name of the Junos OS file to which the host file is copied.

Additional Information

You can use the **show vmhost crash** and **show vmhost logs** commands to list or identify the files in the host OS to be copied to Junos OS.

Required Privilege Level

maintenance

List of Sample Output

request vmhost file-copy on page 202

Sample Output

request vmhost file-copy

`user@host>` **request vmhost file-copy log from-jnode *daemon.log* to-vjunos */var/tmp***

```
:/var/tmp # ls -lrt daemon.log
-rw-r--r--  1 root  wheel  1035126 Mar  4 20:33 daemon.log
```

request vmhost halt

Syntax

```
request vmhost halt
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Stop the host OS and Junos OS running on the device.

Required Privilege Level

maintenance

List of Sample Output

request vmhost halt on page 203

Sample Output

request vmhost halt

```
user@host> request vmhost halt

Halt the vmhost ? [yes,no] (no) yes

Initiating vmhost halt... ok
Initiating Junos shutdown...  shutdown: [pid 8782]
Shutdown NOW!
ok
Junos shutdown is in progress...

*** FINAL System shutdown message from root@ ***

System going down IMMEDIATELY



...
...
```

```
Operating System halted
Please press any key to reboot
```

request vmhost mode

Syntax

```
request vmhost mode [compute | hybrid | throughput]
```

Release Information

Command introduced in Junos OS Release 19.1R1 for NFX150 and NFX250 (NG) devices.

Description

Select the operational mode of the device.

NOTE:

- Starting from Junos OS Release 19.3R1, if the same physical CPU is used for both VNFs and the Junos OS or device components, the request to change the mode fails and an error message is displayed. For example:

  `root>` **request vmhost mode throughput**

  ```
  error: Mode cannot be changed; Reason: Reserved CPUs conflict with VNF
  cpu pinnings: 3
  ```

- When you upgrade the software image that has a VNF CPU conflict to Junos OS Release 19.3R1 by using the CLI upgrade option, the upgrade succeeds and the VNF configuration is applied. The VNF CPU conflict is reported by JDM only if you issue a **commit** command. You must modify the VNF configurations accordingly.

Required Privilege Level

maintenance

List of Sample Output

request vmhost mode compute on page 205

Sample Output

request vmhost mode compute

`user@host>` **request vmhost mode compute**

```
warning: Device will be rebooted to change the mode from hybrid to compute
Do you want to continue? [yes,no] (no)
```

request vmhost power-off

Syntax

```
request vmhost power-off
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

> NOTE: **request vmhost power-on** is not supported on NFX150 and NFX250 (NG) devices.

Description

Shut down the Junos OS software and the host OS.

Required Privilege Level

maintenance

List of Sample Output

request vmhost power-off on page 207

Sample Output

request vmhost power-off

`user@host>` **request vmhost power-off**

```
Power-off the vmhost ? [yes,no] (no) yes

Initiating vmhost shutdown...  ok
Initiating Junos shutdown...  shutdown: [pid 3884]
Shutdown NOW!
ok

*** FINAL System shutdown message from root@host ***

System going down IMMEDIATELY
...
...
```

request vmhost reboot

Syntax

```
request vmhost reboot [disk1 | disk2] [primary | alternate]
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
Command introduced in Junos OS Release 19.4R1 for NFX350 devices.

Description

Reboot the Junos OS software and the host OS from the specified disk and the partition within the disk.

Required Privilege Level

maintenance

List of Sample Output

Sample Output

request vmhost reboot (NFX150)

```
user@host> request vmhost reboot disk1 primary

Reboot the vmhost ? [yes,no] (no) yes

Switching boot to disk1 primary
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
Initiating Junos shutdown...  shutdown: [pid 12151]
Shutdown NOW!
```

```
user@host> request vmhost reboot disk1 alternate

Reboot the vmhost ? [yes,no] (no) yes
```

```
Switching boot to disk1 alternate
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
Initiating Junos shutdown...  shutdown: [pid 16368]
Shutdown NOW!
```

request vmhost reboot (NFX250 NextGen)

`user@host>` **request vmhost reboot disk1 primary**

```
Reboot the vmhost ? [yes,no] (no) yes

Switching boot to disk1 primary
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
Initiating Junos shutdown...  shutdown: [pid 52663]
Shutdown NOW!
```

`user@host>` **request vmhost reboot disk1 alternate**

```
Reboot the vmhost ? [yes,no] (no) yes

Switching boot to disk1 alternate
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
Initiating Junos shutdown...  shutdown: [pid 18763]
Shutdown NOW!
```

request vmhost reboot (NFX350)

`user@host>` **request vmhost reboot disk1 primary**

```
Reboot the vmhost ? [yes,no] (no) yes

Switching boot to disk1 primary
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
```

```
Initiating Junos shutdown...  shutdown: [pid 15575]
Shutdown NOW!
```

user@host> **request vmhost reboot disk1 alternate**

```
Reboot the vmhost ? [yes,no] (no) yes

Switching boot to disk1 alternate
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
Initiating Junos shutdown...  shutdown: [pid 14189]
Shutdown NOW!
```

user@host> **request vmhost reboot disk2 primary**

```
Reboot the vmhost ? [yes,no] (no) yes

Switching boot to disk2 primary
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
Initiating Junos shutdown...  shutdown: [pid 12956]
Shutdown NOW!
```

user@host> **request vmhost reboot disk2 alternate**

```
Reboot the vmhost ? [yes,no] (no) yes

Switching boot to disk2 alternate
Initiating vmhost reboot...  ok
Stopping jrestartd: [  OK  ]
/etc/init.d/functions: line 286: usleep: command not found
Initiating Junos shutdown...  shutdown: [pid 13025]
Shutdown NOW!
```

request vmhost storage

Syntax

```
request vmhost storage
request vmhost storage external-ssd initialize slot [0 | 1] public-dir-name [public-disk0 | public-disk1] force
request vmhost storage external-ssd [add | remove] slot [0 | 1]
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
Command introduced in Junos OS Release 19.4R1 for NFX350 devices.

Description

Initializes an SSD in the specified external SSD slot. It prompts you to confirm and then formats the external SSD so that you can use it for NFX350 device.

Adds or removes an external SSD from its slot. This command also checks the configuration for any VNF path that requires the external disk to be present.

NOTE: External SSDs are not supported on NFX150 and NFX250 devices.

Options

- initialize—Initializes an SSD in the specified external SSD slot 0 or slot 1.
- public-dir-name—Shows the same public-directory path for an SSD even if you move the SSD from one slot to another.
- add—Adds an external SSD to slot 0 or slot 1.
- remove—Removes an external SSD from slot 0 or slot 1.

Required Privilege Level

maintenance

List of Sample Output

request vmhost storage (NFX150) on page 212
request vmhost storage (NFX250 NextGen) on page 212
request vmhost storage (NFX350) on page 212

Sample Output

request vmhost storage (NFX150)

```
user@host> request vmhost storage ?

Possible completions:
  <storage-name>       Storage  name
  self-test-long       Long Storage Test
  self-test-messages   Storage Self Test messages
  self-test-short      Long Storage Test
```

request vmhost storage (NFX250 NextGen)

```
user@host> request vmhost storage

Possible completions:
  <storage-name>       Storage  name
  self-test-long       Long Storage Test
  self-test-messages   Storage Self Test messages
  self-test-short      Long Storage Test
```

request vmhost storage (NFX350)

```
user@host> request vmhost storage external-ssd initialize slot 0 public-dir-name public-disk0 force

Destroy all files on this external SSD and initialize? [yes,no] (no) yes

External SSD in slot 0  initialized, public directory name public-disk0
```

```
user@host> request vmhost storage external-ssd add slot 0

External SSD in slot 0 successfully added, accessible at /var/public-disk0
```

```
user@host> request vmhost storage external-ssd remove slot 0

Remove SSD paths from device? [yes,no] (no) yes

External SSD in slot 0 successfully removed
```

```
user@host> request vmhost storage external-ssd initialize slot 1 public-dir-name public-disk1 force
```

```
Destroy all files on this external SSD and initialize? [yes,no] (no) yes

External SSD in slot 1  initialized, public directory name public-disk1
```

user@host> **request vmhost storage external-ssd add slot 1**

```
External SSD in slot 1 successfully added, accessible at /var/public-disk1
```

user@host> **request vmhost storage external-ssd remove slot 1**

```
Remove SSD paths from device? [yes,no] (no) yes

External SSD in slot 1 successfully removed
```

request vmhost software add

Syntax

```
request vmhost software add package-name <in>| <no-validate>| <reboot>| <set>| <unlink>| <upgrade-to-model
  model-number>
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Install or upgrade the Junos OS and host software packages on the device.

Options

- in—(Optional) Number of minutes to delay before the reboot operation.
- no-validate—(Optional) When loading a software package or bundle with a different release, suppress the default behavior of the **validate** option.
- reboot—(Optional) After adding the software package or bundle, reboot the system.
- set—(Optional) List of URLs or pathnames corresponding to the software packages.
- unlink—(Optional) Removes the software package after successful installation.
- upgrade-to-model—(Optional) *model number*—(Optional) Name of the model to upgrade to.

Required Privilege Level

maintenance

List of Sample Output

request vmhost software add (NFX150) on page 214
request vmhost software add (NFX250 (NG)) on page 215

Sample Output

request vmhost software add (NFX150)

```
user@host> request vmhost software add
/var/public/jinstall-host-nfx-3-x86-64-18.1R1.8-secure-signed.tgz no-validate reboot
```

```
Verified jinstall-host-nfx-3-x86-64-18.1R1.8-secure-signed signed by
PackageProductionEc_2018 method ECDSA256+SHA256
```

```
Pushing Junos image package to the host...
File already present in Host. Skipping pushing the image
Mounting primary partitions to stage upgrade operation
Installing
/mnt/.share/lshare/public/pkginst.7565/install-media-nfx-3-junos-18.1R1.8-secure.tgz
Extracting the package ...
..
..
```

request vmhost software add (NFX250 (NG))

`user@host>` **request vmhost software add**
/var/public/jinstall-host-nfx-3-x86-64-18.4R1.8-secure-signed.tgz

```
Verified jinstall-host-nfx-3-x86-64-18.4R1.8-secure-signed signed by
PackageProductionEc_2018 method ECDSA256+SHA256
Pushing Junos image package to the host...
File already present in Host. Skipping pushing the image
Mounting alternate partitions to stage upgrade operation
Installing
/mnt/.share/lshare/public/pkginst.39634/install-media-nfx-3-junos-18.4R1.8-secure.tgz
Extracting the package ...
============================================
Host OS upgrade is FORCED
Current Host kernel version : 4.1.27-rt30-WR8.0.0.25_ovp
Package Host kernel version : 4.1.27-rt30-WR8.0.0.25_ovp
Current Host version        : 3.0.3
Package Host version        : 3.0.3
Min host version required for applications: 3.0.2
============================================
Validate linux image...
upgrade_platform: -------------------
upgrade_platform: Parameters passed:
upgrade_platform: silent=0
upgrade_platform:
package=/var/tmp/tmp.rV7S1sxWedjunos_cli_upg/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
upgrade_platform: clean install=0
upgrade_platform: on primary   =0
upgrade_platform: clean upgrade=0
upgrade_platform: Need reboot after staging=1
upgrade_platform: -------------------
upgrade_platform:
upgrade_platform: Checking input
```

```
/var/tmp/tmp.rV7S1sxWedjunos_cli_upg/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
 ...
upgrade_platform: Input package
/var/tmp/tmp.rV7S1sxWedjunos_cli_upg/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
 is valid.
Secure Boot is enforced.
ALLOW:usr/secureboot/grub/BOOTX64.EFI
ALLOW:boot/bzImage-intel-x86-64.bin
ALLOW:boot/initramfs.cpio.gz
Setting up Junos host applications for installation ...
Current junos instance is 0
Installing Host OS ...
upgrade_platform: --------------------
upgrade_platform: Parameters passed:
upgrade_platform: silent=0
upgrade_platform: package=/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
upgrade_platform: clean install=0
upgrade_platform: on primary   =0
upgrade_platform: clean upgrade=0
upgrade_platform: Need reboot after staging=0
upgrade_platform: --------------------
upgrade_platform:
upgrade_platform: Checking input
/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz ...
upgrade_platform: Input package
/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz is valid.
Secure Boot is enforced.
ALLOW:usr/secureboot/grub/BOOTX64.EFI
ALLOW:boot/bzImage-intel-x86-64.bin
ALLOW:boot/initramfs.cpio.gz
upgrade_platform: Backing up boot assets..
upgrade_platform: Staging the upgrade package -
/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz..
upgrade_platform: Checksum verified and OK...
upgrade_platform: Staging of /var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
 completed
upgrade_platform: System needs *REBOOT* to complete the upgrade
Host OS upgrade staged. Reboot the system to complete installation!
```

show chassis cluster control-plane statistics

Syntax

