JUNIPER NETWORKS | Engineering Simplicity

How to Configure the NFX250 NextGen

Published
2022-06-13

JUNOS

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

How to Configure the NFX250 NextGen

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 This Guide

Use this guide to perform initial provisioning, configure Junos OS features, chain multiple virtualized network functions, monitor, and manage the NFX250 NextGen devices.

1

CHAPTER

Overview

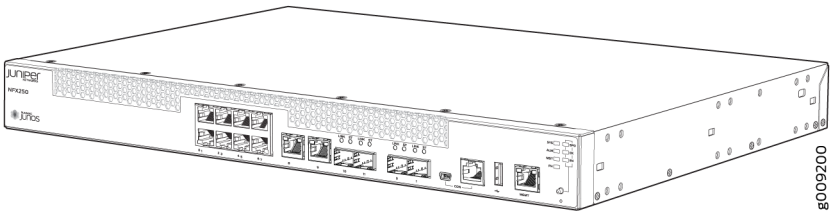
NFX250 NextGen Overview

IN THIS SECTION

The Juniper Networks NFX250 Network Services Platform is a secure, automated, software-driven customer premises equipment (CPE) platform that delivers virtualized network and security services on demand. The NFX250 is part of the Juniper Cloud CPE solution, which leverages Network Functions Virtualization (NFV). It enables service providers to deploy and chain multiple, secure, and high-performance virtualized network functions (VNFs) on a single device.

Figure 1 on page 2 shows the NFX250 device.

Figure 1: NFX250 Device



The NFX250 is a complete SD-WAN CPE, which provides secure router functionality and Next-Generation Firewall (NGFW) solution.

NGFW includes security features such as

- VPN (see VPN User Guide for Security Devices)
- NAT (see NAT User Guide)

- ALG (see Application Layer Gateways User Guide)
- Application Security (see AppSecure User Guide)
- UTM features including Enhanced Web Filtering and Anti-Virus (see UTM User Guide)

The NFX250 device is suitable for small to midsize businesses and large multinational or distributed enterprises.

Junos OS Release 19.1R1 introduces a reoptimized architecture for NFX250 devices. This architecture enables you to use JCP as the single point of management to manage all the NFX250 components.

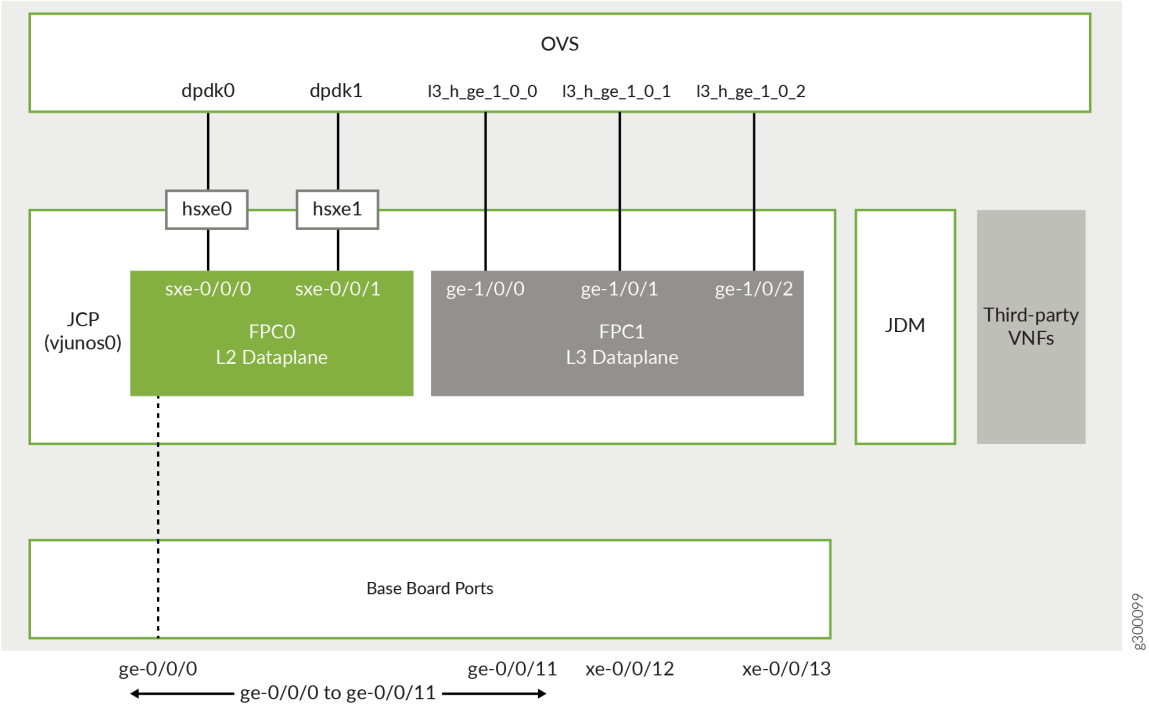
> **NOTE**: For documentation purposes, NFX250 devices that use this architecture are referred to as NFX250 NextGen devices.

Software Architecture

Figure 2 on page 4 illustrates the software architecture of the NFX250 NextGen. The architecture is designed to provide a unified control plane that functions as a single management point. Key

components in the NFX250 NextGen software include the JCP, JDM, Layer 2 data plane, Layer 3 data plane, and VNFs.

Figure 2: NFX250 NextGen Software Architecture



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. In the NFX250 NextGen architecture, 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.
- 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 data plane forwards the LAN traffic to the Open vSwitch (OVS) bridge, which acts as the NFV backplane. The Layer 2 data plane 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.

For the list of supported features, see Feature Explorer.

NFX250 Models

Table 1 on page 5 lists the NFX250 device models and its specifications. For more information, see the *NFX250 Hardware Guide*.

Table 1: NFX250 Models and Specifications

Components	NFX250-S1	NFX250-S2	NFX250-S1E
CPU	2.0 GHz 6-core Intel CPU	2.0 GHz 6-core Intel CPU	2.0 GHz 6-core Intel CPU
RAM	16 GB	32 GB	16 GB
Storage	100 GB SSD	400 GB SSD	200 GB SSD
Form Factor	Desktop	Desktop	Desktop
Ports	Eight 10/100/ 1000BASE-T RJ-45 access ports	Eight 10/100/ 1000BASE-T RJ-45 access ports	Eight 10/100/ 1000BASE-T RJ-45 access ports

Table 1: NFX250 Models and Specifications *(Continued)*

Components	NFX250-S1	NFX250-S2	NFX250-S1E
	Two 10/100/ 1000BASE-T RJ-45 ports which can be used as access ports or uplink ports	Two 10/100/ 1000BASE-T RJ-45 ports which can be used as access ports or uplink ports	Two 10/100/ 1000BASE-T RJ-45 ports which can be used as access ports or uplink ports
	Two 100/1000BASE-X SFP ports which can be used as uplinks	Two 100/1000BASE-X SFP ports which can be used as uplinks	Two 100/1000BASE-X SFP ports which can be used as uplinks
	Two 1-Gigabit or 10-Gigabit Ethernet SFP+ uplink ports	Two 1-Gigabit or 10-Gigabit Ethernet SFP+ uplink ports	Two 1-Gigabit or 10-Gigabit Ethernet SFP+ uplink ports
	One 10/100/ 1000BASE-T RJ-45 management port	One 10/100/ 1000BASE-T RJ-45 management port	One 10/100/ 1000BASE-T RJ-45 management port
	Console ports (RJ-45 and mini-USB)	Console ports (RJ-45 and mini-USB)	Console ports (RJ-45 and mini-USB)
	One USB 2.0 port	One USB 2.0 port	One USB 2.0 port

Interfaces

The NFX250 NextGen device includes the following network interfaces:

- Ten 1-Gigabit Ethernet RJ-45 ports and two 1-Gigabit Ethernet network ports that support small form-factor pluggable (SFP) transceivers. The ports follow the naming convention, ge-0/0/*n*, where *n* ranges from 0 to 11. These ports are used for LAN connectivity.
- Two 1-Gigabit or 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 either 12 or 13. 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.

- Two static interfaces, sxe-0/0/0 and sxe-0/0/1, 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.

NOTE: The NFX250 NextGen devices do not support integrated routing and bridging (IRB) interfaces. The IRB functionality is provided by ge-1/0/0, which is always mapped to the service chaining backplane (OVS). Note that this mapping cannot be changed.

For the list of supported transceivers for your device, see https://apps.juniper.net/hct/product/#prd=NFX250.

Performance Modes

NFX250 NextGen devices offer various operational modes. You can either select the operational mode of the device from a pre-defined list of modes or specify a custom mode.

- Throughput mode—Provides maximum resources (CPU and memory) for Junos software.
- Hybrid mode—Provides a balanced distribution of resources between the Junos software and third-party VNFs.
- Compute mode—Provides minimal resources for Junos software and maximum resources for third-party VNFs.
- Custom mode—Provides an option to allocate resources to Layer 3 data plane and NFV backplane.

NOTE: Compute, hybrid, and throughput modes are supported in Junos OS Release 19.2R1 or later. Custom mode is supported in Junos OS Release 21.1R1 or later.
The default mode is throughput in Junos OS Releases prior to 21.4R1. Starting in Junos OS Release 21.4R1, the default mode is compute.

In throughput mode, you must map SR-IOV VF to Layer 3 data plane interfaces on an NFX250 NextGen device. Three SR-IOV (VFs) are reserved from each NIC (SXE or HSXE) to support a maximum of six Layer 3 data plane 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.

NOTE: Starting in Junos OS Release 21.1R1, mapping OVS to Layer 3 data plane interface is not supported in throughput mode on NFX250 NextGen devices. If the OVS mapping is present in releases prior to Junos OS Release 21.1R1, you must change the mapping before upgrading the device to Junos OS Release 21.1R1 to prevent configuration commit failure.

In hybrid, compute, and throughput modes, you can map Layer 3 data plane interfaces to either SR-IOV or OVS on an NFX250 NextGen device. For example:

Map Layer 3 data plane interfaces to either SR-IOV:

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

Map Layer 3 data plane interfaces to either OVS:

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

NOTE: Starting in Junos OS Release 21.1R1, when your device is in throughput mode, you can map the Layer 3 data plane interfaces only to SR-IOV VFs. When your device is in compute or hybrid modes, you can map the Layer 3 data plane interfaces to either SR-IOV VFs or OVS.

In 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`.

NOTE: When the mapping to a particular Layer 3 data plane interface changes between SR-IOV NICs (eg, hsxe0 to hsxe1) or from hsxe*x* to OVS or vice versa, then FPC1 restarts automatically.

To change the current mode, run the `request vmhost mode` *`mode-name`* command. The `request vmhost mode ?` command lists only the pre-defined modes such as hybrid, compute, and throughput modes.

Before switching to a mode, issue the `show system visibility cpu` and `show vmhost mode` commands to check the availability of CPUs. When switching between operational modes, ensure that resource and configuration conflicts do not occur.