```
show chassis cluster control-plane statistics
```

Release Information

Command introduced in Junos OS Release 9.3. Output changed to support dual control ports in Junos OS Release 10.0.

Description

Display information about chassis cluster control plane statistics.

Required Privilege Level

view

RELATED DOCUMENTATION

clear chassis cluster control-plane statistics

List of Sample Output

show chassis cluster control-plane statistics on page 218
show chassis cluster control-plane statistics (SRX5000 Line Devices) on page 219

Output Fields

Table 16 on page 217 lists the output fields for the **show chassis cluster control-plane statistics** command. Output fields are listed in the approximate order in which they appear.

Table 16: show chassis cluster control-plane statistics Output Fields

Field Name	Field Description
Control link statistics	Statistics of the control link used by chassis cluster traffic. Statistics for **Control link 1** are displayed when you use dual control links (SRX5600 and SRX5800 devices only). • **Heartbeat packets sent**—Number of heartbeat messages sent on the control link. • **Heartbeat packets received**—Number of heartbeat messages received on the control link. • **Heartbeat packet errors**—Number of heartbeat packets received with errors on the control link.

Table 16: show chassis cluster control-plane statistics Output Fields *(continued)*

Field Name	Field Description
Fabric link statistics	Statistics of the fabric link used by chassis cluster traffic. Statistics for **Child Link 1** are displayed when you use dual fabric links. • **Probes sent**—Number of probes sent on the fabric link. • **Probes received**—Number of probes received on the fabric link.
Switch fabric link statistics	Statistics of the switch fabric link used by chassis cluster traffic. • **Probe state**—State of the probe, **UP** or **DOWN**. • **Probes sent**—Number of probes sent. • **Probes received**—Number of probes received. • **Probe recv error** —Error in receiving probe. • **Probe send error**—Error in sending probe.

Sample Output

show chassis cluster control-plane statistics

`user@host>` **show chassis cluster control-plane statistics**

```
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 11646
        Heartbeat packets received: 8343
        Heartbeat packet errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 11644
        Probes received: 8266
   Switch fabric link statistics:
    Probe state : DOWN
    Probes sent: 8145
    Probes received: 8013
    Probe recv errors: 0
    Probe send errors: 0
```

Sample Output

show chassis cluster control-plane statistics (SRX5000 Line Devices)

`user@host>` **show chassis cluster control-plane statistics**

```
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 2061982
        Heartbeat packets received: 2060367
        Heartbeat packet errors: 0
   Control link 1:
        Heartbeat packets sent: 2061982
        Heartbeat packets received: 0
        Heartbeat packet errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 3844342
        Probes received: 3843841
    Child link 1
        Probes sent: 0
        Probes received: 0
```

show chassis cluster data-plane interfaces

Syntax

```
show chassis cluster data-plane interfaces
```

Release Information

Command introduced in Junos OS Release 10.2.

Description

Display the status of the data plane interface (also known as a fabric interface) in a chassis cluster configuration.

Required Privilege Level

view

RELATED DOCUMENTATION

cluster (Chassis)

List of Sample Output

show chassis cluster data-plane interfaces on page 220

Output Fields

Table 17 on page 220 lists the output fields for the **show chassis cluster data-plane interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 17: show chassis cluster data-plane interfaces Output Fields

Field Name	Field Description
fab0/fab1	Name of the logical fabric interface. • **Name**—Name of the physical Ethernet interface. • **Status**—State of the fabric interface: **up** or **down**.

Sample Output

show chassis cluster data-plane interfaces

```
user@host> show chassis cluster data-plane interfaces
```

```
fab0:
    Name                Status
    ge-2/1/9            up
    ge-2/2/5            up
fab1:
    Name                Status
    ge-8/1/9            up
    ge-8/2/5            up
```

show chassis cluster data-plane statistics

Syntax

```
show chassis cluster data-plane statistics
```

Release Information

Command introduced in Junos OS Release 9.3.

Description

Display information about chassis cluster data plane statistics.

Required Privilege Level

view

RELATED DOCUMENTATION

clear chassis cluster data-plane statistics

List of Sample Output

show chassis cluster data-plane statistics on page 223

Output Fields

Table 18 on page 223 lists the output fields for the **show chassis cluster data-plane statistics** command. Output fields are listed in the approximate order in which they appear.

Table 18: show chassis cluster data-plane statistics Output Fields

Field Name	Field Description
Services Synchronized	• **Service name**—Name of the service. • **Rtos sent**—Number of runtime objects (RTOs) sent. • **Rtos received**—Number of RTOs received. • **Translation context**—Messages synchronizing Network Address Translation (NAT) translation context. • **Incoming NAT**—Messages synchronizing incoming Network Address Translation (NAT) service. • **Resource manager**—Messages synchronizing resource manager groups and resources. • **Session create**—Messages synchronizing session creation. • **Session close**—Messages synchronizing session close. • **Session change**—Messages synchronizing session change. • **Gate create**—Messages synchronizing creation of pinholes (temporary openings in the firewall). • **Session ageout refresh request**—Messages synchronizing request session after age-out. • **Session ageout refresh reply**—Messages synchronizing reply session after age-out. • **IPsec VPN**—Messages synchronizing VPN session. • **Firewall user authentication**—Messages synchronizing firewall user authentication session. • **MGCP ALG**—Messages synchronizing MGCP ALG sessions. • **H323 ALG**—Messages synchronizing H.323 ALG sessions. • **SIP ALG**—Messages synchronizing SIP ALG sessions. • **SCCP ALG**—Messages synchronizing SCCP ALG sessions. • **PPTP ALG**—Messages synchronizing PPTP ALG sessions. • **RTSP ALG**—Messages synchronizing RTSP ALG sessions.

Sample Output

show chassis cluster data-plane statistics

```
user@host> show chassis cluster data-plane statistics
```

```
Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          0            0
    Session create                            0            0
    Session close                             0            0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPsec VPN                                 0            0
    Firewall user authentication              0            0
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
    RTSP ALG                                  0            0
```

show chassis cluster information

Syntax

```
show chassis cluster information
```

Release Information

Command introduced in Junos OS Release 12.1X47-D10.

Description

Display chassis cluster messages. The messages indicate each node's health condition and details of the monitored failure.

Required Privilege Level

view

RELATED DOCUMENTATION

show chassis cluster status | 245

List of Sample Output

show chassis cluster information on page 226
show chassis cluster information (Monitoring Abnormal Case) on page 227
show chassis cluster information (Preempt Delay Timer) on page 229

Output Fields

Table 19 on page 225 lists the output fields for the **show chassis cluster information** command. Output fields are listed in the approximate order in which they appear.

Table 19: show chassis cluster information Output Fields

Field Name	Field Description
Node	Node (device) in the chassis cluster (**node0** or **node1**).

Table 19: show chassis cluster information Output Fields *(continued)*

Field Name	Field Description
Redundancy Group Information	• Redundancy Group—ID number (0 - 255) of a redundancy group in the cluster. • Current State—State of the redundancy group: **primary**, **secondary**, **hold**, or **secondary-hold**. • Weight—Relative importance of the redundancy group. • Time—Time when the redundancy group changed the state. • From—State of the redundancy group before the change. • To—State of the redundancy group after the change. • Reason—Reason for the change of state of the redundancy group.
Chassis cluster LED information	• Current LED color—Current color state of the LED. • Last LED change reason—Reason for change of state of the LED.

Sample Output

show chassis cluster information

`user@host>` **show chassis cluster information**

```
node0:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Mar 27 17:44:19 hold           secondary      Hold timer expired
        Mar 27 17:44:27 secondary      primary        Better priority (200/200)

    Redundancy Group 1 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Mar 27 17:44:19 hold           secondary      Hold timer expired
        Mar 27 17:44:27 secondary      primary        Remote yield (0/0)

    Redundancy Group 2 , Current State: secondary, Weight: 255

        Time            From           To             Reason
```

```
        Mar 27 17:44:19 hold           secondary      Hold timer expired
        Mar 27 17:44:27 secondary      primary        Remote yield (0/0)
        Mar 27 17:50:24 primary        secondary-hold Preempt/yield(100/200)
        Mar 27 17:50:25 secondary-hold secondary      Ready to become secondary

Chassis cluster LED information:
    Current LED color: Green
    Last LED change reason: No failures

node1:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Mar 27 17:44:27 hold           secondary      Hold timer expired

    Redundancy Group 1 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Mar 27 17:44:27 hold           secondary      Hold timer expired
        Mar 27 17:50:23 secondary      primary        Remote yield (100/0)
        Mar 27 17:50:24 primary        secondary-hold Preempt/yield(100/200)
        Mar 27 17:50:25 secondary-hold secondary      Ready to become secondary

    Redundancy Group 2 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Mar 27 17:44:27 hold           secondary      Hold timer expired
        Mar 27 17:50:23 secondary      primary        Remote yield (200/0)

Chassis cluster LED information:
    Current LED color: Green
    Last LED change reason: No failures
```

Sample Output

show chassis cluster information (Monitoring Abnormal Case)

`user@host>` **show chassis cluster information**

The following output is specific to monitoring abnormal (unhealthy) case.

```
node0:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:07:38 hold           secondary      Hold timer expired
        Apr  1 11:07:41 secondary      primary        Only node present
        Apr  1 11:29:20 primary        secondary-hold Manual failover
        Apr  1 11:34:20 secondary-hold secondary      Ready to become secondary

    Redundancy Group 1 , Current State: primary, Weight: 0

        Time            From           To             Reason
        Apr  1 11:07:38 hold           secondary      Hold timer expired
        Apr  1 11:07:41 secondary      primary        Only node present

    Redundancy Group 2 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:07:38 hold           secondary      Hold timer expired
        Apr  1 11:07:41 secondary      primary        Only node present

Chassis cluster LED information:
    Current LED color: Amber
    Last LED change reason: Monitored objects are down

Failure Information:

    IP Monitoring Failure Information:
    Redundancy Group 1, Monitoring Status: Failed
    IP Address        Status       Reason
    1.1.1.1           Unreachable  redundancy-group state unknown

node1:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:08:40 hold           secondary      Hold timer expired
```

```
        Apr  1 11:29:20 secondary      primary        Remote is in secondary hold


    Redundancy Group 1 , Current State: secondary, Weight: 0

        Time            From           To             Reason
        Apr  1 11:08:40 hold           secondary      Hold timer expired

    Redundancy Group 2 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:08:40 hold           secondary      Hold timer expired

Chassis cluster LED information:
    Current LED color: Amber
    Last LED change reason: Monitored objects are down

Failure Information:

    IP Monitoring Failure Information:
    Redundancy Group 1, Monitoring Status: Failed
    IP Address        Status       Reason
    1.1.1.1           Unreachable  redundancy-group state unknown
```

Sample Output

show chassis cluster information (Preempt Delay Timer)

`user@host>` **show chassis cluster information**

```
node0:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Aug  4 12:30:02 hold           secondary      Hold timer expired
        Aug  4 12:30:05 secondary      primary        Only node present
        Aug  4 14:19:58 primary        secondary-hold Manual failover
        Aug  4 14:24:58 secondary-hold secondary      Ready to become secondary
```

```
    Redundancy Group 1 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Aug  4 14:07:57 secondary      primary        Remote is in secondary hold

        Aug  4 14:20:23 primary        secondary-hold Monitor failed: IF
        Aug  4 14:20:24 secondary-hold secondary      Ready to become secondary
        Aug  4 14:20:54 secondary      primary        Remote is in secondary hold

        Aug  4 14:21:30 primary        secondary-hold Monitor failed: IF
        Aug  4 14:21:31 secondary-hold secondary      Ready to become secondary

Chassis cluster LED information:
    Current LED color: Green
    Last LED change reason: No failures

node1:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Aug  4 12:33:47 hold           secondary      Hold timer expired
        Aug  4 14:19:57 secondary      primary        Remote is in secondary hold


    Redundancy Group 1 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Aug  4 14:07:56 secondary-hold secondary      Ready to become secondary
        Aug  4 14:20:22 secondary      primary        Remote is in secondary hold

        Aug  4 14:20:37 primary        primary-preempt-hold Preempt (99/101)
        Aug  4 14:20:52 primary-preempt-hold secondary-hold Primary preempt hold
timer e
        Aug  4 14:20:53 secondary-hold secondary      Ready to become secondary
        Aug  4 14:21:28 secondary      primary        Remote yield (99/0)

Chassis cluster LED information:
    Current LED color: Green
    Last LED change reason: No failures
```

show chassis cluster interfaces

Syntax

```
show chassis cluster interfaces
```

Release Information

Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0. Output changed to support control interfaces in Junos OS Release 11.2. Output changed to support redundant pseudo interfaces in Junos OS Release 12.1X44-D10. For SRX5000 line devices, output changed to support the internal security association (SA) option in Junos OS Release 12.1X45-D10. Output changed to support MACsec status on control and fabric interfaces in Junos OS Release 15.1X49-D60. For vSRX, output changed to support the internal security association (SA) option in Junos OS Release 19.4R1.

Description

Display the status of the control interface in a chassis cluster configuration.

Required Privilege Level

view

RELATED DOCUMENTATION

cluster (Chassis)

List of Sample Output

show chassis cluster interfaces (SRX5000 line devices with RE3) on page 233
show chassis cluster interfaces (SRX5000 line devices with RE2) on page 233
show chassis cluster interfaces on page 234
show chassis cluster interfaces(SRX5400, SRX5600, and SRX5800 Devices with SRX5000 line SRX5K-SCB3 [SCB3] with Enhanced Midplanes and SRX5K-MPC3-100G10G [IOC3] or SRX5K-MPC3-40G10G [IOC3]) on page 235
show chassis cluster interfaces (vSRX) on page 236

Output Fields

Table 20 on page 232 lists the output fields for the **show chassis cluster interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 20: show chassis cluster interfaces Output Fields

Field Name	Field Description
Control link status	State of the chassis cluster control interface: **up** or **down**.
Control interfaces	• **Index**—Index number of the chassis cluster control interface. • **Interface**—Name of the chassis cluster control interface. The control interface names differ based on the routing engine. For RE2, the control interfaces are displayed as em0 and em1 and for RE3, the control interfaces are displayed as ixlv0 and igb0. • **Monitored-Status**—Monitored state of the interface: **up** or **down**. • **Internal SA**—State of the internal SA option on the chassis cluster control link: **enabled** or **disabled**. **NOTE:** This field is available only on SRX5000 line devices. • **Security**—State of MACsec on chassis cluster control interfaces.
Fabric link status	State of the fabric interface: **up** or **down**.
Fabric interfaces	• **Name**—Name of the fabric interface. • **Child-interface**—Name of the child fabric interface. • **Status**—State of the interface: **up** or **down**. • **Security**—State of MACsec on chassis cluster fabric interfaces.
Redundant-ethernet Information	• **Name**—Name of the redundant Ethernet interface. • **Status**—State of the interface: **up** or **down**. • **Redundancy-group**—Identification number (1–255) of the redundancy group associated with the redundant Ethernet interface.
Redundant-pseudo-interface Information	• **Name**—Name of the redundant pseudointerface. • **Status**—State of the redundant pseudointerface: **up** or **down**. • **Redundancy-group**—Identification number (1–255) of the redundancy group associated with the redundant pseudointerface.
Interface Monitoring	• **Interface**—Name of the interface to be monitored. • **Weight**—Relative importance of the interface to redundancy group operation. • **Status**—State of the interface: **up** or **down**. • **Redundancy-group**—Identification number of the redundancy group associated with the interface.

Sample Output

show chassis cluster interfaces (SRX5000 line devices with RE3)

user@host> **show chassis cluster interfaces**

```
Control link status: Down

Control interfaces:
    Index   Interface   Monitored-Status   Internal-SA   Security
    0       ixlv0       Down               Enabled       Disabled
    1       igb0        Down               Enabled       Disabled

Fabric link status: Down

Fabric interfaces:
    Name    Child-interface    Status                    Security
                               (Physical/Monitored)
    fab0
    fab0

Redundant-ethernet Information:
    Name         Status      Redundancy-group
    reth0        Down        Not configured
    reth1        Down        Not configured
    reth2        Down        Not configured
    reth3        Down        Not configured
    reth4        Down        Not configured

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    lo0          Up          0
```

Sample Output

show chassis cluster interfaces (SRX5000 line devices with RE2)

user@host> **show chassis cluster interfaces**

```
Control link status: Up
```

```
 Control interfaces:
    Index   Interface   Monitored-Status   Internal-SA   Security
    0       em0         Up                 Disabled      Disabled
    1       em1         Down               Disabled      Disabled
Fabric link status: Up

Fabric interfaces:
    Name    Child-interface    Status                    Security
                               (Physical/Monitored)
    fab0    xe-1/0/3           Up   / Down               Disabled
    fab0
    fab1    xe-7/0/3           Up   / Down               Disabled
    fab1

Redundant-ethernet Information:
    Name         Status      Redundancy-group
    reth0        Up          1
    reth1        Up          2
    reth2        Down        Not configured
    reth3        Down        Not configured
    reth4        Down        Not configured
    reth5        Down        Not configured
    reth6        Down        Not configured
    reth7        Down        Not configured
    reth8        Down        Not configured
    reth9        Down        Not configured
    reth10       Down        Not configured
    reth11       Down        Not configured

Redundant-pseudo-interface Information:
    Name        Status       Redundancy-group
     lo0          Up              1

Interface Monitoring:
    Interface         Weight    Status    Redundancy-group
    ge-0/1/9          100       Up        0
    ge-0/1/9          100       Up
```

Sample Output

show chassis cluster interfaces

user@host> **show chassis cluster interfaces**

The following output is specific to fabric monitoring failure:

```
Control link status: Up

Control interfaces:
    Index   Interface    Monitored-Status   Internal-SA  Security
    0       em0         Up                 Disabled     Disabled

Fabric link status: Down

Fabric interfaces:
    Name    Child-interface    Status       Security
                               (Physical/Monitored)
    fab0    ge-0/0/2           Down / Down     Disabled
    fab0
    fab1    ge-9/0/2           Up   / Up       Disabled
    fab1

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    lo0          Up          0
```

Sample Output

show chassis cluster interfaces
(SRX5400, SRX5600, and SRX5800 Devices with SRX5000 line SRX5K-SCB3 [SCB3] with Enhanced Midplanes and SRX5K-MPC3-100G10G [IOC3] or SRX5K-MPC3-40G10G [IOC3])

`user@host>` **show chassis cluster interfaces**

The following output is specific to SRX5400, SRX5600, and SRX5800 devices in a chassis cluster cluster, when the PICs containing fabric links on the SRX5K-MPC3-40G10G (IOC3) are powered off to turn on alternate PICs. If no alternate fabric links are configured on the PICs that are turned on, RTO synchronous communication between the two nodes stops and the chassis cluster session state will not back up, because the fabric link is missing.

```
Control link status: Up

Control interfaces:
    Index   Interface   Monitored-Status   Internal-SA   Security
```

```
    0       em0         Up                Disabled     Disabled
    1       em1         Down              Disabled     Disabled

Fabric link status: Down

Fabric interfaces:
    Name    Child-interface    Status        Security
                               (Physical/Monitored)
    fab0     <<< fab child missing once PIC off lined    Disabled
    fab0
    fab1    xe-10/2/7          Up   / Down               Disabled
    fab1

Redundant-ethernet Information:
    Name         Status      Redundancy-group
    reth0        Up          Not configured
    reth1        Down        1

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    lo0          Up          0
```

Sample Output

show chassis cluster interfaces (vSRX)

`user@host>` **show chassis cluster interfaces**

The following output is specific to view control link status with internal SAs.

```
Control link status: Up

Control interfaces:
 Index     Interface           Status         Internal SA
 305
      0       em0              Up              enabled
306
      1       em1              Down            enabled
```

show chassis cluster port-peering

Syntax

```
show chassis cluster port-peering
```

Release Information

Command introduced in Junos OS Release 19.1.

Description

Display the status of the L3 back-end interface and the corresponding L2 mapped interface.

Before running this command, you must deploy the chassis-cluster and map the L2 and L3 interfaces:

```
user@host# set groups $node chassis cluster redundant-interface L3 interface
mapping-interface L2 interface
```

Required Privilege Level

view

RELATED DOCUMENTATION

cluster (Chassis)

List of Sample Output

show chassis cluster port-peering on page 238

Output Fields

Table 21 on page 237 lists the output fields for the **show chassis cluster port-peering** command. Output fields are listed in the approximate order in which they appear.

Table 21: show chassis cluster port-peering Output Fields

Field Name	Field Description
Backend L3 Interface	L3 interface of the peer map.
Backend L3 Interface status	Status of the L3 interface.
Mapped L2 Interface	L2 interface of the peer map.
Mapped L2 Interface status	Status of the L2 interface.

Sample Output

show chassis cluster port-peering

user@host> **show chassis cluster port-peering**

```
{primary:node0}
user@host> show chassis cluster port-peering
node0:
--------------------------------------------------------------------------

Port peering interfaces:
    Backend L3                Mapping L2
    Interface  Status        Interface  Status
    ge-1/0/1   Up              ge-0/0/1   Up

node1:
--------------------------------------------------------------------------

Port peering interfaces:
    Backend L3                Mapping L2
    Interface  Status        Interface  Status
    ge-8/0/1   Up             ge-7/0/1   Up
```

show chassis cluster statistics

Syntax

```
show chassis cluster statistics
```

Release Information

Command modified in Junos OS Release 9.0.
Output changed to support dual control ports in Junos OS Release 10.0.

Description

This command displays information about chassis cluster services and interfaces.

Required Privilege Level

view

RELATED DOCUMENTATION

clear chassis cluster statistics

List of Sample Output

show chassis cluster statistics on page 241
show chassis cluster statistics (SRX5000 Line Devices) on page 242
show chassis cluster statistics (SRX5000 Line Devices) on page 243

Output Fields

Table 22 on page 240 lists the output fields for the **show chassis cluster statistics** command. Output fields are listed in the approximate order in which they appear.

Table 22: show chassis cluster statistics Output Fields

Field Name	Field Description
Control link statistics	Statistics of the control link used by chassis cluster traffic. Statistics for **Control link 1** are displayed when you use dual control links (SRX5000 lines only). Note that the output for the SRX5000 lines will always show **Control link 0** and **Control link 1** statistics, even though only one control link is active or working. • **Heartbeat packets sent**—Number of heartbeat messages sent on the control link. • **Heartbeat packets received**—Number of heartbeat messages received on the control link. • **Heartbeat packet errors**—Number of heartbeat packets received with errors on the control link.
Fabric link statistics	Statistics of the fabric link used by chassis cluster traffic. Statistics for **Child Link 1** are displayed when you use dual fabric links. • **Probes sent**—Number of probes sent on the fabric link. • **Probes received**—Number of probes received on the fabric link.

Table 22: show chassis cluster statistics Output Fields *(continued)*

Field Name	Field Description
Services Synchronized	• **Service name**—Name of the service. • **Rtos sent**—Number of runtime objects (RTOs) sent. • **Rtos received**—Number of RTOs received. • **Translation context**—Messages synchronizing Network Address Translation (NAT) translation context. • **Incoming NAT**—Messages synchronizing incoming Network Address Translation (NAT) service. • **Resource manager**—Messages synchronizing resource manager groups and resources. • **Session create**—Messages synchronizing session creation. • **Session close**—Messages synchronizing session close. • **Session change**—Messages synchronizing session change. • **Gate create**—Messages synchronizing creation of pinholes (temporary openings in the firewall). • **Session ageout refresh request**—Messages synchronizing request session after age-out. • **Session ageout refresh reply**—Messages synchronizing reply session after age-out. • **IPsec VPN**—Messages synchronizing VPN session. • **Firewall user authentication**—Messages synchronizing firewall user authentication session. • **MGCP ALG**—Messages synchronizing MGCP ALG sessions. • **H323 ALG**—Messages synchronizing H.323 ALG sessions. • **SIP ALG**—Messages synchronizing SIP ALG sessions. • **SCCP ALG**—Messages synchronizing SCCP ALG sessions. • **PPTP ALG**—Messages synchronizing PPTP ALG sessions. • **RTSP ALG**—Messages synchronizing RTSP ALG sessions. • **MAC address learning**—Messages synchronizing MAC address learning.

Sample Output

show chassis cluster statistics

```
user@host> show chassis cluster statistics
```

```
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 798
        Heartbeat packets received: 784
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 793
        Probes received: 0
Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          0            0
    Session create                            0            0
    Session close                             0            0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPsec VPN                                 0            0
    Firewall user authentication              0            0
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
    RTSP ALG                                  0            0
    MAC address learning                      0            0
```

Sample Output

show chassis cluster statistics (SRX5000 Line Devices)

`user@host>` **show chassis cluster statistics**

```
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
    Control link 1:
```

```
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258681
        Probes received: 258681
    Child link 1
        Probes sent: 258501
        Probes received: 258501
Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          0            0
    Session create                            1            0
    Session close                             1            0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPSec VPN                                 0            0
    Firewall user authentication              0            0
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
    RPC ALG                                   0            0
    RTSP ALG                                  0            0
    RAS ALG                                   0            0
    MAC address learning                      0            0
    GPRS GTP                                  0            0
```

Sample Output

show chassis cluster statistics (SRX5000 Line Devices)

`user@host>` **show chassis cluster statistics**

```
Control link statistics:
    Control link 0:
```

```
        Heartbeat packets sent: 82371
        Heartbeat packets received: 82321
        Heartbeat packets errors: 0
    Control link 1:
        Heartbeat packets sent: 0
        Heartbeat packets received: 0
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258681
        Probes received: 258681
    Child link 1
        Probes sent: 258501
        Probes received: 258501
Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          0            0
    Session create                            1            0
    Session close                             1            0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPSec VPN                                 0            0
    Firewall user authentication              0            0
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
    RPC ALG                                   0            0
    RTSP ALG                                  0            0
    RAS ALG                                   0            0
    MAC address learning                      0            0
    GPRS GTP                                  0            0
```

show chassis cluster status

Syntax