For example, if you move from compute mode that supports VNFs to throughput mode that does not support VNFs, conflicts occur:

```
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 data plane is not mapped to SR-IOV, then switching from hybrid or compute mode to throughput mode results in an error.

You can define a custom mode template in Junos configuration by using the following commands:

1. `user@host#` **set vmhost mode custom *custom-mode-name* layer-3-infrastructure cpu count *count***
2. `user@host#` **set vmhost mode custom *custom-mode-name* layer-3-infrastructure memory size *mem-size***
3. `user@host#` **set vmhost mode custom *custom-mode-name* nfv-back-plane cpu count *count***
4. `user@host#` **set vmhost mode custom *custom-mode-name* nfv-back-plane memory size *mem-size***

Starting in Junos OS Release 22.1R1, you can opt to configure the CPU quota for the Layer 3 data plane by using the `set vmhost mode custom` *`custom-mode-name`* `layer-3-infrastructure cpu colocation quota` *`quota-value`* command, where *quota-value* can range from 1 through 99. If you configure `cpu colocation quota`, then the

sum total of the CPU quotas of the cpu colocation components must be less than or equal to 100. You must configure `cpu count` using numeric values and not keywords like MIN as MIN can have different values for different components.

The number of CPUs and the specific CPUs (by CPU ID) available for VNF usage in a custom mode is automatically determined based on the `cpu count` and `cpu colocation quota` in the custom mode configuration and the internally fixed CPU allocation for other Juniper system components.

The amount of memory, in terms of 1G units, available for VNF usage in a custom mode is automatically determined based on the custom mode specific memory size configuration and the per-SKU internally fixed memory allocation for other Juniper system components. Note that this number is only an approximate value and the actual maximum memory allocation for VNFs might be less than that.

If you do not configure the memory size for a VNF, then the memory is considered as 1G (default value).

CPU count for both NFV backplane and Layer 3 data plane must be configured in integral numbers.

Memory for Layer 3 data plane and NFV backplane must be specified in Gigabytes in a custom mode. The memory specified through a custom mode is created and backed by 1G huge pages for NFV backplane usage and 2M huge pages for Layer 3 data plane usage. It is recommended to configure NFV backplane memory size in integral numbers, whereas Layer 3 data plane memory can be configured in decimals.

You must configure the CPU count and memory for both Layer 3 data plane and NFV backplane. The CPU and memory resources for the remaining Junos software infrastructure is internally determined by the device.

Custom mode template supports a keyword MIN, which is a device-specific pre-defined value for allocating minimal resources.

flex and *perf* are the custom mode templates that are present in the default Junos configuration.

- *flex* mode—Uses MIN keyword for allocating resources to system components such as Layer 3 data plane and NFV backplane. In this mode, device provides maximum memory and CPUs to third-party VNFs.

  To allocate resources in *flex* mode:

  1. `user@host#` **set vmhost mode custom *custom-mode-name* layer-3-infrastructure cpu count MIN**
  2. `user@host#` **set vmhost mode custom *custom-mode-name* layer-3-infrastructure memory size MIN**
  3. `user@host#` **set vmhost mode custom *custom-mode-name* nfv-back-plane cpu count MIN**
  4. `user@host#` **set vmhost mode custom *custom-mode-name* nfv-back-plane memory size MIN**

  In flex mode, you can configure a maximum of:

  - 8 IPSec VPN tunnels

  - 16 IFL
  - 4 IFD
- *perf* mode—Another example custom mode template that is available in the default Junos configuration.

> **NOTE**: Currently, Layer 3 data plane supports only MIN in a custom mode for both CPU count and memory size.

When the device is in custom mode with MIN keyword, only basic firewall features are supported and you can use Layer 3 data plane only for IPsec termination.

When you allocate CPUs to NFV backplane and Layer 3 data plane, the device allocates full cores. When a full core is allocated to NFV backplane, both the logical CPUs on that hyper-threaded core are allocated to it. However, to get the optimal performance, the device disables one of the logical CPUs and is still counted as 2 CPUs allocated. When full cores are not available, the device allocates individual CPUs from different cores.

While allocating CPUs for VNF usage, the device allocates full cores. Both the logical CPUs on that core are enabled. When full cores are not available, the device allocates individual CPUs from different cores.

> **NOTE**: The requested CPU count and memory should not exceed the total CPU count and memory available on the system.

When the device is operating in custom mode, you can make changes to the custom mode configuration. Reboot the device for the changes to take effect.

Commit checks are performed for basic validation when a custom mode is defined in the configuration and when you change the device mode to a custom mode.

You cannot delete a custom mode configuration when the device is operating in the same mode.

To delete a custom mode configuration when the device is operating in custom mode:

1. Change the device mode from custom mode to another mode.
2. Delete the custom mode configuration.

When the device in a custom mode is downgraded to an image that does not support custom mode, then the default throughput mode is applied on the device.

NOTE: Before performing such an image downgrade process, you must remove all VNF configurations from the device.

When multiple custom modes are configured in the device and when the device is in a custom mode other than the *flex* or *perf* custom mode, which are defined in the factory-default Junos configuration, you cannot reset the device configuration to factory-default configuration. Before you reset such a device to factory-default Junos configuration, you must change the device mode to one of the pre-defined modes such as compute, hybrid, throughput, or to the *flex* or *perf* custom mode that are already defined in the factory-default configuration.

Benefits and Uses

The NFX250 NextGen 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.
- 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.

Junos OS Releases Supported on NFX Series Hardware

The Table 2 on page 13 provides details of Junos OS software releases supported on the NFX Series devices.

NOTE: Support for Linux bridge mode on NFX250 devices ended in Junos OS Release 18.4.

NOTE: Support for nfx-2 software architecture on NFX250 devices ended in Junos OS Release 19.1R1.

Table 2: 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
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

Table 2: Supported Junos OS Releases on NFX Series Devices *(Continued)*

NFX Series Platform	Supported Junos OS Release	Software Package	Software Downloads 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

Upgrade the NFX250 Software to NFX250 NextGen Software

IN THIS SECTION

NFX250 NextGen Software Upgrade Overview

Starting in Junos OS Release 19.1R1, the NFX250 devices support the NFX250 NextGen software architecture. This is a re-optimized architecture that enables you to use JCP as the single point of

management to manage all the NFX250 components. For more information about the NFX250 NextGen architecture, see NFX250 NextGen Overview.

NOTE: For documentation purposes, NFX250 devices that use the reoptimized architecture are referred to as NFX250 NextGen devices.

You can upgrade the software using a USB or through a CLI. This topic provides information about prerequisites and the procedure to upgrade through a CLI from NFX250 software architecture to NFX250 NextGen software architecture.

NOTE: The upgrade procedure using a USB remains the same for all NFX Series devices.

Prerequisites

To upgrade an NFX250 device, you must meet the following prerequisites:

Device-specific prerequisites

- An NFX250 device with BIOS => CBDE_SFP_00.21_01.01

  To verify the BIOS version:

  ```
  root@jdm> request execute-command "jhost dmidecode -t bios"
  ```

  For the BIOS information, see the `BIOS Information` section in the command output message.

  If the BIOS version is not CBDE_SFP_00.21_01.01, you can upgrade the BIOS:

  1. Download the BIOS from Downloads page.
  2. Copy and save the BIOS image to the **/var/third-party** directory.
  3. From the JDM CLI, access the hypervisor:

     ```
     root@jdm> ssh hypervisor
     ```

4. Upgrade the BIOS:

```
root@host:~# rpm -ivh /var/third-party/firmware/BIOS RPM package name
```

The system generates the following output:

```
Preparing...             ########################################### [100%]
1:nfx-2-secure-bios      ########################################### [100%]
A reboot is required to install the secure BIOS
Please reboot the system to complete the install
```

5. Reboot the device to load new BIOS.

   a. Exit from hypervisor shell:

```
root@local-node:~# exit
logout
Connection to hypervisor closed.
{master:0}
root@JDM>
```

   b. Reboot the device from JDM CLI.

```
{master:0}
root@porter-p2a-sys1> request system reboot
Reboot the system ? [yes,no] (no) yes
```

- An NFX250 NextGen configuration file with minimal or necessary configurations is required for remote management access to the device after migrating to NFX250 NextGen. This file is an input data for the `request system software add clean-install` *package-name* command.

Release-specific prerequisites

The NFX250 software must be compatible with the following releases:

- NFX250 software running Junos OS Release 18.4R2 or later to accept the configuration by using the command:

```
user@host> request system software add clean-install package-name
```

> **CAUTION**: The `clean-install` command removes all contents on the hard disk. To avoid data loss, copy all important files, configuration files (JDM, JCP, vSRX, and third-party VNFs), log files, and VNF disk or image file, and save them in a secure location before you upgrade the device.

- Releases prior to 18.4R2 must be upgraded to 18.4R2 or later.

> **CAUTION**: The NFX250 device will crash if you upgrade the NFX250 software image running Junos OS Release prior to 18.4R2 to a release that supports NFX250 NextGen software image.
>
> The NFX250 NextGen configuration must be compatible with the NFX250 NextGen software version. The configuration command syntax is not validated.

NOTE: The NFX250 software architecture and NFX250 NextGen software architecture are different and the configurations are different for both the software.

Upgrade to NFX250 NextGen Software Architecture

Before you upgrade the NFX device:

- Create backup of the configuration files (JDM, JCP, vSRX, and third-party VNFs), log files, VNF disk or image file, and other important files stored on the device.
- Check the prerequisites.

To upgrade the NFX250 software architecture to NFX250 NextGen software architecture:

1. Copy the configuration files that are required for in-band and out-of-band management and save it in the **/var/third-party** folder. The configuration file should be of the same format as the file format obtained by running the `show configuration` CLI command.
2. Copy the NFX250 NextGen software image and save it in the **/var/third-party/images** folder.
3. Initiate the software upgrade by using the following command:

```
root@jdm> request system software add clean-install reboot /var/third-party/images/jinstall-image.tgz upgrade-with-config /var/third-party/config-file
```

The device is formatted and the NFX250 NextGen software image is installed. The device loads the configurations and boots up the NFX250 Nextgen software image. You can access the device remotely through the in-band and out-of-band management.

4. The device is now ready for additional configurations and third-party VNF onboarding.

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 3 on page 19.
- 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 3: 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

Table 3: Software Compatibility Details with vSRX and Cloud CPE Solution *(Continued)*

NFX250 Junos OS Release	vSRX	Cloud CPE Solution
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
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 NFX250 NextGen Devices

IN THIS SECTION

Factory Default Settings

The NFX250 NextGen is shipped with the following factory default settings:

Table 4: Security Policies

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

The following table lists the ports in the trust and untrust zones for a device with factory-default configuration.

Table 5: Interfaces

Port Label	Interface	Security Zone	DHCP State	IP Address
0/1 to 0/11	ge-0/0/1 to ge-0/0/11	trust	server	192.168.2.1/24
0/12 to 0/13	xe-0/0/12 to xe-0/0/13	untrust	client	ISP assigned

Table 5: Interfaces *(Continued)*

Port Label	Interface	Security Zone	DHCP State	IP Address
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 (0/12 and 0/13) to the ISP. The device is assigned an IP address by the ISP through DHCP.

   NOTE: For information about NFX250 (NG) interfaces, see Table 5 on page 22.

2. Connect the laptop to one of the front panel LAN ports (0/0 to 0/11). 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.

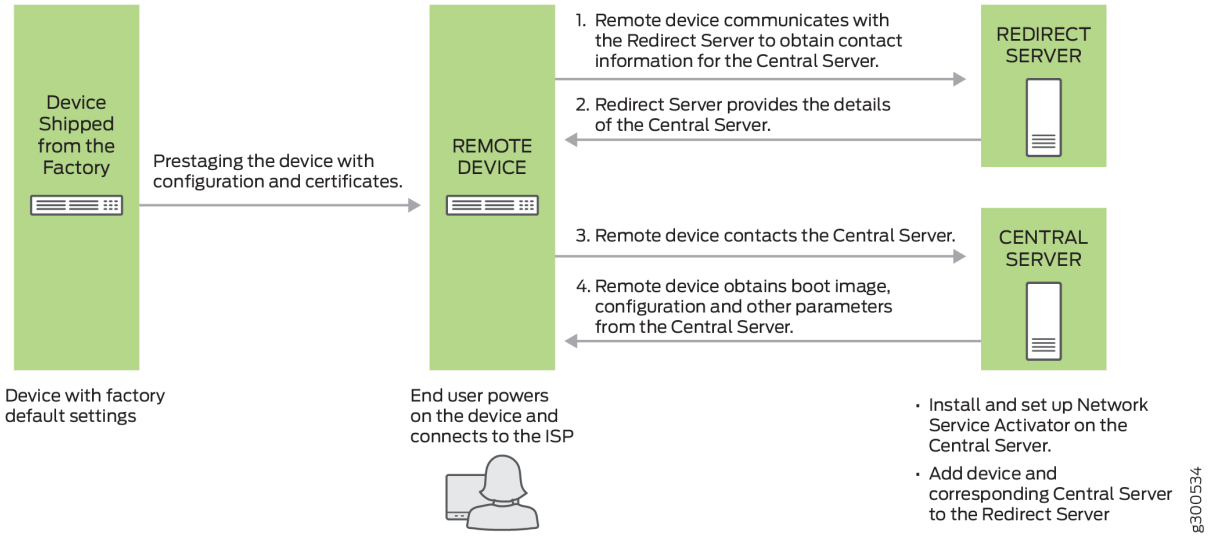
Pre-staging an NFX Series Device

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 3 on page 27 illustrates the workflow of prestaging the NFX Series devices.

Figure 3: 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 " on page 22.
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]
```

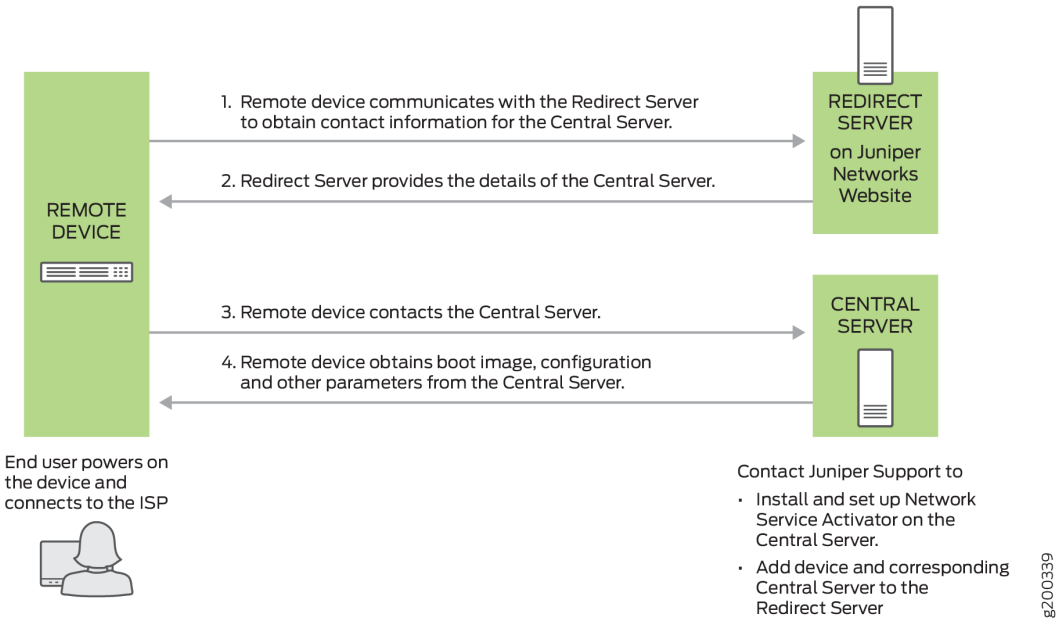
To disable trace operation:

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

Provisioning an NFX Series Device

Figure 4 on page 29 illustrates the workflow of the initial provisioning of NFX Series devices.

Figure 4: 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 4 on page 29 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.

CHAPTER 3

Generating YANG Files

YANG files on NFX250 NextGen Devices

IN THIS SECTION

Understanding YANG on NFX250 NextGen 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 NFX250 NextGen 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.

CHAPTER 4

Configuring Interfaces

Configuring the In-Band Management Interface on NFX250 NextGen Devices

In in-band management, you configure a network interface as a management interface and connect it to the management device. By default. ports ge-1/0/0, ge-1/0/1, and ge-1/0/2 are configured as network interfaces. In addition, you can configure network interfaces from ge-1/0/3 to ge-1/0/9.

To configure in-band management:

1. Log in to the 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. Configure the physical network interface (ge or xe) as a member of the management VLAN:

   ge interface configuration:

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

   Where **x** ranges from 0 to 11.

   xe interface configuration:

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

Where **x** can be 12 or 13.

3. Configure the service interface as a member of the management VLAN:

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

Where **x** can be 0 or 1.

NOTE: You can map ge-1/0/*x* to OVS by using the `set vmhost virtualization-options interfaces ge-1/0/x` command. After you change the mapping, FPC1 restarts automatically.

ADSL2 and ADSL2+ Interfaces on NFX250 NextGen Devices

IN THIS SECTION

ADSL Interface Overview

IN THIS SECTION

Asymmetric digital subscriber line (ADSL) technology is part of the *x*DSL 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 6 on page 38.

Table 6: 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 NFX250 Devices

IN THIS SECTION

Requirements

This example uses the following hardware and software components:

- NFX250 device running the Junos OS Release 19.1R1 version, which supports the reoptimized architecture.

Overview

In this example, you are configuring ADSL SFP interface on an NFX250 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

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure ADSL SFP interfaces on NFX250 NextGen 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 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
```

5. Commit the configuration.

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

Results

VDSL2 Interfaces on NFX250 NextGen 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 5 on page 43.
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 6 on page 44.

> **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 5 on page 43 shows a typical VDSL2 network topology.

Figure 5: Typical VDSL2 End-to-End Connectivity and Topology Diagram

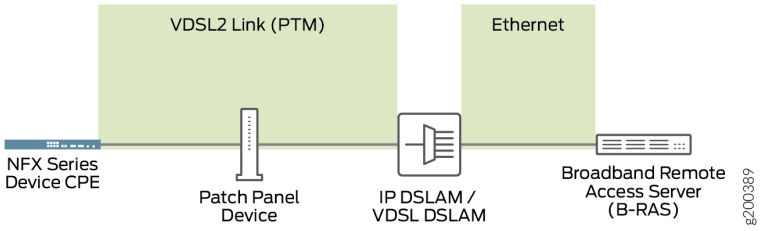
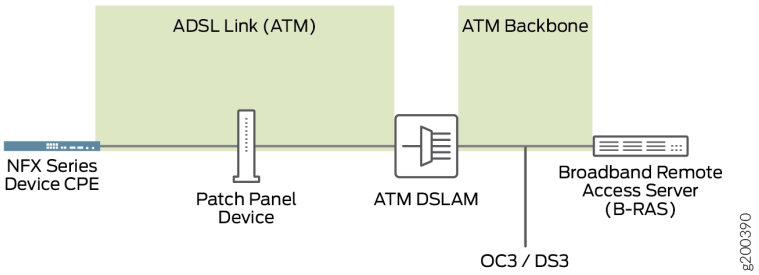


Figure 6 on page 44 shows a backward-compatible ADSL topology using ATM DSLAM.

Figure 6: Backward-Compatible ADSL Topology (ATM DSLAM)



VDSL2 Interface Support on NFX Series Devices

IN THIS SECTION

The VDSL2 interface is supported on the NFX Series devices listed in Table 7 on page 44. (Platform support depends on the Junos OS release in your installation.)

Table 7: 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

Table 7: VDSL2 Annex A and Annex B Features *(Continued)*

Features	POTS
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 8 on page 46 lists the different profiles supported on the VDSL2 interfaces and their properties.

Table 8: 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 NFX250 Devices

IN THIS SECTION

Requirements

This example uses the following hardware and software components:

- NFX250 NextGen device running Junos OS Release 19.1R1.

Overview

In this example, you are configuring VDSL SFP interface on an NFX250 device with the following configurations:

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

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

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure VDSL SFP interfaces on NFX250 NextGen 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

RELATED DOCUMENTATION

NFX250 Overview

JDM Architecture Overview

JDM CLI Overview

Configuring the LTE Module on NFX Devices

IN THIS SECTION

The LTE module can be configured in three modes:

- Always-on—The LTE module connects to the 3G/4G network after booting. The connection is always maintained, as long as there are no network or connectivity problems.

  NOTE: The default mode for LTE module is always-on. For the LTE module to be operational, you only need to install one SIM card on the LTE module before powering on the device. There is no additional configuration required.

- Dial-on-demand—The LTE module initiates a connection when it receives interesting traffic. You define interesting traffic using the dialer filter. To configure dial-on-demand using a dialer filter, you first configure the dialer filter and then apply the filter to the dialer interface.

- Backup—The LTE module connects to the 3G/4G network when the primary connection fails.

You can configure the LTE module either as a primary interface or as a backup interface. When configured as the primary interface, the LTE module supports both the always-on and dial-on-demand modes. When configured as the backup interface, the LTE module connects to the network only when the primary interface fails.

NOTE: Starting in Junos OS Release 19.1R1, you can configure LTE modules on both nodes in a chassis cluster to provide backup WAN support.

Profile configuration is not needed in most scenarios, as LTE has a built-in database of many service providers and can automatically select the profile to use. Occasionally, you might need to specify profiles explicitly in the configuration, in which case, the automatic profile selection is disabled.

Before you begin the configuration, insert the Subscriber Identity Module (SIM) in the LTE module. The SIM uses a profile to establish a connection with the network. You can configure up to 16 profiles for each SIM card. The LTE module supports two SIM cards and so you can configure a total of 32 profiles, although only one profile can be active at a time. To configure the SIM profile, you will require the following information from the service provider:

- Username and password
- Access Point Name (APN)
- Authentication (Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP))

Configuring the LTE Module for Primary Mode

Before you begin the procedure, ensure that the logical interface (dl0.0) is not configured as a backup. If dl0.0 is configured as a backup option for any interface on the device, then this configuration overrides the configuration outlined in this procedure, and the LTE module will function as a backup interface.

Use the `show interfaces | display set | match backup-option | match dl0.0` command to check whether any interface uses dl0.0 as a backup interface. If dl0.0 is configured as a backup interface, then delete the configuration by issuing the following command:

`delete interfaces` *`interface-name`* `unit 0 backup-options interface dl0.0`

To configure the LTE module as a primary interface:

1. Configure the dialer interface:

```
user@host# set interfaces dl0 unit 0 family inet negotiate-address
user@host# set interfaces dl0 unit 0 family inet6 negotiate-address
user@host# set interfaces dl0 unit 0 dialer-options pool dialer-pool-number
user@host# set interfaces dl0 unit 0 dialer-options dial-string dial-number
user@host# set interfaces dl0 unit 0 dialer-options always-on
```

2. Configure the dialer pool for the LTE physical interface:

```
user@host# set interfaces cl-1/1/0 dialer-options pool dialer-pool-number
```

The *`dialer-pool-number`* is always 1 as there is only one LTE interface on the NFX150.

3. Configure the profile.