```
show chassis cluster status
<redundancy-group  group-number >
```

Release Information

Support for monitoring failures added in Junos OS Release 12.1X47-D10.

Description

Display the current status of the Chassis Cluster. You can use this command to check the status of chassis cluster nodes, redundancy groups, and failover status.

Options

- none—Display the status of all redundancy groups in the chassis cluster.
- **redundancy-group *group-number***—(Optional) Display the status of the specified redundancy group.

Required Privilege Level

view

RELATED DOCUMENTATION

redundancy-group (Chassis Cluster)

clear chassis cluster failover-count

request chassis cluster failover node | **194**

request chassis cluster failover reset | **198**

List of Sample Output

show chassis cluster status on page 246
show chassis cluster status with preemptive delay on page 247
show chassis cluster status redundancy-group 1 on page 248

Output Fields

Table 23 on page 246 lists the output fields for the **show chassis cluster status** command. Output fields are listed in the approximate order in which they appear.

Table 23: show chassis cluster status Output Fields

Field Name	Field Description
Cluster ID	ID number (1-15) of a cluster is applicable for releases upto Junos OS Release 12.1X45-D10. ID number (1-255) is applicable for Releases 12.1X45-D10 and later. Setting a cluster ID to 0 is equivalent to disabling a cluster.
Redundancy-Group	You can create up to 128 redundancy groups in the chassis cluster.
Node name	Node (device) in the chassis cluster (**node0** or **node1**).
Priority	Assigned priority for the redundancy group on that node.
Status	State of the redundancy group (**Primary**, **Secondary**, **Lost**, or **Unavailable**). • **Primary**—Redundancy group is active and passing traffic. • **Secondary**—Redundancy group is passive and not passing traffic. • **Lost**—Node loses contact with the other node through the control link. Most likely to occur when both nodes are in a cluster and there is a control link failure, one node cannot exchange heartbeats, or when the other node is rebooted. • **Unavailable**—Node has not received a single heartbeat over the control link from the other node since the other node booted up. Most likely to occur when one node boots up before the other node, or if only one node is present in the cluster.
Preempt	• **Yes**: Primary state can be preempted based on priority. • **No**: Change in priority will not preempt the primary state.
Manual failover	• **Yes**: Primary state is set manually through the CLI with the **request chassis cluster failover node** or **request chassis cluster failover redundancy-group** command. This overrides **Priority** and **Preempt**. • **No**: Primary state is not set manually through the CLI.
Monitor-failures	• **None**: Cluster working properly. • **Monitor Failure code**: Cluster is not working properly and the respective failure code is displayed.

Sample Output

show chassis cluster status

```
user@host> show chassis cluster status
```

```
Monitor Failure codes:
  CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  200      primary        no      no       None
node1  1        secondary      no      no       None

Redundancy group: 1 , Failover count: 1
node0  101      primary        no      no       None
node1  1        secondary      no      no       None
```

Sample Output

show chassis cluster status with preemptive delay

`user@host>` **show chassis cluster status**

```
Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures
Redundancy group: 0, Failover count: 1
node0  200      primary        no      no       None
node1  100      secondary      no      no       None
Redundancy group: 1, Failover count: 3
node0 200    primary-preempt-hold    yes   no    None node1  100      secondary
        yes     no       None
```

Sample Output

show chassis cluster status redundancy-group 1

```
user@host> show chassis cluster status redundancy-group 1
```

```
Monitor Failure codes:
    CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 1 , Failover count: 1
node0  101      primary        no      no       None
node1  1        secondary      no      no       None
```

show interfaces

Syntax

show interfaces extensive *interface-name*
show interfaces *interface-name*

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Description

Display details of the interfaces configured on NFX150.

Required Privilege Level

view

Sample Output

show interfaces extensive

`user@host>` **show interfaces extensive heth-0-1**

```
Physical interface: heth-0-1, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Copper, MTU: 9192, Speed: 1Gbps, Duplex:
 Full-duplex, Auto-negotiation: Enabled
  Device flags   : Present Running
  Current address: e8:b6:c2:cb:b8:89, Hardware address: e8:b6:c2:cb:b8:89
  Traffic statistics:
   Input  bytes  :                   0                 N/A bps
   Output bytes  :                   0                 N/A bps
   Input  packets:                   0                 N/A pps
   Output packets:                   0                 N/A pps
  MAC statistics:                    Receive         Transmit
    Total octets                           0                0
    Total packets                          0                0
    Unicast packets                        0                0
    Broadcast packets                      0                0
    Multicast packets                      0                0
    CRC/Align errors                       0              N/A
    FIFO errors                            0                0
    DMA errors                             0                0
    MAC control frames                     0                0
```

```
    MAC pause frames                        0               0
    Oversized frames                        0
  VF statistics:
    VF Number: 0, PCI Address: 0000:06:10:0, Mapped to: ge-1/0/1
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
    VF Number: 1, PCI Address: 0000:06:10:4, Mapped to: ge-1/0/2
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
    VF Number: 2, PCI Address: 0000:06:11:0, Mapped to: None
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
    VF Number: 3, PCI Address: 0000:06:11:4, Mapped to: None
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
```

show interfaces <interface1>

user@host> **show interfaces ge-1/0/1**

```
Physical interface: ge-1/0/1, Enabled, Physical link is Up
  Interface index: 184, SNMP ifIndex: 551
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: e8:b6:c2:cb:b8:5c, Hardware address: e8:b6:c2:cb:b8:5c
  Last flapped   : 2019-09-12 15:43:49 UTC (00:00:03 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
```

```
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled
```

show interfaces <interface1>

user@host> **show interfaces ge-1/0/2**

```
Physical interface: ge-1/0/2, Enabled, Physical link is Up
  Interface index: 185, SNMP ifIndex: 552
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: e8:b6:c2:cb:b8:5d, Hardware address: e8:b6:c2:cb:b8:5d
  Last flapped   : 2019-09-12 15:43:49 UTC (00:00:07 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled
```

show system visibility cpu

Syntax

```
show system visibility cpu
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display details such as per CPU statistics, per CPU usage, and CPU pinning for a Junos OS platform.

Required Privilege Level

view

RELATED DOCUMENTATION

- show system visibility host | 256
- show system visibility memory | 267
- show system visibility network | 270
- show system visibility vnf | 276

List of Sample Output

show system visibility cpu (NFX150) on page 253
show system visibility cpu (NFX250 (NG)) on page 254

Output Fields

Table 24 on page 252 lists the output fields for the **show system visibility cpu** command. Output fields are listed in the approximate order in which they appear.

Table 24: show system visibility cpu Output Fields

Field Name	Field Description
Fields for CPU Statistics	
CPU ID	The CPU ID
User Time	The amount of user time, in seconds.
System Time	The amount of system time, in seconds.

Table 24: show system visibility cpu Output Fields *(continued)*

Field Name	Field Description
Idle Time	The amount of time spent in idle mode, in seconds.
Nice Time	The amount of spent nice time, in seconds.
I/O Wait Time	The amount of time spent waiting for input/output (I/O) operations, in seconds.
Interrupt Service Time	The amount of interrupt service time, in seconds.
Service Time	The amount of service time, in seconds.
Fields for CPU Usages	
CPU ID	The CPU ID
CPU Usage	The percentage of CPU used.
Fields for CPU Pinning Information	
Virtual Machine	The name of the virtual machine.
vCPU	The ID of virtual CPUs used by the virtual machine.
CPU	The ID of CPUs used by the virtual machine.
System Component	The name of the system component.
CPUs	The ID of CPUs used by the system component.

Sample Output

show system visibility cpu (NFX150)

```
user@host> show system visibility cpu
```

```
CPU Statistics (Time in sec)
------------------------------------------------------------------------------
CPU Id User Time System Time Idle Time Nice Time IOWait Time Intr. Service Time
------ --------- ----------- --------- --------- ----------- ------------------
0      26583     40107       105816    0         102         0
```

```
1      53183     64078       56959     0         0          0
2      72        67          171189    0         1          0
3      0         96          171241    0         0          0

CPU Usages
----------------
CPU Id CPU Usage
------ ---------
0      36.399999999999999
1      66.700000000000003
2      0.0
3      0.0

CPU Pinning Information
-----------------------------------
Virtual Machine             vCPU CPU
--------------------------- ---- ---
vjunos0                     0    0

System Component                CPUs
------------------------------- --------
ovs-vswitchd                    1
```

show system visibility cpu (NFX250 (NG))

`user@host>` **show system visibility cpu**

```
CPU Statistics (Time in sec)
-------------------------------------------------------------------------------
CPU Id User Time System Time Idle Time Nice Time IOWait Time Intr. Service Time
------ --------- ----------- --------- --------- ----------- ------------------
0      28568     4549        236916    0         205         0
1      272502    0           48        0         0           0
2      165       45          272268    0         11          0
3      40        9           272470    0         0           0
4      0         0           272494    0         0           0
5      0         0           272550    0         0           0
6      0         0           272552    0         0           0
7      272507    0           47        0         0           0
8      0         0           272552    0         0           0
9      0         0           272553    0         0           0
10     0         0           272553    0         0           0
11     0         0           272547    0         0           0
```

```
CPU Usages
----------------
CPU Id CPU Usage
------ ---------
0      11.9
1      100.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      100.0
8      0.0
9      0.0
10     0.0
11     0.0

CPU Pinning Information
-----------------------------------
Virtual Machine             vCPU CPU
--------------------------- ---- ---
vjunos0                     0    0


System Component                CPUs
------------------------------- --------
ovs-vswitchd                    0, 1, 7
```

show system visibility host

Syntax

```
show system visibility host
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Displays details such as the host uptime, number of tasks, CPU statistics, list of disk partitions, disk usage, disk I/O statistics, list of network interfaces, and per port statistics for a Junos OS platform.

Required Privilege Level

view

RELATED DOCUMENTATION

- show system visibility cpu | 252
- show system visibility memory | 267
- show system visibility network | 270
- show system visibility vnf | 276

List of Sample Output

show system visibility host (NFX150) on page 259
show system visibility host (NFX250 (NG)) on page 263

Output Fields

Table 25 on page 256 lists the output fields for the **show system visibility host** command. Output fields are listed in the approximate order in which they appear.

Table 25: show system visibility host Output Fields

Field Name	Field Description
Field for Host Uptime	
Uptime	The time the host has been operational.
Fields for Host Tasks	

Table 25: show system visibility host Output Fields *(continued)*

Field Name	Field Description
Total	The total number of tasks.
Running	The total number of tasks running.
Sleeping	The total number of tasks in sleeping state.
Stopped	The total number of tasks that are stopped.
Zombie	The total number of zombie processes.
Fields for Host CPU Information	
User Time	The amount of user time, in seconds.
System Time	The amount of system time, in seconds.
Idle Time	The amount of time spent in idle mode, in seconds.
Nice Time	The amount of spent nice time, in seconds.
I/O Wait Time	The amount of time spent waiting for input/output (I/O) operations, in seconds.
Interrupt Service Time	The amount of interrupt service time, in seconds.
Fields for Host Disk Partitions	
Device	The device path.
Mount Point	The mount point of the device path.
File System	The file system type.
Options	Options available for the device path.
Fields for Host Disk Usage Information	
Total	The total amount of disk usage space, in mebibytes (MiB).
Used	The amount of used disk usage space, in mebibytes (MiB).
Free	The amount of free disk usage space, in mebibytes (MiB).

Table 25: show system visibility host Output Fields *(continued)*

Field Name	Field Description
Percentage Used	The percentage of used disk space.
Fields for Host Disk I/O Information	
Read Count	The number of times the disk has been read.
Write Count	The number of times a write operation has happened on the disk.
Read Bytes	The number of bytes used in read operations on the disk.
Write Bytes	The number of bytes used in write operations on the disk.
Read Time	The amount of time the disk has been read, in milliseconds.
Write Time	The amount of time write operations have been performed on the disk, in milliseconds.
Fields for List of Host Interfaces	
Interfaces	The name of the interface.
State	The state of the Host Interface.
MAC	The MAC address of the interface.
Fields for List of Host Port Statistics	
Interface	The name of the interface.
Bytes Sent	The number of bytes sent.
Bytes Received	The number of bytes received.
Packets Sent	The number of packets sent.
Packets Received	The number of packets received.
Errors In	The number of errors in.
Errors Out	The number of errors out.
Drops In	The number of drops in.

Table 25: show system visibility host Output Fields *(continued)*

Field Name	Field Description
Drops Out	The number of drops out.

Sample Output

show system visibility host (NFX150)

`user@host>` **show system visibility host**

```
Host Uptime
-----------
Uptime: 1 day 23:19:41.21000

Host Tasks
----------
Total:    187
Running:  3
Sleeping: 179
Stopped:  0
Zombie:   5

Host CPU Information (Time in sec)
----------------------------------
User Time:              79359
System Time:            0
Idle Time:              502215
I/O Wait Time:          103
Nice Time:              103724
Interrupt Service Time: 0

Host Disk Partitions
-------------------------------------------------------------------------------------------------------
Device                                    Mount Point      File System  Options
----------------------------------------- ---------------- -----------
---------------------------------
/dev/sda2                                 /                ext4
rw,relatime,i_version,data=ordered
/dev/sda1                                 /boot/efi        vfat
rw,noatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro
/dev/sda7                                 /config          ext4
```

```
rw,noatime,data=ordered
/dev/sda8                                 /var/log         ext4
rw,noatime,data=ordered
/dev/sda9                                 /mnt/.share      ext4
rw,noatime,discard,data=ordered
/dev/sda5                                 /junos           ext4
rw,noatime,discard,data=ordered
/dev/loop0                                /var/tmp         ext4
rw,relatime,data=ordered
/dev/loop1                                /mnt/.share/lshare/jnpr/jlog ext4
rw,relatime,data=ordered
/dev/loop0                                /mnt/.share/lshare/jnpr/jtmp ext4
rw,relatime,data=ordered

Host Disk Usage Information
---------------------------
Total (MiB):     1469
Used  (MiB):     948
Free  (MiB):     429
Percentage Used: 64.5

Host Disk I/O Information
-------------------------
 Read Count: 187083
Write Count: 256206
 Read Bytes: 2290787328
Write Bytes: 3331667456
  Read Time: 33977
 Write Time: 258864

Host Interfaces
-----------------------------------------------
Interface          State     MAC
------------------ --------- -----------------
heth-0-1           active    00:00:5e:00:53:8e
heth-0-0           active    00:00:5e:00:53:8d
heth-0-3           active    00:00:5e:00:53:90
heth-0-2           active    00:00:5e:00:53:8f
heth-0-5           inactive  00:00:5e:00:53:92
heth-0-4           inactive  00:00:5e:00:53:91
ctrlbr0            active    00:00:5e:00:53:10
docker0            inactive  00:00:5e:00:53:8c
eth0br             active    00:00:5e:00:53:00
eth1br             inactive  00:00:5e:00:53:67
```

```
l3_h_ge_1_0_0      active    00:00:5e:00:53:6d
l3_h_ltectrl       active    00:00:5e:00:53:f1
l3_h_ltedata       active    00:00:5e:00:53:91
lo                 inactive  00:00:00:00:00:00
lte_crtl0          active    00:00:5e:00:53:91
lte_data0          active    00:00:5e:00:53:fc
ovs-sys-br         inactive  00:00:5e:00:53:4f
ovs-system         inactive  00:00:5e:00:53:1b
sit0               inactive  00:00:00:00
veth00             active    00:00:5e:00:53:79
veth01             active    00:00:5e:00:53:87
veth10             active    00:00:5e:00:53:40
veth11             active    00:00:5e:00:53:65
virbr0             active    00:00:5e:00:53:83
virbr1             active    00:00:5e:00:53:6f

Host Port Statistics
-------------------------------------------------------------------------------------------------
Interface Bytes Sent   Bytes Rcvd   Packets Sent Packets Rcvd Errors In Errors Out
 Drops In Drops Out
--------- ------------ ------------ ------------ ------------ --------- ----------
 -------- ---------
l3_h_ge_1_0_0 11025    648          74           8            0         0
  0        0
veth10    0            11673        0            82           0         0
  12       0
veth11    11673        0            82           0            0         0
  0        0
ovs-system 0           0            0            0            0         0
  0        0
ovs-sys-br 0           0            0            0            0         0
  82       0
vnet0     31080352     10698402     153074       136451       0         0
  0        0
vnet1     858553596    712231555    9325949      10546588     0         0
  0        0
vnet2     735033102    50689829     4956943      180168       0         0
  0        0
vnet3     4428680      602          85168        13           0         0
  0        0
eth0      50689829     1077880063   180168       5551593      0         0
  6146     0
eth1br    0            0            0            0            0         0
  0        0
```

```
lte_data0 0           1648        0           14          0         0
  0        0
lo        96584       96584       1219        1219        0         0
  0        0
lte_crtl0 749623      12570778    22710       22762       0         0
  0        0
virbr0-nic 0          0           0           0           0         0
  0        0
docker0   0           0           0           0           0         0
  0        0
veth01    4558        4743808     53          89402       0         0
  0        0
veth00    4743808     4558        89402       53          0         0
  8        0
dcapi-tap 0           0           0           0           0         0
  0        0
l3_h_ltedata 1648     648         14          8           0         0
  0        0
sit0      0           0           0           0           0         0
  0        0
flowd_h_mgmt 391536979 448871585  5975703     5507199     0         0
  0        0
virbr1    29553905    8096581     137792      128808      0         0
  0        0
virbr0    46365       48232       467         540         0         0
  0        0
l3_h_ltectrl 12570778 818395      22762       22718       0         0
  0        0
jdm-hbme1 4474379     55866       85622       537         0         0
  0        0
jdm-hbme2 813479      1526643     7992        15288       0         0
  0        0
eth0br    0           595875398   0           4835907     0         0
  222      0
ctrlbr0   408483097   256713674   3800585     4571275     0         0
  0        0
heth-0-1  0           5368334     0           89330       0         0
  0        0
heth-0-0  0           5366462     0           89349       0         0
  0        0
heth-0-3  0           5367002     0           89358       0         0
  0        0
heth-0-2  0           5365262     0           89329       0         0
  0        0
```

```
heth-0-5  0           0           0           0           0         0
  0        0
heth-0-4  0           0           0           0           0         0
  0        0
```

show system visibility host (NFX250 (NG))

`user@host>` **show system visibility host**

```
Host Uptime
-----------
Uptime: 3 days 3:47:05.09000

Host Tasks
----------
Total:    198
Running:  1
Sleeping: 194
Stopped:  0
Zombie:   3

Host CPU Information (Time in sec)
----------------------------------
User Time:              574351
System Time:            0
Idle Time:              2692218
I/O Wait Time:          216
Nice Time:              4609
Interrupt Service Time: 0

Host Disk Partitions
-------------------------------------------------------------------------------------------------------
Device                                    Mount Point      File System  Options
----------------------------------------- ---------------- -----------
----------------------------------
/dev/sda2                                 /                ext4
rw,relatime,i_version,data=ordered
/dev/sda1                                 /boot/efi        vfat
rw,noatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro
/dev/sda7                                 /config          ext4
rw,noatime,data=ordered
/dev/sda8                                 /var/log         ext4
rw,noatime,data=ordered
/dev/sda9                                 /mnt/.share      ext4
```

```
rw,noatime,discard,data=ordered
/dev/sda5                                   /junos           ext4
rw,noatime,discard,data=ordered
/dev/loop0                                  /var/tmp         ext4
rw,relatime,data=ordered

Host Disk Usage Information
---------------------------
Total (MiB):     1469
Used  (MiB):     906
Free  (MiB):     470
Percentage Used: 61.7

Host Disk I/O Information
-------------------------
 Read Count: 245805
Write Count: 333782
 Read Bytes: 2967304704
Write Bytes: 6147921408
  Read Time: 34906
 Write Time: 448918

Host Interfaces
-----------------------------------------------
Interface          State     MAC
------------------ --------- -----------------
hsxe0              active    30:7c:5e:4c:78:44
hsxe1              active    30:7c:5e:4c:78:45
ctrlbr0            active    02:00:00:00:00:10
docker0            inactive  02:42:f9:e7:08:5f
eth0br             active    4c:96:14:00:00:00
eth1br             inactive  66:7e:98:6c:9d:a7
l3_h_ge_1_0_0      active    ca:6b:5a:fe:39:2c
lo                 inactive  00:00:00:00:00:00
sit0               inactive  00:00:00:00
virbr0             active    30:7c:5e:4c:78:43
virbr1             active    be:51:f7:ac:03:1b

Host Port Statistics
-------------------------------------------------------------------------------------------------
Interface Bytes Sent   Bytes Rcvd   Packets Sent Packets Rcvd Errors In Errors Out
 Drops In Drops Out
--------- ------------ ------------ ------------ ------------ --------- ----------
 -------- ---------
```

```
l3_h_ge_1_0_0 0        648          0            8            0         0
  0        0
ovs-sys-br 0           0            0            0            0         0
  0        0
vnet0     2573491477   117345734    2448205      1790887      0         0
  0        0
vnet1     670930985    585788796    7585078      8400542      0         0
  0        0
vnet2     454043208    224389433    2873376      416585       0         0
  0        0
vnet3     7129616      9814         137213       231          0         0
  0        0
eth0      224389433    464747548    416585       2889060      0         0
  9829     0
lo        61305        61305        920          920          0         0
  0        0
virbr1    2475291351   90762062     1008399      1774468      0         0
  0        0
irb       0            0            0            0            0         0
  0        0
hsxe1     0            0            0            0            0         0
  0        0
hsxe0     0            0            0            0            0         0
  0        0
docker0   0            0            0            0            0         0
  0        0
dcapi-tap 0            0            0            0            0         0
  0        0
sit0      0            0            0            0            0         0
  0        0
flowd_h_mgmt 387545386 426690199    5662328      5294853      0         0
  0        0
virbr0-nic 0           0            0            0            0         0
  0        0
virbr0    3021873      1067179      4573         6153         0         0
  0        0
jdm-hbme1 1785562      33378        34145        404          0         0
  0        0
jdm-hbme2 41904        72344        321          323          0         0
  0        0
eth0br    0            401858893    0            2755416      0         0
  226      0
ctrlbr0   243770080    159923150    2283092      2738720      0         0
  0        0
```

```
eth1br    0           0           0           0           0         0
  0        0
ovs-netdev 0          0           0           0           0         0
  0        0
```

show system visibility memory

Syntax

```
show system visibility memory
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display the details about virtual memory and shared memory for a Junos OS platform.

Required Privilege Level

view

RELATED DOCUMENTATION

- show system visibility cpu | 252
- show system visibility host | 256
- show system visibility network | 270
- show system visibility vnf | 276

List of Sample Output

show system visibility memory (NFX150) on page 268
show system visibility memory (NFX250 (NG)) on page 268

Output Fields

Table 26 on page 267 lists the output fields for the **show system visibility memory** command. Output fields are listed in the approximate order in which they appear.

Table 26: show system visibility memory Output Fields

Field Name	Field Description
Fields for Memory Information—Virtual Memory	
Total	The total amount of available virtual memory, in kibibytes (KiBs).
Used	The total amount of used virtual memory, in kibibytes (KiBs).
Available	The total amount of available virtual memory, in kibibytes (KiBs).

Table 26: show system visibility memory Output Fields *(continued)*

Field Name	Field Description
Free	The total amount of free virtual memory, in kibibytes (KiBs).
Percent Used	The percentage of buffer virtual memory used.
Fields for Memory Information—Swap Memory	
Total	The total amount of available swap memory, in kibibytes (KiBs).
Used	The total amount of used swap memory, in kibibytes (KiBs).
Free	The total amount of free swap memory, in kibibytes (KiBs).
Percent Used	The percentage of buffer swap memory used.

Sample Output

show system visibility memory (NFX150)

```
user@host> show system visibility memory
```

```
Memory Information
------------------
Virtual Memory:
---------------
Total       (KiB): 7946732
Used        (KiB): 3292908
Available   (KiB): 5844376
Free        (KiB): 4653824
Percent Used     : 26.50
```

show system visibility memory (NFX250 (NG))

```
user@host> show system visibility memory
```

```
Memory Information
------------------
```

```
Virtual Memory:
---------------
Total       (KiB): 15914412
Used        (KiB): 6723092
Available   (KiB): 10250492
Free        (KiB): 9191320
Percent Used     : 35.60

Huge Pages:
------------
Total 1GiB Huge Pages:      2
Free 1GiB Huge Pages:       0
Configured 1GiB Huge Pages: 0
Total 2MiB Huge Pages:      401
Free 2MiB Huge Pages:       1
Configured 2MiB Huge Pages: 0

Hugepages Usage:
-----------------------------------------------------------------------------------------------------
Name                              Type                                 Used 1G
Hugepages  Used 2M Hugepages
--------------------------------- ----------------------------------
------------------ ------------------
srxpfe                            other process                        1
       400
ovs-vswitchd                      other process                        2
       0
```

show system visibility network

Syntax

```
show system visibility network
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
Command introduced in Junos OS Release 19.4R1 for NFX350 devices.

Description

Displays details such as the list of MAC addresses assigned to VNF interfaces, the list of internal IP addresses for VNFs, the list of virtual functions used by VNFs, and the list of VNF interfaces for a Junos OS platform.

Required Privilege Level

view

RELATED DOCUMENTATION

- show system visibility cpu | 252
- show system visibility host | 256
- show system visibility memory | 267
- show system visibility vnf | 276

List of Sample Output

show system visibility network (NFX150) on page 271
show system visibility network (NFX250 (NG)) on page 274

Output Fields

Table 27 on page 270 lists the output fields for the **show system visibility network** command. Output fields are listed in the approximate order in which they appear.

Table 27: show system visibility network Output Fields

Field Name	Field Description
Fields for List of VNF MAC Addresses	
VNF	The name of the VNF.
MAC	The MAC address of the VNF.

Table 27: show system visibility network Output Fields *(continued)*

Field Name	Field Description
Fields for List of VNF Internal IP Addresses	
VNF	The name of the VNF.
IP	The IP address of the VNF.
Fields for List of VNF Virtual Functions	
VNF	The name of the VNF.
PF	The names of the Physical Functions available.
VF	The names of the Virtual Functions available for each Physical Function.
Fields for List of Free Virtual Functions	
PF	The names of the Physical Functions available.
VF	The names of the Virtual Functions available for each Physical Function.
Fields for List of VNF Interfaces	
VNF	The name of the VNF.
Interface	The name of the interface.
Type	The type of interface.
Source	The connectivity source.
Model	The connectivity model.
MAC	The MAC address of the VNF.

Sample Output

show system visibility network (NFX150)