```
user@host# run request modem wireless create-profile profile-id profile-id cl-1/1/0 slot sim-slot-number
access-point-name apn-name authentication-method none
```

NOTE: `sim-slot-number` is the slot on the module in which the SIM card is inserted.

4. Verify that the profile is configured successfully:

```
user@host# run show modem wireless profiles cl-1/1/0 slot 1
```

5. Activate the SIM card:

```
user@host# set interfaces cl-1/1/0 act-sim sim-slot-number
```

6. Select the profile and configure the radio access type for the SIM card:

```
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number select-profile profile-id profile-
id
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number radio-access automatic
```

NOTE: If a SIM card is installed in the second slot, then select the profile and configure the radio access type for the SIM card in the second slot as well.

7. Verify the status of the wireless network and dialer interface:

```
user@host#  run show modem wireless network
user@host#  run show interfaces dl0.0
```

Configuring the LTE Module for Dial-on-Demand Mode

When the LTE module is configured as a primary interface, it can function either in always-on mode or in dial-on-demand mode. In always-on mode, the interface remains connected to the network whereas In dial-on-demand mode, the connection is established only when needed.

In dial-on-demand mode, the dialer interface is enabled only when network traffic configured as an “interesting traffic” arrives on the network. Interesting traffic triggers or activates the wireless WAN connection. You define an interesting packet by using the dialer filter. To configure dial-on-demand by using a dialer filter, you first configure the dialer filter and then apply the filter to the dialer interface.

Once the traffic is sent over the network, an inactivity timer is triggered and the connection is closed after the timer expires.

NOTE: The dial-on-demand mode is supported only if the LTE module is configured as a primary interface.

To configure the LTE module as a dial-on-demand interface:

1. Configure the dialer interface:

```
user@host# set interfaces dl0 unit 0 family inet negotiate-address
user@host# set interfaces dl0 unit 0 family inet6 negotiate-address
user@host# set interfaces dl0 unit 0 family inet filter dialer dialer-filter-name
user@host# set interfaces dl0 unit 0 dialer-options pool dialer-pool-number
user@host# set interfaces dl0 unit 0 dialer-options dial-string dial-number
```

2. (Optional) Configure the `idle-timeout` value, which determines the duration for which the connection will remain enabled in the absence of interesting traffic.

```
user@host# set interfaces dl0 unit 0 dialer-options idle-timeout idle-timeout-value
```

3. Configure the dialer pool for the LTE physical interface:

```
user@host# set interfaces cl-1/1/0 dialer-options pool dialer-pool-number
```

   The *dialer-pool-number* is always 1 as there is only one LTE interface on the NFX150.

4. Create the dialer filter rule:

```
user@host# set firewall family inet dialer-filter dialer-filter-name term term1 from destination-
address ip-address then note
```

5. Set the default route:

```
user@host# set routing-options static route ip-address next-hop dl0.0
```

6. Configure the profile.

```
user@host# run request modem wireless create-profile profile-id profile-id cl-1/1/0 slot sim-slot-
number access-point-name apn-name authentication-method none
```

NOTE: *sim-slot-number* is the slot on the module in which the SIM card is inserted.

7. Verify that the profile is configured successfully:

```
user@host# run show modem wireless profiles cl-1/1/0 slot 1
```

8. Activate the SIM card:

```
user@host# set interfaces cl-1/1/0 act-sim sim-slot-number
```

9. Select the profile and configure the radio access type for the SIM card:

```
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number select-profile profile-id
profile-id
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number radio-access automatic
```

NOTE: If a SIM card is installed in the second slot, then select the profile and configure the radio access type for the SIM card in the second slot as well.

10. Verify the configuration by sending traffic to the destination address. The traffic is routed to the dl0 interface and if it matches the dialer filter rule, then the dl0 is triggered to dial.
11. Verify the status of the wireless network and dialer interface:

```
user@host#  run show modem wireless network
user@host#  run show interfaces dl0.0
```

Configuring the LTE Module for Backup Mode

You can configure the LTE module as a backup interface. If the primary interface fails, the LTE module connects to the network and remains online only until the primary interface becomes functional. The dialer interface is enabled only when the primary interface fails.

To configure the LTE module as a backup interface:

1. Configure the dialer interface:

```
user@host# set interfaces dl0 unit 0 family inet negotiate-address
user@host# set interfaces dl0 unit 0 family inet6 negotiate-address
user@host# set interfaces dl0 unit 0 dialer-options pool dialer-pool-number
user@host# set interfaces dl0 unit 0 dialer-options dial-string dial-number
```

2. Configure the dialer pool for the LTE physical interface:

```
user@host# set interfaces cl-1/1/0 dialer-options pool dialer-pool-number
```

   The *dialer-pool-number* is always 1 as there is only one LTE interface on the NFX150.

3. Configure the profile.

```
user@host# run request modem wireless create-profile profile-id profile-id cl-1/1/0 slot sim-slot-number
access-point-name l3vpn.corp authentication-method none
```

   NOTE: `sim-slot-number` is the slot on the LTE module in which the SIM card is inserted.

4. Verify that the profile is configured successfully:

```
user@host# run show modem wireless profiles cl-1/1/0 slot 1
```

5. Activate the SIM card:

```
user@host# set interfaces cl-1/1/0 act-sim sim-slot-number
```

6. Select the profile and configure the radio access type for the SIM card:

```
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number select-profile profile-id profile-id
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number radio-access automatic
```

> **NOTE**: If a SIM card is installed in the second slot, then select the profile and configure the radio access type for the SIM card in the second slot as well.

7. Configure the Ethernet interface as the primary interface, which connects to the wireless network. Configure the dl0 interface as the backup interface.

```
user@host# set interfaces ge-1/0/2 unit 0 family inet address 192.168.2.1/24
user@host# set interfaces ge-1/0/2 unit 0 backup-options interface dl0.0
```

8. Verify the status of the wireless network and dialer interface:

```
user@host#  run show modem wireless network
user@host#  run show interfaces dl0.0
```

Configuring the LTE Interface Module in an NFX Chassis Cluster

An NFX150 chassis cluster supports two `cl` interfaces, `cl-1/1/0` (primary node) and `cl-8/1/0` (secondary node).

To configure the LTE modules in a chassis cluster:

1. Configure the dialer interface (`dl0`):

```
{primary:node0}[edit]
user@host# set interfaces dl0 unit 0 family inet negotiate-address
user@host# set interfaces dl0 unit 0 family inet6 negotiate-address
user@host# set interfaces dl0 unit 0 dialer-options pool dialer-pool-number
user@host# set interfaces dl0 unit 0 dialer-options dial-string dial-number
user@host# set interfaces dl0 unit 0 dialer-options always-on
```

Sample configuration for the dl0 interface:

```
set interfaces dl0 unit 0 family inet negotiate-address
set interfaces dl0 unit 0 dialer-options pool 1
set interfaces dl0 unit 0 dialer-options always-on
set interfaces dl0 unit 0 dialer-options dial-string 1234
```

2. Configure the LTE interface (cl-1/1/0) on the primary node:

   a. Configure the dialer pool for the LTE physical interface:

   ```
   {primary:node0}[edit]
   user@host# set interfaces cl-1/1/0 dialer-options pool dialer-pool-number
   ```

   b. Specify the priority for the interface. The interface with the higher priority becomes the active interface.

   ```
   {primary:node0}[edit]
   user@host# set interfaces cl-1/1/0 dialer-options pool dialer-pool-number priority priority
   ```

   c. Configure the profile:

   ```
   {primary:node0}[edit]
   user@host# run request modem wireless create-profile profile-id profile-id cl-1/1/0 slot
   sim-slot-number access-point-name apn-name
   ```

   d. Verify that the profile is configured successfully:

   ```
   {primary:node0}[edit]
   user@host# run show modem wireless profiles cl-1/1/0 slot 1
   Profile details
     Max profiles: 16
     Default profile Id: 1

   Profile 1: ACTIVE
     Valid: TRUE
     Username: user1
     Password: ********
     Access point name (APN): 3gnet
   ```

```
  Authentication: CHAP
  IP Version: IPV4V6
Profile 2: Invalid
Profile 3: Invalid
Profile 4: Invalid
Profile 5: Invalid
Profile 6: Invalid
Profile 7: Invalid
Profile 8: Invalid
Profile 9: Invalid
Profile 10: Invalid
Profile 11: Invalid
Profile 12: Invalid
Profile 13: Invalid
Profile 14: Invalid
Profile 15: Invalid
Profile 16: Inactive
  Valid: TRUE
  Access point name (APN): 3gnet
  Authentication: None
  IP Version: IPV4V6
```

e. Activate the SIM card:

```
{primary:node0}[edit]
user@host# set interfaces cl-1/1/0 act-sim sim-slot-number
```

f. Select the profile and configure the radio access type for the SIM card:

```
{primary:node0}[edit]
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number select-profile
profile-id profile-id
user@host# set interfaces cl-1/1/0 cellular-options sim sim-slot-number radio-access
automatic
```

NOTE: If a SIM card is installed in the second slot, then select the profile and configure the radio access type for the SIM card in the second slot as well.

Sample configuration for the cl-1/1/0 interface:

```
set interfaces cl-1/1/0 act-sim 1
set interfaces cl-1/1/0 cellular-options sim 1 select-profile profile-id 1
set interfaces cl-1/1/0 cellular-options sim 1 radio-access automatic
set interfaces cl-1/1/0 cellular-options sim 2 select-profile profile-id 1
set interfaces cl-1/1/0 cellular-options sim 2 radio-access automatic
set interfaces cl-1/1/0 dialer-options pool 1 priority 1
```

3. Repeat Step 2 to configure the LTE interface (`cl-8/1/0`) for the secondary node.

   If you assign the same priority to both interfaces, then the interface that is listed first in the configuration becomes the active interface.

   To verify which interface is the active interface:

```
root@host> show dialer pools
Pool: 1
  Dialer interfaces:      Name            State
                          dl0.0           Active
  Subordinate interfaces: Name            Flags         Priority
                          cl-1/1/0        Active        100
                          cl-8/1/0        Inactive      1
```

   Sample configuration for the cl-8/1/0 interface:

```
set interfaces cl-8/1/0 act-sim 1
set interfaces cl-8/1/0 cellular-options sim 1 select-profile profile-id 1
set interfaces cl-8/1/0 cellular-options sim 1 radio-access automatic
set interfaces cl-8/1/0 cellular-options sim 2 select-profile profile-id 1
set interfaces cl-8/1/0 cellular-options sim 2 radio-access automatic
set interfaces cl-8/1/0 dialer-options pool 1 priority 254
```

4. Verify the status of the wireless network and dialer interface:

```
{primary:node0}[edit]
user@host# run show modem wireless network
LTE Connection details
  Connected time: 210
  IP: 10.90.51.234
```

```
  Gateway: 10.90.51.233
  DNS: 123.123.123.123
  IPv6: ::
  Gatewayv6: ::
  DNSv6: ::
  Input bps: 0
  Output bps: 14
  Bytes Received: 7236
  Bytes Transferred: 25468
  Packets Received: 89
  Packets Transferred: 316
Wireless Modem Network Info
  Current Modem Status: Connected
  Current Service Status: Normal
  Current Service Type: CS
  Current Service Mode: LTE
  Network: CHN-UNICOM
  Mobile Country Code (MCC): 0
  Mobile Network Code (MNC): 0
  Location Area Code (LAC): 0
  Routing Area Code (RAC): 0
  Cell Identification: 0
  Access Point Name (APN): ctnet
  Public Land Mobile Network (PLMN): CHN-UNICOM
  Physical Cell ID (PCI): N/A
  International Mobile Subscriber Identification (IMSI): ****************
  International Mobile Equipment Identification (IMEI/MEID): ****************
  Integrate Circuit Card Identity (ICCID): 89860118802425942389
  Reference Signal Receiving Power (RSRP): N/A
  Reference Signal Receiving Quality (RSRQ): N/A
  Signal to Interference-plus-Noise Ratio (SiNR): N/A
  Signal Noise Ratio (SNR): N/A
  Energy per Chip to Interference (ECIO): 0
```

```
{primary:node0}[edit]
user@host#  run show interfaces dl0.0
Physical interface: dl0, Enabled, Physical link is Up
  Interface index: 522, SNMP ifIndex: 0
  Type: 27, Link-level type: Ethernet, MTU: 1504
  Device flags   : Present Running
  Interface flags: SNMP-Traps
```

```
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 00:00:5e:00:53:82, Hardware address: 00:00:5e:00:53:82
  Last flapped   : Never
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)

  Logical interface dl0.0 (Index 101) (SNMP ifIndex 0)
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    Dialer:
      State: Active, Dial pool: 1
      Dial strings: 1234
      Subordinate interfaces: cl-8/1/0 (Index 519)
      Activation delay: 0, Deactivation delay: 0
      Initial route check delay: 120
      Redial delay: 255
      Callback wait period: 5
      Load threshold: 0, Load interval: 60
    Bandwidth: 300mbps
    Input packets : 1
    Output packets: 4
    Protocol inet, MTU: 1490
    Max nh cache: 0, New hold nh limit: 0, Curr nh cnt: 0, Curr new hold cnt: 0, NH drop cnt:
0
      Flags: Sendbcast-pkt-to-re, Negotiate-Address
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.34.163.0/26, Local: 10.34.163.31, Broadcast: 10.34.163.63
```

By default, the time interval taken to switch to the secondary cl interface when the active cl interface times out is 120 seconds. You can change the time interval by configuring the `redial-delay` option:

```
{primary:node0}[edit]
user@host# user@host# set interfaces dl0 unit 0 dialer-options redial-delay time-in-seconds
```

RELATED DOCUMENTATION

Upgrading the Modem Firmware on NFX Devices Through Over-the-Air (OTA)

CHAPTER 5

Configuring USB Pass-Through on NFX Series Devices

Supporting File Transfer from USB on NFX Series Devices

Starting from Junos OS Release 21.1R1, you can transfer VNF images, NFX software, or any user scripts from USB to NFX devices by enabling the USB pass-through feature. By default, the USB pass-through feature is disabled.

NOTE: Built-in LTE functionality does not work after you enable the USB pass-through feature.

To enable USB pass-through to Junos and mount a USB:

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

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

2. Configure the USB pass-through feature:

```
root@host# set system services usb-pass-through
```

```
root@host# commit
```

3. Restart the device to enable the USB pass-through feature.
4. Verify whether the USB pass-through feature is enabled:

```
root@host# run show system services usb-pass-through
```

```
USB pass through Information

-----------------------------------

Mode: Enabled
```

5. Mount a USB device on an NFX device. This is helpful if network connectivity is unavailable and you need to copy files to or from the device.

NOTE: It is recommended to use a USB with the FAT32 format.

Enter the shell prompt as a root user:

```
root@host>
```

```
root@host> start shell user root
```

```
Password:
```

```
root@host%
```

6. Before inserting the USB device, perform the following:

```
root@host:~ # ls -l /dev/da*
```

```
root@host:~ # ls -l /dev/da*
```

ls: No match.

7. Insert the USB drive in the USB port. An output similar to the following is displayed:

```
root@% umass1: TOSHIBA TransMemory, rev 2.00/1.00, addr 3
        da2 at umass-sim1 bus 1 target 0 lun 0
        da2: <TOSHIBA TransMemory 5.00> Removable Direct Access SCSI-0 device
        da2: 40.000MB/s transfers
        da2: 983MB (2013184 512 byte sectors: 64H 32S/T 983C)

        root@:~ # ls -l /dev/da*
        crw-r-----  1 root  operator  0x93 Feb  4 04:22 /dev/da0
        crw-r-----  1 root  operator  0x94 Feb  4 04:22 /dev/da0p1
```

In the sample output, `/dev/da0p1` is the USB drive. If the device supports multiple USBs, use the right file that is corresponding to the attached USB. If the console session is not available while inserting

the USB, check the **messages** var log file for logs related to `da` (for example, `show log messages | match da`). It logs the same four lines as shown on console if the USB is inserted.

8. Create a directory for the USB drive to mount to:

```
root@host% mkdir /var/tmp/usb
```

9. Mount the USB drive to the **/var/tmp/usb** directory:

> **NOTE**: **ls /var/tmp/usb** directory shows all files that are present in the USB drive.

```
root@host% mount_msdosfs /dev/da0p1 /var/tmp/usb
```

```
root@host% ls /var/tmp/usb
```

images.tgz

10. Unmount the USB drive after the file is completely copied:

```
root@host% umount /var/tmp/usb
```

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

Table 9: 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:

> **NOTE**: Multiple IKE gateway address redundancy is not supported on NFX350 devices if the deamon is IKED daemon. Only KMD daemon supports this functionality.