```
user@host> show system visibility network
```

```
VNF MAC Addresses
------------------------------------------------------------
VNF                                     MAC
--------------------------------------- -----------------
centos1_ethdef0                         00:00:5E:00:53:9E
centos1_ethdef1                         00:00:5E:00:53:9F
centos1_eth2                            00:00:5E:00:53:A0
centos1_eth3                            00:00:5E:00:53:A1
centos2_ethdef0                         00:00:5E:00:53:A2
centos2_ethdef1                         00:00:5E:00:53:A3
centos2_eth2                            00:00:5E:00:53:A4
centos2_eth3                            00:00:5E:00:53:A5

VNF Internal IP Addresses
------------------------------------------------------------
VNF                                     IP
--------------------------------------- ---------------
centos1                                 192.0.2.103
centos2                                 192.0.2.102

VNF Virtual Functions
--------------------------------------------------------------------
VNF                                     PF        VF
--------------------------------------- --------- ------------
l3_ge_1_0_4_vfdef0                      heth-0-1  0000:04:10:0
l2_ge_0_0_0_vfdef0                      heth-0-0  0000:04:10:1
l2_ge_0_0_0_vfdef1                      heth-0-0  0000:04:10:5
l2_ge_0_0_0_vfdef2                      heth-0-0  0000:04:11:1
l2_ge_0_0_0_vfdef3                      heth-0-0  0000:04:11:5
l3_ge_1_0_2_vfdef0                      heth-0-5  0000:07:10:0
l2_ge_0_0_2_vfdef0                      heth-0-2  0000:04:10:3
l2_ge_0_0_2_vfdef1                      heth-0-2  0000:04:10:7
l2_ge_0_0_2_vfdef2                      heth-0-2  0000:04:11:3
l2_ge_0_0_2_vfdef3                      heth-0-2  0000:04:11:7
l3_ge_1_0_1_vfdef0                      heth-0-4  0000:07:10:1
l2_ge_0_0_3_vfdef0                      heth-0-3  0000:04:10:2
l2_ge_0_0_3_vfdef1                      heth-0-3  0000:04:10:6
l2_ge_0_0_3_vfdef2                      heth-0-3  0000:04:11:2
l2_ge_0_0_3_vfdef3                      heth-0-3  0000:04:11:6

Free Virtual Functions
----------------------
PF        VF
--------- ------------
```

```
heth-0-1  0000:04:10:4
heth-0-1  0000:04:11:0
heth-0-1  0000:04:11:4
heth-0-5  0000:07:10:2
heth-0-5  0000:07:10:4
heth-0-5  0000:07:10:6
heth-0-4  0000:07:10:3
heth-0-4  0000:07:10:5
heth-0-4  0000:07:10:7

VNF Interfaces
--------------------------------------------------------------------------------------------
VNF             Interface Type   Source             Model      MAC
   VLAN-ID
--------------  ---------------  -----------------  ---------- --------------------
 --------
centos2         centos2_vnet6    network default    virtio     00:00:5e:00:53:a2
   --
centos2         centos2_vnet7    bridge eth0br      virtio     00:00:5e:00:53:a3
   --
centos2         centos2_eth2     bridge ovs-sys-br  virtio     00:00:5e:00:53:a4
   199
centos2         centos2_eth3     bridge custom1     virtio     00:00:5e:00:53:a5
   --
centos1         centos1_vnet4    network default    virtio     00:00:5e:00:53:9e
   --
centos1         centos1_vnet5    bridge eth0br      virtio     00:00:5e:00:53:9f
   --
centos1         centos1_eth2     bridge ovs-sys-br  virtio     00:00:5e:00:53:a0
   100
centos1         centos1_eth3     bridge custom1     virtio     00:00:5e:00:53:a1
   --

OVS Interfaces
----------------- ------
NAME              MTU
----------------- ------
custom1           1500
centos2_eth3      1500
centos1_eth3      1500
veth11            9200
l3_h_ge_1_0_0     9200
veth01            9200
ovs-sys-br        1500
```

```
centos1_eth2      1500
centos2_eth2      1500
```

show system visibility network (NFX250 (NG))

`user@host>` **show system visibility network**

```
VNF Virtual Functions
-----------------------------------------------------------------
VNF                                     PF        VF
--------------------------------------- --------- ------------
System_vfdef0                           hsxe0     0000:03:13:6
System_vfdef0                           hsxe1     0000:03:13:7

Free Virtual Functions
----------------------
PF        VF
--------- ------------
hsxe0     0000:03:10:0
hsxe0     0000:03:10:2
hsxe0     0000:03:10:4
hsxe0     0000:03:10:6
hsxe0     0000:03:11:0
hsxe0     0000:03:11:2
hsxe0     0000:03:11:4
hsxe0     0000:03:11:6
hsxe0     0000:03:12:0
hsxe0     0000:03:12:2
hsxe0     0000:03:12:4
hsxe0     0000:03:12:6
hsxe0     0000:03:13:0
hsxe0     0000:03:13:2
hsxe0     0000:03:13:4
hsxe1     0000:03:10:1
hsxe1     0000:03:10:3
hsxe1     0000:03:10:5
hsxe1     0000:03:10:7
hsxe1     0000:03:11:1
hsxe1     0000:03:11:3
hsxe1     0000:03:11:5
hsxe1     0000:03:11:7
hsxe1     0000:03:12:1
hsxe1     0000:03:12:3
hsxe1     0000:03:12:5
```

```
hsxe1     0000:03:12:7
hsxe1     0000:03:13:1
hsxe1     0000:03:13:3
hsxe1     0000:03:13:5

OVS Interfaces
------------------------
NAME              MTU
----------------- ------
dpdk1             1500
ovs-sys-br        1500
l3_h_ge_1_0_0     1500
dpdk0             1500
```

show system visibility vnf

Syntax

```
show system visibility vnf vnf name
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

If a VNF name is not specified, this command displays the details of all VNFs present in the system. Details include VNF memory usage, CPU statistics, the list of network interfaces, the list of disk files, per disk usage, per port I/O statistics, and media information, which includes details about CD-ROM and USB storage devices.

If a VNF name is specified, this command displays the details of that particular VNF. Details include VNF memory usage, CPU statistics, the list of network interfaces, the list of disk files, per disk usage, per port I/O statistics, and media information, which includes details about CD-ROM and USB storage devices.

Required Privilege Level

view

RELATED DOCUMENTATION

- show system visibility cpu | 252
- show system visibility host | 256
- show system visibility memory | 267
- show system visibility network | 270

List of Sample Output

show system visibility vnf on page 279

Output Fields

Table 28 on page 277 lists the output fields for the **show system visibility vnf** command. Output fields are listed in the approximate order in which they appear.

Table 28: show system visibility vnf Output Fields

Field Name	Field Description
Fields for List of VNFs	
ID	ID of the VNF.
Name	Name of the VNF.
State	State of the VNF.
Fields for VNF Memory Usage	
Name	Name of the VNF.
Maximum Memory	The maximum amount of memory, in kibibytes (KiBs).
Used Memory	The total amount of used memory, in kibibytes (KiBs).
Used 1G Hugepages	The total number of 1G hugepages used.
Used 2M Hugepages	The total number of 2M hugepages used.
Fields for VNF CPU Stats	
Name	Name of the VNF.
CPU Time	The total CPU time, in seconds.
System Time	The amount of system CPU time, in seconds.
User Time	The amount of user CPU time, in seconds.
Fields for List of VNF MAC Addresses	
VNF	Names of the VNFs.
MAC	MAC addresses of the VNFs.
Fields for List of VNF Internal IP Addresses	
VNF	Names of the VNFs.
IP	Internal IP addresses of the VNFs.

Table 28: show system visibility vnf Output Fields *(continued)*

Field Name	Field Description
Fields for List of Virtual Functions per VNF	
VNF	Names of the VNFs.
PF	The names of the Physical Functions available.
VF	The names of the Virtual Functions available for each Physical Function.
Fields for the VNF Interfaces	
VNF	The name of the VNF.
Interface	The name of the interface.
Type	The type of interface.
Source	The connectivity source.
Model	The connectivity model.
MAC	The MAC address of the VNF.
Fields for List of VNF Disk Information	
VNF	The name of the VNF.
Disk	The name of the disk.
File	The path to the disk.
Fields for List of VNF Disk Usage	
VNF	The name of the VNF.
Disk	The name of the disk.
Read Requests	The number of times a read operation has happened on the disk.
Bytes Read	The number of read bytes on the disk.
Write Requests	The number of times a write operation has happened on the disk.

Table 28: show system visibility vnf Output Fields *(continued)*

Field Name	Field Description
Bytes Written	The number of bytes written on the disk.
Fields for List of VNF Port Statistics	
VNF	The name of the VNF.
Port	The name of the port.
Rcvd Bytes	The number of bytes received.
Rcvd Packets	The number of packets received.
Rcvd Error	The number of errors received.
Rcvd Drop	The number of drops received.
Trxd Bytes	The number of bytes transferred.
Trxd Packets	The number of packets transferred.
Trxd Error	The number of errors transferred.
Trxd Drop	The number of drops transferred.

Sample Output

show system visibility vnf

`user@host>` **show system visibility vnf**

```
List of VNFs
------------------------------------------------------------
ID   Name                                 State
---- ------------------------------------ ---------------
5    centos                               Running
VNF Memory Usage
-------------------------------------------------------------------------------------------------------

Name                                 Maximum Memory (KiB)  Used Memory (KiB)
```

```
Used 1G Hugepages  Used 2M Hugepages
------------------------------------ -------------------- ------------------
------------------ -----------------
centos                               2097152              260741             0
                 0
VNF CPU Statistics (Time in ms)
-------------------------------------------------------------------------------

Name                                 CPU Time           System Time  User Time
------------------------------------ ------------------ ------------ ------------

centos                               14029              3650         1540
VNF MAC Addresses
------------------------------------------------------------
VNF                                    MAC
---------------------------------------- -----------------
centos_ethdef0                         E8:B6:C2:CC:66:9B
centos_ethdef1                         E8:B6:C2:CC:66:9C
VNF Internal IP Addresses
------------------------------------------------------------
VNF                                    IP
---------------------------------------- ---------------
centos                                 192.0.2.100
------------------------------------------------------------------------
VNF Virtual Functions
------------------------------------------------------------------------
VNF                                    PF        VF
---------------------------------------- --------- ------------
l2_ge_0_0_0_vfdef0                     heth-0-0  0000:02:10:1
l2_ge_0_0_0_vfdef1                     heth-0-0  0000:02:10:5
l2_ge_0_0_0_vfdef2                     heth-0-0  0000:02:11:1
l2_ge_0_0_0_vfdef3                     heth-0-0  0000:02:11:5
l2_ge_0_0_2_vfdef0                     heth-0-2  0000:02:10:3
l2_ge_0_0_2_vfdef1                     heth-0-2  0000:02:10:7
l2_ge_0_0_2_vfdef2                     heth-0-2  0000:02:11:3
l2_ge_0_0_2_vfdef3                     heth-0-2  0000:02:11:7
l3_ge_1_0_2_vfdef0                     heth-0-5  0000:05:10:0
l2_ge_0_0_1_vfdef0                     heth-0-1  0000:02:10:0
l2_ge_0_0_1_vfdef1                     heth-0-1  0000:02:10:4
l2_ge_0_0_1_vfdef2                     heth-0-1  0000:02:11:0
l2_ge_0_0_1_vfdef3                     heth-0-1  0000:02:11:4
l2_ge_0_0_3_vfdef0                     heth-0-4  0000:05:10:1
l2_ge_0_0_3_vfdef1                     heth-0-4  0000:05:10:3
l2_ge_0_0_3_vfdef2                     heth-0-4  0000:05:10:5
```

```
l2_ge_0_0_3_vfdef3                        heth-0-4  0000:05:10:7
l3_ge_1_0_1_vfdef0                        heth-0-3  0000:02:10:2
VNF Interfaces
-------------------------------------------------------------------------------------------------

VNF                  Interface Type       Source       Model      MAC
   IPv4-address
-------------------- --------- ---------  ------------ ---------- -----------------
 ---------------------
centos              centos_vnet4 network default     virtio     e8:b6:c2:cc:66:9b
 --
centos               centos_vnet5 bridge  eth0br        virtio
e8:b6:c2:cc:66:9c --
VNF Disk Information
-------------------------------------------------------------------------------------------------

VNF                           Disk        File
----------------------------- -----------
-----------------------------------------------------------------
centos                        vda         /var/public/centos-linux-1.img
centos                        hda         /var/public/vnf_config_data_vnf0
VNF Disk Usage
-----------------------------------------------------------------------------
VNF                  Disk      Read Req   Read Bytes   Write Req  Write Bytes
-------------------- --------- ---------- ------------ ---------- ------------
centos               vda       5382       84654592     2068       4372480
centos               hda       15         37068        0          0
VNF Port Statistics
-------------------------------------------------------------------------------------------------------------------
VNF                  Port      Rcvd Bytes   Rcvd Packets Rcvd Error Rcvd Drop Trxd
 Bytes   Trxd Packets Trxd Error Trxd Drop
-------------------- --------- ------------ ------------ ---------- ---------
------------ ------------ ---------- ---------
centos               centos_vnet4 572       11           0          0         850
          7            0          0
centos               centos_vnet5 21729     258          0          395       0
          0            0          0
VNF Media Information
-------------------------------------------------------------------------------------------------

VNF                           Media Disk         File
----------------------------- ----- ------------
-----------------------------------------------------------------
```

```
vnf0                          CDROM hda          /var/public/vnf_config_data_vnf0
```

show vmhost connections

Syntax

```
show vmhost connections
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display the details for the cross-connect connections. The NFX150 and NFX250 (NG) supports VLAN PUSH, POP, and SWAP operations.

Options

name—Display the details of a specific connection.

down—Display the details of connections that are not operational.

up—Display the details of connections that are operational.

up-down—Display the details of both operational and non-operational connections.

Required Privilege Level

view

List of Sample Output

show vmhost connections on page 284

Output Fields

Table 29 on page 283 lists the output fields for the **show vmhost connections** command. Output fields are listed in the approximate order in which they appear.

Table 29: show vmhost connections Output Fields

Field Name	Field Description
Connection	Displays the type of the cross-connect.
Function	Displays the name of the virtual network function.
Interface	Specifies an interface on which the connection is established.
Status	Displays the status of the connection.

Sample Output

show vmhost connections

`user@host>` **show vmhost connections**

```
Connection                  Function                    Interface Vlan  Status
--------------------------------------------------------------------------------

phy_cc                      system                      sxe0     200   up
                            centos1                     eth2      500

push_pop_cc                 centos1                     eth2      none  down
                            centos2                     eth3      none

swap_cc                     centos1                     eth2      300   up
                            centos2                     eth2      400

vlan_cc                     centos1                     eth2      100   up
                            centos2                     eth2      100
```

show vmhost control-plane

Syntax

```
show vmhost control-plane
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display the status of the JCP, JDM, Layer 2 dataplane, Layer 3 dataplane, and LTE.

Required Privilege Level

view

List of Sample Output

show vmhost control-plane on page 285

Sample Output

show vmhost control-plane

`user@host>` **show vmhost control-plane**

```
Vmhost Control Plane Information
--------------------------------
    Name                       | State              | Status
    ------------------------------------------------------------
    Junos Control Plane          RUNNING              OK
    Juniper Device Manager       RUNNING              OK
    Layer 2 Infrastructure       RUNNING              OK
    Layer 3 Infrastructure       RUNNING              OK
    LTE                          RUNNING              OK
```

show vmhost crash

Syntax

```
show vmhost crash
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display host OS crash information.

Required Privilege Level

view

List of Sample Output

show vmhost crash on page 286

Sample Output

show vmhost crash

`user@host>` **show vmhost crash**

```
-rw-r--r-- 1 root root 306773 Mar 22 10:41
local-node.srxpfe.7439.1521715280.core.tgz
-rw-r--r-- 1 root root 307058 Mar 22 10:42
local-node.srxpfe.8184.1521715324.core.tgz
-rw-r--r-- 1 root root 306999 Mar 22 10:42
local-node.srxpfe.8918.1521715357.core.tgz
-rw-r--r-- 1 root root 315121 Apr 18 05:35
localhost.dummy_flowdapp.3037.1524029709.core.tgz
-rw-r--r-- 1 root root 315033 Apr 18 05:17
localhost.dummy_flowdapp.3432.1524028674.core.tgz
-rw-r--r-- 1 root root 315088 Apr 13 18:11
localhost.dummy_flowdapp.3435.1523643106.core.tgz
```

show vmhost forwarding-options analyzer

Syntax

```
show vmhost forwarding-options analyzer analyzer-name
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Displays information about the VNF analyzers that are configured for port mirroring on a Junos OS platform.

Options

analyzer-name—Displays the details of a specific analyzer on the device.

Required Privilege Level

view

List of Sample Output

show vmhost forwarding-options analyzer on page 288

Output Fields

Table 30 on page 287 lists the output fields for the **show vmhost forwarding-options analyzer** command. Output fields are listed in the approximate order in which they appear.

Table 30: show vmhost forwarding-options analyzer Output Fields

Field Name	Field Description
Analyzer name	Displays the name of the analyzer instance.
Egress monitored interfaces	Displays interfaces for which the traffic leaving the interfaces is mirrored.
Output interface	Specifies an interface to which mirrored packets are sent.
Ingress monitored interfaces	Displays interfaces for which the traffic entering the interfaces is mirrored.

Sample Output

show vmhost forwarding-options analyzer

user@host> **show vmhost forwarding-options analyzer**

```
Analyzer name                    : mon1
Egress monitored interfaces      : vnf-name1:eth2
Output interface                 : analyzer1:eth2

Analyzer name                    : mon2
Ingress monitored interfaces     : vnf-name2:eth2
Output interface                 : analyzer1:eth3
```

show vmhost memory

Syntax

```
show vmhost memory
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display the memory information for the host OS.

Required Privilege Level

view

List of Sample Output

show vmhost memory on page 289

Output Fields

Sample Output

show vmhost memory

`user@host>` **show vmhost memory**

```
Memory Controller Information
-----------------------------

Id :MC0
correctable-error                  :0
uncorrectable-error                :0
```

show vmhost mode

Syntax

```
show vmhost mode
```

Release Information

Command introduced in Junos OS Release 19.1R1 for NFX150 and NFX250 (NG) devices.

Description

Display the CPU and memory allocations for various components.

Required Privilege Level

view

List of Sample Output

Sample Output

show vmhost mode (Throughput mode)

```
user@host> show vmhost mode
```

```
Mode:
--------
Current Mode: throughput

CPU Allocations:
Name                              Configured                                Used
-------------------------------------------------------------------------------------------
Junos Control Plane               0                                         0

Juniper Device Manager            0                                         0

LTE                               0                                         -

NFV Backplane Control Path        0                                         0

NFV Backplane Data Path           1,2                                       1,2
```

```
Layer 2 Control Path              0                                   0

Layer 2 Data Path                 3,4                                 3,4

Layer 3 Control Path              0                                   0

Layer 3 Data Path                 5,6,7                               5,6,7


Memory Allocations:
Name                              Configured                           Used
-------------------------------------------------------------------------------------------------
Junos Control Plane (mB)          2048                                1548

NFV Backplane 1G hugepages        1                                   1

NFV Backplane 2M hugepages        -                                   0

Layer 2 1G hugepages              1                                   1

Layer 2 2M hugepages              -                                   0

Layer 3 1G hugepages              1                                   1

Layer 3 2M hugepages              651                                 650
```

Sample Output

show vmhost mode (Hybrid mode)

`user@host>` **show vmhost mode**

```
Mode:
--------
Current Mode: hybrid

CPU Allocations:
Name                              Configured                           Used
-------------------------------------------------------------------------------------------------
Junos Control Plane               0                                   0
```

```
Juniper Device Manager           0                                    0

LTE                              0                                    -

NFV Backplane Control Path       0                                    0

NFV Backplane Data Path          1,2                                  1,2

Layer 2 Control Path             0                                    0

Layer 2 Data Path                3                                    3

Layer 3 Control Path             0                                    0

Layer 3 Data Path                4,5                                  4,5


Memory Allocations:
Name                             Configured                           Used
-------------------------------------------------------------------------------------------------
Junos Control Plane (mB)         2048                                 1548

NFV Backplane 1G hugepages       1                                    1

NFV Backplane 2M hugepages       -                                    0

Layer 2 1G hugepages             1                                    1

Layer 2 2M hugepages             -                                    0

Layer 3 1G hugepages             1                                    1

Layer 3 2M hugepages             651                                  650
```

Sample Output

show vmhost mode (Compute mode)

`user@host>` **show vmhost mode**

```
Mode:
--------
Current Mode: compute

CPU Allocations:
Name                                  Configured                                 Used
-------------------------------------------------------------------------------------------------------------------
Junos Control Plane                   16                                         16,6
Juniper Device Manager                16                                          16
LTE                                   16                                          -
NFV Backplane Control Path            16                                          16
NFV Backplane Data Path               1,2,3                                      1,2,3
Layer 2 Control Path                   -                                          -
Layer 2 Data Path                      -                                          -
Layer 3 Control Path                   0                                          0
Layer 3 Data Path                      4,5                                        4,5
CPUs available for VNFs
6,7,8,9,10,11,12,13,14,15,22,23,24,25,26,27,28,29,30,31 -

CPUs turned off                        17,18,19,20,21                             -

Memory Allocations:
Name                                  Configured                                 Used
-------------------------------------------------------------------------------------------------------------------
Junos Control Plane (mB)              2048                                       2012
NFV Backplane 1G hugepages            12                                          18
NFV Backplane 2M hugepages             -                                          0
Layer 2 1G hugepages                   -                                          -
Layer 2 2M hugepages                   -                                          -
Layer 3 1G hugepages                   6                                          6
Layer 3 2M hugepages                  20481                                      20481
```

show vmhost status

Syntax

```
show vmhost status
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display the virtualization status and status of all the CPUs.

Required Privilege Level

view

List of Sample Output

show vmhost status on page 294

Sample Output

show vmhost status

`user@host>` **show vmhost status**

```
Virtualization status :
------------------------------
kvm_status      : ok
libvirt_status  : ok
qemu_status     : ok

CPU Status [Since Boot Time]:

--------------------------------------------------------------------------------

CPU        %usr   %nice    %sys   %iowait  %irq   %soft  %steal  %guest  %gnice
%idle

Load Avg  : 4.04   0.00     4.74   0.01     0.00   0.01   0.00    0.30    0.00
 90.90
cpu0      : 8.26   0.00     15.91  0.06     0.00   0.06   0.00    2.47    0.00
 73.23
```

```
cpu1      : 24.73  0.00     22.95  0.00     0.00   0.00   0.00    0.00    0.00
 52.32
cpu2      : 0.00   0.00     0.01   0.00     0.00   0.00   0.00    0.02    0.00
 99.97
cpu3      : 0.00   0.00     0.00   0.00     0.00   0.00   0.00    0.00    0.00
 100.00
cpu4      : 0.00   0.00     0.00   0.00     0.00   0.02   0.00    0.00    0.00
 99.98
cpu5      : 0.00   0.00     0.00   0.00     0.00   0.00   0.00    0.00    0.00
 100.00
cpu6      : 0.00   0.00     0.00   0.00     0.00   0.00   0.00    0.00    0.00
 100.00
cpu7      : 0.00   0.00     0.00   0.00     0.00   0.00   0.00    0.00    0.00
 100.00

Device:  tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
------------------------------------------------------------
sda      2.15   7.60         30.04        4057951    16046703
```

show vmhost storage

Syntax

```
show vmhost storage
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
Command introduced in Junos OS Release 19.4R1 for NFX350 devices.

Description

Display the vmhost storage information.

Required Privilege Level

view

List of Sample Output

Sample Output

show vmhost storage (NFX150)

```
user@host> show vmhost storage
```

```
Vmhost Storage Information
---------------------------
Storage Name          : sda
SSD Description       : Internal disk 1
SSD Model Number      : SFSA100GQ1AA4TO-C-LB-216-JUN
SSD Serial Number     : 000060124205B1000099
SSD Firmware Version  : SBR13025

  ID      Storage S.M.A.R.T attribute        Raw value

  1       Raw_Read_Error_Rate                0
  5       Reallocated_Sector_Ct              0
  9       Power_On_Hours                     20792
  12      Power_Cycle_Count                  66
```

```
  160     Uncorrectable_Sector_Count         0
  161     Spare_Blocks                       568
  163     Number_of_Initial_Invalid_Blocks   18
  164     Total_Erase_Count                  163038
  165     Maximum_Erase_Count                160
  166     Minimum_Erase_Count                34
  167     Average_Erase_Count                78
  168     Maximum_Specified_Erase_Count      3000
  169     Power-On_UECC_Count                54
  192     Power-Off_Retract_Count            568
  193     Dynamic_Remaps                     0
  194     Temperature_Celsius                32
  195     Hardware_ECC_Recovered             1345461
  196     Reallocated_Event_Count            0
  198     Offline_Uncorrectable              0
  199     UDMA_CRC_Error_Count               0
  215     TRIM_Count                         71048
  235     Total_Flash_LBAs_Written           289438408
  237     Total_Flash_LBAs_Written_Expanded  0
  241     Total_LBAs_Written                 13595913833
  242     Total_LBAs_Read                    6786635984
  243     Total_Host_LBAs_Written_Expanded   0
  244     Total_Host_LBAs_Read_Expanded      0
  248     SSD_Remaining_Life                 98
  249     Spare_Blocks_Remaining_Life        100
```

Sample Output

show vmhost storage (NFX250 NextGen)

user@host> **show vmhost storage**