```
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
```

UTM on NFX Devices

The Unified threat management (UTM) solution consolidates several security features to protect against multiple threat types. The UTM solution for NFX devices consists of the following security features:

- Antispam—Examines e-mail messages to identify spam. When the device detects an e-mail spam, it drops the message or tags the message header or subject field with a preprogrammed string. For more information, see *Antispam Filtering Overview*.
- Antivirus—Offers a less CPU-intensive alternative to the full file-based antivirus feature. Sophos uses a scanning engine and virus signature databases to protect against virus-infected files, worms, trojans, spyware, and other malware over POP3, HTTP, SMTP, IMAP, and FTP protocols. The virus pattern and malware database is located on external servers maintained by Sophos (Sophos Extensible List) servers. For more information, see Sophos Antivirus Protection on NFX Devices (OBSOLETE).
- Content filtering—Blocks or permits certain types of traffic based on the MIME type, file extension, protocol command, and embedded object type. For more information, see *Content Filtering*.

- Web filtering—Allows you to manage Internet usage by preventing access to inappropriate Web content. The Web filtering solution consists of the following types:
  - Redirect web filtering
  - Local web filtering
  - Enhanced Web filtering

  For more information, see *Web Filtering Overview*.

NOTE: Antispam, Sophos antivirus, and enhanced web filtering are licensed features and will not function until you install the respective licenses.

RELATED DOCUMENTATION

Intrusion Detection and Prevention on NFX Devices

Integrated User Firewall Support on NFX Devices

Application Security on NFX Devices

The NFX150 devices support the AppSecure feature, which is a suite of application-aware security services that deliver security services to provide visibility and control over the types of applications traversing in the networks. AppSecure uses a sophisticated classification engine to accurately identify applications regardless of port or protocol, including nested applications that reside within trusted network services.

The AppSecure feature comprises of the following services:

- Application identification (AppID)- Recognizes traffic at different network layers using characteristics other than port number. Once the application is determined, AppSecure service modules can be configured to monitor and control traffic for tracking, prioritization, access control, detection, and prevention based on the application ID of the traffic. For more information, see Application Identification.
- Application Tracking (AppTrack)—Tracks and reports applications passing through the device. For more information, see Application Tracking on NFX Devices.
- Application Firewall (AppFW)—Implements an application firewall using application-based rules. For more information, see Application Firewall on NFX Devices.

- Application Quality of Service (AppQoS)—Provides quality-of-service prioritization based on application awareness. For more information, see Application QoS.
- Advanced policy-based routing (APBR)— Classifies session based on applications and applies the configured rules to reroute the traffic. For more information, see Advanced Policy-Based Routing on NFX Devices.

AppSecure works with additional content security on the device through integrated unified threat management (UTM), intrusion prevention systems (IPS), and Juniper Networks Sky Advanced Threat Prevention (Sky ATP) for deeper protection against malware, spam, phishing, and application exploits.

RELATED DOCUMENTATION

> *Integrated User Firewall Support on NFX Devices*

Intrusion Detection and Prevention on NFX Devices

Intrusion detection is the process of monitoring the events occurring in your network and analyzing them for signs of possible incidents, violations, or imminent threats to your security policies. Intrusion prevention is the process of performing intrusion detection and then stopping the detected incidents. These security measures are available as intrusion detection systems (IDS) and intrusion prevention systems (IPS), which become part of your network to detect and stop potential incidents.

An Intrusion Detection and Prevention (IDP) policy lets you selectively enforce various attack detection and prevention techniques on the network traffic passing through your device. Juniper devices offer the same set of IDP signatures that are available on Juniper Networks IDP Series Intrusion Detection and Prevention Appliances to secure networks against attacks. The basic IDP configuration involves the following tasks:

- Download and install the IDP license.
- Download and install the signature database—You must download and install the IDP signature database. The signature databases are available as a security package on the Juniper Networks website. This database includes attack object and attack object groups that you can use in IDP policies to match traffic against known attacks.
- Configure recommended policy as the IDP policy—Juniper Networks provides predefined policy templates to use as a starting point for creating your own policies. Each template is a set of rules of a specific rulebase type that you can copy and then update according to your requirements.

  To get started, we recommend you use the predefined policy named “Recommended”.

- Enable a security policy for IDP inspection—For transit traffic to pass through IDP inspection, you configure a security policy and enable IDP application services on all traffic that you want to inspect.

For information on configuring IDP on NFX Series devices, see the Intrusion Detection and Prevention User Guide.

RELATED DOCUMENTATION

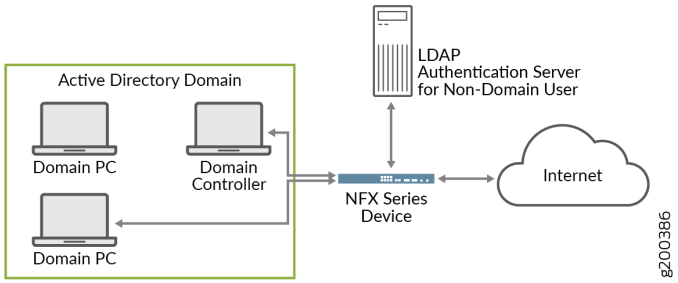
UTM on NFX Devices

Integrated User Firewall Support on NFX Devices

The integrated user firewall feature introduces an authentication source via integration with Microsoft Active Directory. This feature consists of the device polling the event log of the Active Directory controller to determine, by username and source IP address, who has logged in to the device. Then the username and group information are queried from the LDAP service in the Active Directory controller. Once the device has the IP address, username, and group relationship information, it generates authentication entries. With the authentication entries, the device user firewall module enforces user-based and group-based policy control over traffic.

Figure 7 on page 79 illustrates a typical scenario where the integrated user firewall feature is deployed. Users in the Active Directory domain and users outside the Active Directory domain want access to the Internet through the device. The domain controller might also act as the LDAP server.

Figure 7: Scenario for Integrated User Firewall



The device reads and analyzes the event log of the domain controller and generates an authentication table as an Active Directory authentication source for this feature. The user firewall is aware of any domain user on an Active Directory domain device via the Active Directory authentication source. The

device administrator configures a user firewall policy that enforces the desired user-based or group-based access control.

For information on configuring the integrated user firewall on NFX Series devices, see Authentication and Integrated User Firewalls User Guide.

RELATED DOCUMENTATION

- *UTM on NFX Devices*

CHAPTER 7

Configuring Virtual Network Functions

Prerequisites to Onboard Virtual Network Functions on NFX250 NextGen Devices

IN THIS SECTION

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 NFX250 NextGen devices. Before onboarding a VNF, you must test the VNF according to your use case scenario.

NFX250 NextGen Device Prerequisites to Onboard a VNF

To onboard VNFs on NFX250 NextGen, 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 "NFX250 NextGen Overview" on page 2.

Before you onboard the VNF, check the following NFX250 NextGen device capabilities:

- Check the current performance mode of the device by using the `show vmhost mode` command. The NFX250 NextGen 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 10 on page 83 lists the possible memory availability for VNF usage for the NFX250 NextGen models.

Table 10: Memory Availability for VNF Usage

Model	Memory Availability for VNF Usage (Junos OS 19.1R1 Release)
NFX250-S1	6 GB
NFX250-S1E	6 GB
NFX250-S2	22 GB
NFX250-LS1	6 GB

- 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 11 on page 83 lists the CPUs available for VNF usage for the NFX250 NextGen models.

  Table 11: CPUs Available for VNF Usage (Junos OS 19.1R1 Release)

Model	CPUs Available for VNF Usage		
	Throughput Mode	Hybrid Mode	Compute Mode
NFX250-S1	0	4	8
NFX250-S2	0	4	8
NFX250-S1E	0	4	8
NFX250-LS1	0	2	4

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

For more information, see "Configuring VNFs on NFX250 NextGen Devices" on page 90.

VNF Prerequisites to Onboard on an NFX250 NextGen Device

To onboard a VNF on an NFX250 NextGen device, the following VNF properties should be met:

- 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.

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 | no-more
Memory Information
------------------

Virtual Memory:
---------------
Total       (KiB): 15914872
Used        (KiB): 8242468
Available   (KiB): 8265920
Free        (KiB): 7672404
Percent Used     : 48.1

Huge Pages:
------------
Total 1GiB Huge Pages:      2
Free 1GiB Huge Pages:       0
Configured 1GiB Huge Pages: 0
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
```

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 | no-more
Mode:
--------
Current Mode: compute

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

Memory Allocations:
Name                                Configured                          Used
-------------------------------------------------------------------------------------------------
------------------------
Junos Control Plane (mB)            2048                                1994
NFV Backplane 1G hugepages          1                                   2
NFV Backplane 2M hugepages          -                                   0
Layer 2 1G hugepages                -                                   -
Layer 2 2M hugepages                -                                   -
Layer 3 1G hugepages                1                                   1
Layer 3 2M hugepages                1376                                1375
```

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 NFX250 NextGen Devices

IN THIS SECTION

The NFX250 NextGen 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, you can either use the `file-copy` command or copy the image from a USB by using the `usb-pass-through` command.

NOTE: You must save the VNF image in the **/var/public** directory.

```
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`.

To copy a VNF image from a USB, see "Supporting File Transfer from USB on NFX Series Devices" on page 62.

Prepare the Bootstrap Configuration

You can bootstrap a VNF using an attached 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.

The config drive is a virtual drive, which can be a CD-ROM, USB drive or Disk drive associated to a VNF with the configuration data. Configuration data can be files or folders, which are bundled in the ISO file that makes a virtual CD-ROM, USB drive, or Disk drive.

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.

By attaching a config drive, you can pass the networking configurations such as the IP address, subnet mask, and gateway to the VNFs through a CLI. After receiving the configuration inputs, the device generates a bootstrap-config ISO file, and attaches the file to the VNF as a CD-ROM, USB drive, or Disk drive.

For more information about configuring and attaching a config drive, see "(Optional) Attach a Config Drive to the VNF " on page 98.

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.
- The config drive is a read-only drive. Based on the VNF, you can specify the config drive as a read-only CD-ROM drive, USB drive, or a Disk drive.

The config drive supports the following data for VNFs:

- Static content as files—The device accepts one or more file paths through a CLI, converts these files to an ISO image, and attaches it to the VNF. The config drive supports multiple static files in a VNF configuration.
- Jinja2 template and parameters—Jinja2 parameters consist of key-value pairs. The key is specified in the template and the value replaces the key when the template is rendered. The system adds the rendered output file to the ISO image, and attaches it to the VNF. The maximum number of parameters for a template is 256 key-value pairs. The config drive supports multiple templates and its parameters in a VNF configuration.

  NOTE: The config drive supports only Jinja2 templates.

- Directory—The device accepts the specific directory contents, converts the folder structure in the given folder to an ISO image, and attaches it to the VNF. The config drive accepts only one folder. That folder becomes the root directory in the ISO image, and all the subsequent folders and files are added to the ISO image.

NOTE:

- You can add multiple source templates and source files in a VNF configuration.
- To add multiple source templates and one source folder in a VNF configuration, the target template file must be inside the source folder.

- You can add only one source folder in a VNF configuration.
- If two VNFs share the same set of files, separate bootstrap-config ISO files are generated for each VNF. Deleting one VNF will not affect the other VNF.

Allocate CPUs for a VNF

Table 12 on page 93 lists the CPUs available for VNF usage for the NFX250 models.

Table 12: CPUs Available for VNF Usage

Model	CPUs Available for VNF Usage				
	Throughput Mode	Hybrid Mode	Compute Mode	Custom Mode	
				Flex Mode	Perf Mode
NFX250-S1	0	4	8	8	8
NFX250-S1E	0	4	8	8	8
NFX250-S2	0	4	8	8	8

NOTE: The resource allocations for *flex* and *perf* custom modes are based on the templates provided in the default Junos configuration.

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
```

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.

Starting in Junos OS Release 22.1 R1, you can pin the emulator to specific physical CPUs by using the following command:

```
user@host# set virtual-network-functions vnf-name emulator physical-cpu cpu-range
```

You cannot use CPU 0 or offline CPUs for emulator pinning. If you do not pin the emulator to a specific physical CPU, QEMU automatically pins it to a virtual CPU. Changes to emulator pinning take effect immediately on a running VNF.

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 13 on page 96 lists the possible memory availability for VNF usage for the NFX250 models.

Table 13: Memory Availability for VNF Usage

Model	Total Memory Available	Hugepages Availability for VNF Usage in Compute, Hybrid, and Throughput Modes	Hugepages Availability for VNF Usage in Custom Mode	
			Flex Mode	Perf Mode
NFX250-LS1	16 GB	6 1G hugepages	9 1G hugepages	9 1G hugepages
NFX250-S1 and NFX250-S1E	16 GB	6 1G hugepages	9 1G hugepages	9 1G hugepages
NFX250-S2	32 GB	22 1G hugepages	24 1G hugepages	24 1G hugepages

NOTE: The resource allocations for *flex* and *perf* custom modes are based on the templates provided in the default Junos configuration.

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.

(Optional) Attach a Config Drive to the VNF

To attach a config drive to a VNF:

1. Launch the VNF:

```
user@host# set virtual-network-functions vnf-name image image-file-path
user@host# set virtual-network-functions vsrx2 image image-type image-type
```

For example:

```
user@host# set virtual-network-functions vsrx2 image /var/public/media-vsrx-vmdisk-15.1X49-D78.4.qcow2.1
user@host# set virtual-network-functions vsrx2 image image-type qcow2
```

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

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

For example:

```
user@host# set virtual-network-functions vsrx2 virtual-cpu count 2
```

3. Pin virtual CPUs to physical CPUs:

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

For example:

```
user@host# set virtual-network-functions vsrx2 virtual-cpu 0 physical-cpu 4
user@host# set virtual-network-functions vsrx2 virtual-cpu 1 physical-cpu 5
```

4. Enable hardware virtualization for the VNF CPUs:

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

For example:

```
user@host# set virtual-network-functions vsrx2 virtual-cpu features hardware-virtualization
```

5. Specify the maximum primary memory that the VNF can use:

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

For example:

```
user@host# set virtual-network-functions vsrx2 memory size 4194304
```

6. Allocate hugepages:

```
user@host# set virtual-network-functions vnf-name memory features hugepages page-size page-size
```

For example:

```
user@host# set virtual-network-functions vsrx2 memory features hugepages page-size 1024
```

7. Disable autostart of the VNF when the VNF configuration is committed:

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

For example:

```
user@host# set virtual-network-functions vsrx2 no-autostart
```

8. Specify the source file to add in the config drive:

```
user@host# set virtual-network-functions vnf-name config-data source file source-file-path
user@host# set virtual-network-functions vnf-name config-data source file source-file-path
```

For example:

```
user@host# set virtual-network-functions vsrx2 config-data source file /var/public/
source_file1
user@host# set virtual-network-functions vrsx2 config-data source file /var/public/
source_file2
```

9. Specify the template file to add in the config drive:

> **NOTE**: A template file can be of any format and keys are written inside the double {}. This feature replaces keys with values provided in the CLI to create a file and attach as storage media to the VNF. Its use depends upon the VNF. For more information about how to create a template, refer to jinja2 template guidelines.