```
Vmhost Storage Information
------------------------------
Storage Name          : sda
SSD Description       : Internal disk 1
SSD Model Number      : StorFly VSFBM6CC100G-JUN
SSD Serial Number     : P1T13004007308160267
SSD Firmware Version  : 1130-000

  ID      Storage S.M.A.R.T attribute        Raw value
```

```
  1       Raw_Read_Error_Rate               0
  9       Power_On_Hours                    1
  12      Power_Cycle_Count                 37
  192     Power-Off_Retract_Count           28
  194     Temperature_Celsius               40
  199     UDMA_CRC_Error_Count              0
  160     Uncorrectable_Sector_Count        0
  161     Spare_Blocks                      100
  241     Total_LBAs_Written                30678
  242     Total_LBAs_Read                   7542
  169     Power-On_UECC_Count               100
  248     SSD_Remaining_Life                100
  249     Spare_Blocks_Remaining_Life       100
```

Sample Output

show vmhost storage (NFX350)

`user@host>` **show vmhost storage**

```
Vmhost Storage Information
------------------------------
Storage Name          : sda
SSD Description       : Internal disk 1
SSD Model Number      : SFSA050GM3AA2TO-C-LB-34A-JUN
SSD Serial Number     : 000060154396B1000059
SSD Firmware Version  : SBR12050

  ID      Storage S.M.A.R.T attribute        Raw value

  1       Raw_Read_Error_Rate               1
  5       Reallocated_Sector_Ct             0
  9       Power_On_Hours                    8467
  12      Power_Cycle_Count                 120
  160     Uncorrectable_Sector_Count        0
  161     Spare_Blocks                      277
  163     Number_of_Initial_Invalid_Blocks  15
  164     Total_Erase_Count                 113168
  165     Maximum_Erase_Count               146
  166     Minimum_Erase_Count               47
  167     Average_Erase_Count               108
  168     Maximum_Specified_Erase_Count     3000
```

```
  169     Power-On_UECC_Count               85
  192     Power-Off_Retract_Count           277
  193     Dynamic_Remaps                    0
  194     Temperature_Celsius               42
  195     Hardware_ECC_Recovered            2092
  196     Reallocated_Event_Count           0
  198     Offline_Uncorrectable             0
  199     UDMA_CRC_Error_Count              1
  215     TRIM_Count                        20355
  235     Total_Flash_LBAs_Written          110143092
  237     Total_Flash_LBAs_Written_Expanded 0
  241     Total_LBAs_Written                9943202407
  242     Total_LBAs_Read                   6158124561
  243     Total_Host_LBAs_Written_Expanded  0
  244     Total_Host_LBAs_Read_Expanded     0
  248     SSD_Remaining_Life                97
  249     Spare_Blocks_Remaining_Life       100

Vmhost Storage Information
------------------------------
Storage Name          : sdb
SSD Description       : Public disk 0
SSD Model Number      : SFSA800GM3AA8TO-C-OC-626-JUN
SSD Serial Number     : 000060154239B1000059
SSD Firmware Version  : SBR13056
External SSD State    : INITIALIZED
External SSD Slot     : SSD0
Public Directory Path : /var/public-disk0

  ID      Storage S.M.A.R.T attribute       Raw value

  1       Raw_Read_Error_Rate               0
  5       Reallocated_Sector_Ct             0
  9       Power_On_Hours                    7604
  12      Power_Cycle_Count                 98
  160     Uncorrectable_Sector_Count        0
  161     Spare_Blocks                      1068
  163     Number_of_Initial_Invalid_Blocks  98
  164     Total_Erase_Count                 15715
  165     Maximum_Erase_Count               43
  166     Minimum_Erase_Count               0
  167     Average_Erase_Count               3
  168     Maximum_Specified_Erase_Count     3000
  169     Power-On_UECC_Count               28
```

```
  192     Power-Off_Retract_Count            1068
  193     Dynamic_Remaps                     971
  194     Temperature_Celsius                37
  195     Hardware_ECC_Recovered             18110
  196     Reallocated_Event_Count            0
  198     Offline_Uncorrectable              0
  199     UDMA_CRC_Error_Count               0
  215     TRIM_Count                         343556
  235     Total_Flash_LBAs_Written           81364321
  237     Total_Flash_LBAs_Written_Expanded  0
  241     Total_LBAs_Written                 5041956446
  242     Total_LBAs_Read                    3934034061
  243     Total_Host_LBAs_Written_Expanded   0
  244     Total_Host_LBAs_Read_Expanded      0
  248     SSD_Remaining_Life                 100
  249     Spare_Blocks_Remaining_Life        100

Vmhost Storage Information
------------------------------
Storage Name          : sdc
SSD Description       : Internal disk 2
SSD Model Number      : SFSA050GM3AA2TO-C-LB-34A-JUN
SSD Serial Number     : 000060154396B1000058
SSD Firmware Version  : SBR12050

  ID      Storage S.M.A.R.T attribute        Raw value

  1       Raw_Read_Error_Rate                0
  5       Reallocated_Sector_Ct              0
  9       Power_On_Hours                     8467
  12      Power_Cycle_Count                  122
  160     Uncorrectable_Sector_Count         0
  161     Spare_Blocks                       275
  163     Number_of_Initial_Invalid_Blocks   17
  164     Total_Erase_Count                  7492
  165     Maximum_Erase_Count                19
  166     Minimum_Erase_Count                0
  167     Average_Erase_Count                7
  168     Maximum_Specified_Erase_Count      3000
  169     Power-On_UECC_Count                30
  192     Power-Off_Retract_Count            275
  193     Dynamic_Remaps                     0
  194     Temperature_Celsius                42
  195     Hardware_ECC_Recovered             207
```

```
  196     Reallocated_Event_Count           0
  198     Offline_Uncorrectable             0
  199     UDMA_CRC_Error_Count              0
  215     TRIM_Count                        3843
  235     Total_Flash_LBAs_Written          4950046
  237     Total_Flash_LBAs_Written_Expanded 0
  241     Total_LBAs_Written                532128913
  242     Total_LBAs_Read                   291859128
  243     Total_Host_LBAs_Written_Expanded  0
  244     Total_Host_LBAs_Read_Expanded     0
  248     SSD_Remaining_Life                100
  249     Spare_Blocks_Remaining_Life       100

Vmhost Storage Information
------------------------------
Storage Name          : sdd
SSD Description       : Public disk 1
SSD Model Number      : M.2 (S80) 3MG2-P
SSD Serial Number     : B0021811130190037
SSD Firmware Version  : M271112J
External SSD State    : ADDED
External SSD Slot     : SSD1
Public Directory Path : /var/public-disk1

  ID      Storage S.M.A.R.T attribute       Raw value

  1       Raw_Read_Error_Rate               0
  5       Reallocated_Sector_Ct             0
  9       Power_On_Hours                    8
  12      Power_Cycle_Count                 137
  160     Uncorrectable_Sector_Count        0
  163     Number_of_Initial_Invalid_Blocks  78
  164     Total_Erase_Count                 1001
  165     Maximum_Erase_Count               2
  166     Minimum_Erase_Count               0
  167     Average_Erase_Count               0
  168     Maximum_Specified_Erase_Count     3000
  175     Program_Fail_Count_Chip           0
  176     Erase_Fail_Count_Chip             0
  177     Wear_Leveling_Count               0
  178     Used_Rsvd_Blk_Cnt_Chip            0
  181     Program_Fail_Cnt_Total            0
  182     Erase_Fail_Count_Total            0
  192     Power-Off_Retract_Count           12
```

```
194     Temperature_Celsius              42
195     Hardware_ECC_Recovered           7633
196     Reallocated_Event_Count          0
197     Current_Pending_Sector           0
198     Offline_Uncorrectable            0
199     UDMA_CRC_Error_Count             0
232     Available_Reservd_Space          100
241     Total_LBAs_Written               4154
242     Total_LBAs_Read                  183
245     Unknown_Attribute                8008
248     SSD_Remaining_Life               100
249     Spare_Blocks_Remaining_Life      100
```

show vmhost uptime

Syntax

```
show vmhost uptime
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display the current time and information such as how long the host OS has been running, number of users, average load, and the last reboot reason.

Required Privilege Level

view

List of Sample Output

show vmhost uptime on page 303

Reboot Reason Codes

Vmhost last reboot reason: 0x20—power cycle

Vmhost last reboot reason: 0x04—reset button

Vmhost last reboot reason: 0x01—cold reset

Vmhost last reboot reason: 0x80—hypervisor reboot

Vmhost last reboot reason: 0x40—watchdog reset

Sample Output

show vmhost uptime

`user@host>` **show vmhost uptime**

```
Vmhost Current time: 2020-02-05 10:04:09+00:00
Vmhost Uptime:
      10:04:09 up 7 days, 21:43, 0 users, load average: 1.33, 1.26, 1.19
Vmhost last reboot reason: 0x20
```

In the output message, the **vmhost last reboot reason** field provides the reboot reason code. To understand various reboot reason codes and its description, see “Reboot Reason Codes” on page 303.

show vmhost version

Syntax

```
show vmhost version detail
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
Command introduced in Junos OS Release 19.4R1 for NFX350 devices.

Description

Display host version information including Linux host kernel version and host software version.

Required Privilege Level

view

List of Sample Output

show vmhost version (NFX150) on page 305
show vmhost version (NFX250 NextGen) on page 306
show vmhost version (NFX350) on page 306

Sample Output

show vmhost version (NFX150)

`user@host>` **show vmhost version detail**

```
Partition set     : primary
Software version  : 20.3I-20200601_dev_common.0.0613
                    Host kernel release  : 4.1.27-rt30-WR8.0.0.30_ovp
                    Host kernel version  : #1 SMP Fri Jun  1 22:42:16 IST 2019

Partition set             : primary
Software version          : 20.3I-20200601_dev_common.0.0613
Installed/Upgraded at     : Wed Jun  3 12:58:46 UTC 2020
Status                    : Boot success

Partition set             : alternate
Software version          : 20.3I-20200404_dev_common.0.0613
Installed/Upgraded at     : Fri May 5 05:33:45 UTC 2020
Status                    : Boot success
```

Sample Output

show vmhost version (NFX250 NextGen)

`user@host>` **show vmhost version detail**

```
Partition set     : primary
Software version  : 20.3I-20200518_dev_common.0.2122
                    Host kernel release  : 4.1.27-rt30-WR8.0.0.30_ovp
                    Host kernel version  : #1 SMP Fri Dec 27 22:42:16 IST 2019

Partition set             : primary
Software version          : 20.3I-20200518_dev_common.0.2122
Installed/Upgraded at     : Wed May 20 10:11:27 UTC 2020
Status                    : Boot success

Partition set             : alternate
Software version          : 20.3I-20200601_dev_common.0.0613
Installed/Upgraded at     : Thu Jun  4 12:50:37 UTC 1970
Status                    : Boot success
```

Sample Output

show vmhost version (NFX350)

`user@host>` **show vmhost version detail**

```
Partition set     : alternate
Software version  : 20.3I-20200601_dev_common.0.0613
                    Host kernel release  : 4.1.27-rt30-WR8.0.0.30_ovp
                    Host kernel version  : #1 SMP Fri Dec 27 22:42:16 IST 2019

Partition set             : primary
Software version          : 20.3I-20200527_dev_common.0.1016
Installed/Upgraded at     : Mon Jun  1 20:02:10 UTC 2020
Status                    : Boot success

Partition set             : alternate
Software version          : 20.3I-20200601_dev_common.0.0613
Installed/Upgraded at     : Mon Jun  1 08:17:51 UTC 2020
Status                    : Boot success
```

```
Partition set             : second primary
Software version          : 20.3I-20200527_dev_common.0.1016
Installed/Upgraded at     : Mon May 28 09:05:30 UTC 2020
Status                    : Boot success

Partition set             : second alternate
Software version          : 20.3I-20200527_dev_common.0.1016
Installed/Upgraded at     : Mon May 28 09:05:34 UTC 2020
Status                    : Boot success
```

show vmhost vlans

Syntax

```
show vmhost vlans
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 Network Services Platform.
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Display details about the vmhost VLANs.

Options

vlan-name—Display information for a specified VLAN.

brief | detail | extensive —Display the specified level of output.

instance—Display information for a specified instance.

interface—Name of interface for which the table is displayed.

logical-system—Name of logical system.

Required Privilege Level

view

List of Sample Output

show vmhost vlans on page 309

Output Fields

Table 31 on page 308 describes the output fields for the**show vmhost forwarding-options analyzershow vmhost vlans** **show vmhost vlans** command. Output fields are listed in the approximate order in which they appear.

Table 31: show vmhost vlans Output Fields

Field Name	Field Description
vlan-name	Display information for a specified VLAN
brief	Display brief output
detail	Display detailed output

Table 31: show vmhost vlans Output Fields *(continued)*

Field Name	Field Description
extensive	Display extensive output
instance	Display information for a specified instance
interface	Name of interface for which to display table
logical-system	Name of logical system

Sample Output

show vmhost vlans

```
root@host> show vmhost vlans

Routing instance        VLAN name             Tag          Interfaces
vmhost                  test-1                56
                                                           centos1_eth2.0

----
```