```
user@host# set virtual-network-functions vnf-name config-data source template template_name
file file-path
user@host# set virtual-network-functions vnf-name config-data source template template_name
parameters image_path image-path
user@host# set virtual-network-functions vnf-name config-data source template template_name
parameters image_type image-type
```

For example:

```
user@host# set virtual-network-functions vsrx2 config-data source template template_sample
file /var/public/template_sample
user@host# set virtual-network-functions vsrx2 config-data source template template_sample
parameters image_path /var/tmp/disk_image.qcow2
user@host# set virtual-network-functions vsrx2 config-data source template template_sample
parameters image_type qcow2
```

Following is a sample template:

```
user@host# cat /var/public/template_sample
Image {
        {{image_path}};
               Image-type {{image_type}};
}
memory {
```

```
    size {{mem_size}};
    features {
        hugepages {
            page-size {{page_size}};
        }
    }
}
```

10. Specify the maximum memory of the source template:

```
user@host# set virtual-network-functions vnf-name config-data source template template_name
parameters mem-size memory-size
```

For example:

```
user@host# set virtual-network-functions vsrx2 config-data source template template_sample
parameters mem-size 4096
```

11. Allocate pages for the source template:

```
user@host# set virtual-network-functions vnf-name config-data source template template_name
parameters page-size page-size
```

For example:

```
user@host# set virtual-network-functions vsrx2 config-data source template template_sample
parameters page-size 1024
```

12. Specify the target file that contains the generated file from the source template:

```
user@host# set virtual-network-functions vnf-name config-data source template template_name
target target-file-path
```

For example:

```
user@host# set virtual-network-functions vsrx2 config-data source template template_sample
target /var/public/template_output
```

13. Specify the device name, device type, and device label to add in the config drive:

```
user@host# set virtual-network-functions vnf-name config-data target device-name device-name
user@host# set virtual-network-functions vnf-name config-data target device-type device-type
user@host# set virtual-network-functions vnf-name config-data target device-label device-
label
```

For example:

```
user@host# set virtual-network-functions vsrx2 config-data target device-name hda
user@host# set virtual-network-functions vsrx2 config-data target device-type cdrom
user@host# set virtual-network-functions vsrx2 config-data target device-label
template_label
```

The `target device-type` is optional. If you do not specify, it takes the device type as `cd-rom`.

The `target device-label` is optional. If you do not specify, it takes the device label as `config-data`.

14. Commit the configuration:

```
user@host# commit
```

To verify whether the config drive is attached to the VNF, see the `VNF Disk Information` section in the "show system visibility vnf" on page 288 command output message.

For example:

```
user@host> show system visibility vnf vsrx2
VNF Memory Usage
-------------------------------------------------------------------------------------------------
-------------------
Name                                  Maximum Memory (KiB)  Used Memory (KiB)  Used 1G
Hugepages  Used 2M Hugepages
------------------------------------ -------------------- ------------------
------------------ -----------------
vsrx2                                 4194304               4194304
4                  0

VNF CPU Statistics (Time in ms)
-------------------------------------------------------------------------
Name                                  CPU Time     System Time  User Time
```

```
------------------------------------ ------------ ------------ ------------
vsrx2                                3288         1510         770

VNF MAC Addresses
------------------------------------------------------------
VNF                                     MAC
---------------------------------------- -----------------
vsrx2_ethdef0                           9C:CC:83:BD:8C:42
vsrx2_ethdef1                           9C:CC:83:BD:8C:43

VNF Internal IP Addresses
------------------------------------------------------
VNF                                     IP
---------------------------------------- ---------------
vsrx2                                   192.0.2.100

VNF Interfaces
------------------------------------------------------------------------------------------------------
-------
VNF                  Interface Type      Source       Model      MAC               IPv4-address
-------------------- --------- --------- ------------ ---------- -----------------
---------------------
vsrx2                vnet4     network   default      virtio     9c:cc:83:bd:8c:42 --
vsrx2                vnet5     bridge    eth0br       virtio     9c:cc:83:bd:8c:43 --

VNF Disk Information
------------------------------------------------------------------------------------------------------
--------
VNF                          Disk        File
---------------------------- -----------
-------------------------------------------------------------
vsrx2                        vda         /var/public/media-vsrx-vmdisk-15.1X49-D78.4.qcow2.1
vsrx2                        hda         /var/public/vnf_config_data_vsrx2

VNF Disk Usage
------------------------------------------------------------------------------
VNF                  Disk      Read Req   Read Bytes   Write Req  Write Bytes
-------------------- --------- ---------- ------------ ---------- ------------
vsrx2                vda       14125      109740032    0          0
vsrx2                hda       0          0            0          0

VNF Port Statistics
------------------------------------------------------------------------------------------------------
```

```
--------------------------
VNF                 Port      Rcvd Bytes   Rcvd Packets Rcvd Error Rcvd Drop Trxd Bytes   Trxd
Packets Trxd Error Trxd Drop
------------------- --------- ------------ ------------ ---------- --------- ------------
------------ ---------- ---------
vsrx2               vnet4     52           1            0          0         0
0            0          0
vsrx2               vnet5     60           1            0          0         0
0            0          0

VNF Media Information
-------------------------------------------------------------------------------------------------
--------------
VNF                           Media Disk         File
----------------------------- ----- ------------
-------------------------------------------------------------
vsrx2                         CDROM hda          /var/public/vnf_config_data_vsrx2
```

Configure Interfaces and VLANs for a VNF

You can configure a VNF interface, map a VNF interface to a virtual function, and attach the interface to a physical NIC port, a management interface, or VLANs, assign a VLAN ID to it, and enable trust mode on it.

Prior to Junos OS Releases 21.3R1, 21.2R2, 21.2R1, 21.1R2, and 20.4R3, the step to configure an SR-IOV VNF interface and to assign a VLAN ID is as follows:

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

Starting from Junos OS Releases 21.3R1, 21.2R2, 21.2R1, 21.1R2, and 20.4R3, the steps to configure an SR-IOV VNF interface, to assign a VLAN ID, and to enable trust mode are as follows:

To map a VNF interface to a virtual function:

```
user@host# set virtual-network-functions vnf-name interfaces vnf-interface-name mapping
interface physical-interface-name
```

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

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

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

To enable trust mode:

```
user@host# set virtual-network-functions vnf-name interfaces vnf-interface-name mapping
interface virtual-function trust
```

NOTE:

- Trust mode is supported on NFX Series devices from Junos OS Releases 21.3R1, 21.2R2, 21.2R1, 21.1R2, and 20.4R3.
- If you enable trust mode on VNF SR-IOV interface, then the VNF interface goes into promiscuous mode.

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]
```

A VNF interface can be mapped to one or more physical interface .You can enable this functionality by configuring the virtual port peer (VPP) feature. You can configure mappings between an OVS interface of a VNF to one or more front panel interfaces. The VNF interface becomes inactive if all of the mapped physical interfaces are inactive. The VNF interface becomes active even if at least one of the mapped physical interface is active.

NOTE:

- The mapped physical interface does not become inactive if a VNF interface is inactive.
- Before upgrading a software image that does not support trust mode to an image that supports trust mode, it is recommended to delete all VNF interface to virtual-function mappings from the configuration.
- Before downgrading a software image that supports trust mode to an image that does not support trust mode, it is necessary to delete all VNF interface to virtual-function mappings from the configuration. Else, the device goes into **Amnesiac** state after the downgrade.

The interface to the VNF is an OVS port and this mapping is defined in the configuration. If the mapping rules can view multiple physical ports before triggering the action, configuring the VPP feature allows you to manage multiple, redundant physical links.

You can configure a mapping between VNF virtual interfaces and JCP physical interfaces (ge-0/0/*x* and xe-0/0/*x*). One virtual interface can be mapped to one or more physical interfaces. There is no limit on the number of physical interfaces to which a VNF virtual interface can be mapped to. You can map a VNF virtual interface to all the physical interfaces or you can map multiple VNF interfaces to a single physical interface.

To configure VPP:

```
root@host# set virtual-network-functions vnf-name interfaces interface-name mapping peer-
interfaces physical-interface-name
```

For example:

```
root@host# set virtual-network-functions centos1 interfaces eth2 mapping peer-interfaces ge-0/0/6
```

To view mapping of the peer interfaces, run the `show system visibility vnf vnf-name` command.

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 NFX250 (NG) 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.

Quick CLI Configuration

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

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

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

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

```
user@host# set virtual-network-functions vnf-name interfaces vnf-interface-name mapping interface physical-interface-name
```

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

```
user@host# set virtual-network-functions vnf-name interfaces vnf-interface-name mapping interface virtual-function trust
```

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

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

```
root@host# set virtual-network-functions vnf-name interfaces interface-name mapping peer-interfaces physical-interface-name
```

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

```
file-name
```

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

```
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
```

```
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
```

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.

Virtual Route Reflector on NFX250 NextGen Overview

The virtual route reflector (vRR) feature allows you to implement the route reflector capability in a virtualized environment. Starting in Junos OS Release 21.4R2, you can implement the vRR feature on an NFX250 NextGen device. You can configure the vRR VNF in compute or hybrid mode. However, we

recommend that you configure the vRR VNF in flex mode as you can allocate maximum resources to the VNF in flex mode. This topic describes how to configure the vRR VNF in flex mode.

For more information about vRR, refer Virtual Route Reflector (vRR) Documentation.

How to Configure a vRR VNF on an NFX250 NextGen Device

Starting in Junos OS Release 21.4R2, you can configure a vRR as a VNF on an NFX250 NextGen device. Before you configure the vRR VNF:

- Delete all third-party VNFs deployed on the device.
- Verify that there are no hugepages configured on the device. If hugepages are configured, then delete them.
- Allocate minimum resources to the Layer 2 data plane and OVS. This ensures that maximum resources are allocated to the vRR VNF.

```
user@host# set vmhost mode custom flex layer-3-infrastructure cpu count MIN
user@host# set vmhost mode custom flex layer-3-infrastructure memory size MIN
user@host# set vmhost mode custom flex nfv-back-plane cpu count MIN
user@host# set vmhost mode custom flex nfv-back-plane memory size MIN
user@host# commit
```

To configure a vRR VNF:

1. Change the mode to flex mode. The default mode is throughput. We recommend deploying the vRR VNF in flex mode on NFX250 NextGen devices.

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

   Note that auto-complete might not include flex mode (compute, hybrid, and throughput modes are available with auto-complete). You might have to manually type in the flex keyword.

   When prompted, enter **yes** to reboot the device.

   After the device reboots, issue the `show vmhost mode` command to verify that the device is in flex mode.

   ```
   user@host> show vmhost mode
   Mode:
   ```

```
--------
Current Mode: flex

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

Memory Allocations:
Name                                Configured                          Used
-------------------------------------------------------------------------------------------
-------------------------
Junos Control Plane (mB)            1536                                1521
NFV Backplane 1G hugepages          1                                   5
NFV Backplane 2M hugepages          -                                   0
Layer 2 1G hugepages                -                                   -
Layer 2 2M hugepages                -                                   -
Layer 3 1G hugepages                -                                   0
Layer 3 2M hugepages                282                                 275
VNF max memory limit (gB) (approx)  24                                  24.000
```

2. Configure hugepages for memory requirements. For example, the following configuration configures 24GB hugepages, which can be used for a vRR VNF:

```
user@host# set system memory hugepages page-size 1024 page-count 24
```

NOTE: You must reboot the system after configuring hugepages to pre-allocate hugepages during bootup.

3. Copy the vRR VNF image to the **/var/public** folder.

4. Define the vRR VNF. For example:

```
user@host# set virtual-network-functions VRR-1 image /var/public/junos-x86-64-18.2X75-
D420.28.img
```

5. Specify the number of virtual CPUs required for the vRR VNF. It is recommended that at least two virtual CPUs are assigned to a vRR VNF. For example:

```
user@host# set virtual-network-functions VRR-1 virtual-cpu count 2
```

6. Connect a virtual CPU to a physical CPU. You can use the `show vmhost mode` command to identify the CPUs that you want to use. For example:

```
user@host# set virtual-network-functions VRR-1 virtual-cpu 0 physical-cpu 10
user@host# set virtual-network-functions VRR-1 virtual-cpu 1 physical-cpu 11
```

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

```
user@host# set virtual-network-functions VRR-1 interfaces eth2 mapping vlan mode trunk

user@host# set virtual-network-functions VRR-1 interfaces eth2 mapping vlan members vlan2
user@host# set virtual-network-functions VRR-1 interfaces eth3 mapping vlan mode trunk
user@host# set virtual-network-functions VRR-1 interfaces eth3 mapping vlan members vlan3
```

8. Specify the memory allocation for the vRR VNF. For example:

```
user@host# set virtual-network-functions VRR-1 memory size 25165824
```

9. Configure hugepages for memory requirements. For example:

```
user@host# set virtual-network-functions VRR-1 memory features hugepages
```

10. Commit the configuration to activate the vRR VNF.

```
user@host# commit
```

After you commit the configuration, VNF takes some time to boot.

11. Verify that the VNF is up. For example:

```
user@host> show virtual-network-functions
ID Name       State     Liveliness
------------------------------------------------------------------------------
2 VRR-1       Running     alive
1 vjunos0     Running     alive
```

You can use the `request virtual-network-functions console VRR-1` command to access the vRR VNF console. You can also ssh to the vRR VNF using the `request virtual-network-functions ssh VRR-1` command after configuring vRR Junos for ssh connectivity.

Managing VNFs on NFX Series 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 or specified in an init-descriptor XML file, are assigned a globally unique and persistent MAC address. A common pool of 64 MAC addresses is used to

assign MAC addresses to VNF interfaces. You can configure a MAC address other than what is available in the common pool, and this address will not be overwritten.

There are 160 MAC addresses for the network interfaces on the VNF. These MAC addresses are automatically allocated when a VNF is instantiated.

- 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 2048 bytes as the MTU size. The default MTU value is 1500 bytes, and the maximum MTU size for a VNF interface is 2048 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 25.

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 vnf-name ssh
```

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

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

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
```

CHAPTER 8

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.

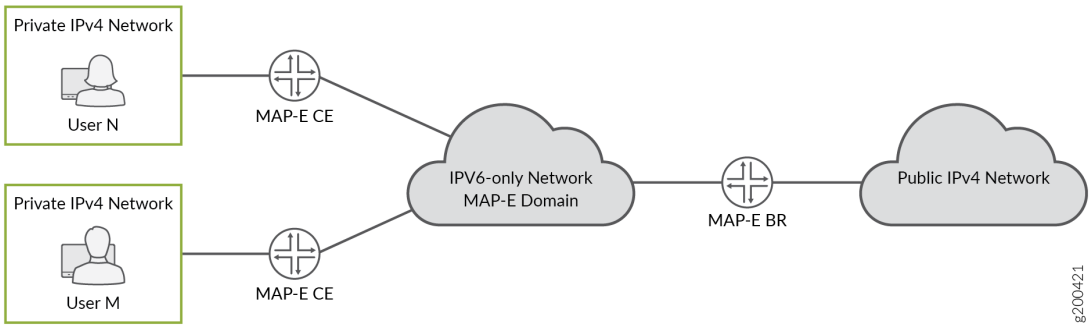
MAP-E Terminology

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 126 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*.

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.

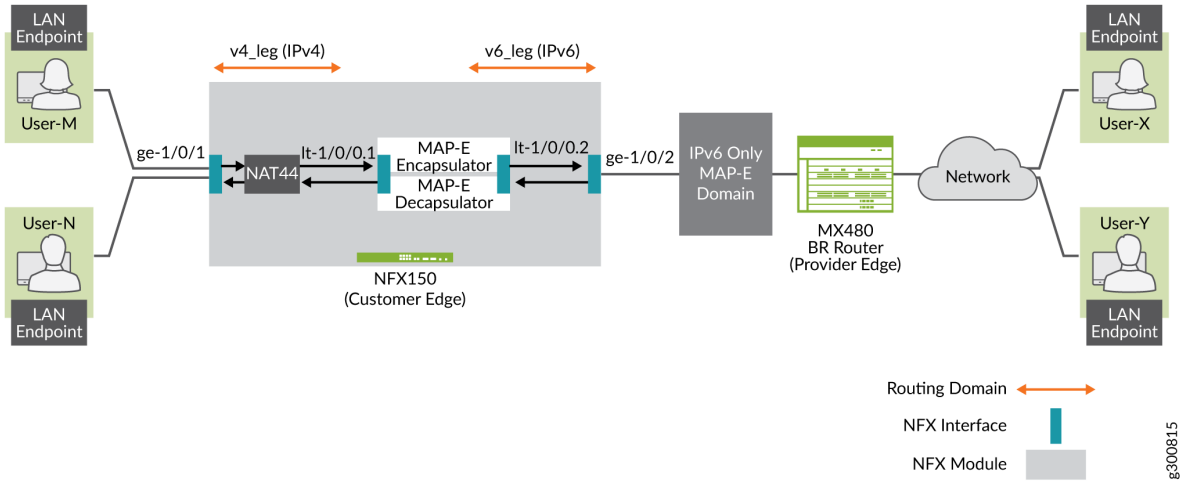
Topology Overview

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 128 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. (Optional) Configure the `confidentiality` option for MAP-E if you want to hide the MAP-E parameters in show command output for non-super users:

```
user@host# set security softwires map-e confidentiality
```

For more information, see confidentiality and show security softwires map-e confidentiality status.

7. 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
```

8. 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

IN THIS SECTION

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 High Availability

Chassis Cluster on NFX250 NextGen Devices

IN THIS SECTION

A chassis cluster, where two devices operate as a single device, provides high availability (HA) on NFX250 NextGen devices. Chassis clustering involves the synchronizing of configuration files and the dynamic runtime session states between the devices, which are part of the chassis cluster setup.

NFX250 NextGen Chassis Cluster Overview

IN THIS SECTION

You can configure NFX250 NextGen devices to operate in cluster mode by connecting and configuring a pair of devices to operate like a single node, providing redundancy at the device, interface, and service level.

When two devices are configured to operate as a *chassis cluster*, each device becomes a node of that cluster. The two nodes back up each other, with one node acting as the primary device and the other node acting as the secondary device, ensuring stateful failover of processes and services when the system or hardware fails. If the primary device fails, the secondary device takes over the processing of traffic.

The nodes of a cluster are connected together through two links called control link and fabric link. The devices in a chassis cluster synchronize the configuration, kernel, and PFE session states across the cluster to facilitate high availability, failover of stateful services, and load balancing.

- Control link—Synchronizes the configuration between the nodes. When you submit configuration statements to the cluster, the configuration is automatically synchronized over the control interface.

  To create a control link in a chassis cluster, connect the ge-0/0/0 interface on one node to the ge-0/0/0 interface on the second node.

  NOTE: You can use only the ge-0/0/0 interface to create a control link.

- Fabric link (data link)—Forwards traffic between the nodes. Traffic arriving on a node that needs to be processed on the other node is forwarded over the fabric link. Similarly, traffic processed on a node that needs to exit through an interface on the other node is forwarded over the fabric link.

  You can use any interface except the ge-0/0/0 to create a fabric link.

Chassis Cluster Modes

The chassis cluster can be configured in active/passive or active/active mode.

- **Active/passive mode**—In active/passive mode, the transit traffic passes through the primary node while the backup node is used only in the event of a failure. When a failure occurs, the backup device becomes the primary device and takes over all forwarding tasks.
- **Active/active mode**—In active/active mode, the transit traffic passes through both nodes all the time.

Chassis Cluster Interfaces

The chassis cluster interfaces include:

- Redundant Ethernet (reth) interface—A pseudo-interface that includes a physical interface from each node of a cluster. The reth interface of the active node is responsible for passing the traffic in a chassis cluster setup.

  A reth interface must contain, at minimum, a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces that are referred to as child interfaces of the redundant Ethernet interface (the redundant parent). If two or more child interfaces from each node are assigned to the redundant Ethernet interface, a redundant Ethernet interface link aggregation group can be formed.

  NOTE: You can configure a maximum of 128 reth interfaces on NFX250 NextGen devices.

- Control interface—An interface that provides the control link between the two nodes in the cluster. This interface is used for routing updates and for control plane signal traffic, such as heartbeat and threshold information that trigger node failover.

  NOTE: By default, the ge-0/0/0 interface is configured as the dedicated control interface on NFX250 NextGen devices. Therefore, you cannot apply any configuration to ge-0/0/0 in HA mode.

- Fabric interface—An interface that provides the physical connection between two nodes of a cluster. A fabric interface is formed by connecting a pair of Ethernet interfaces back-to-back (one from each node). The Packet Forwarding Engines of the cluster uses this interface to transmit transit traffic and to synchronize the runtime state of the data plane software. You must specify the physical interfaces to be used for the fabric interface in the configuration.

Chassis Cluster Limitation

Redundant LAG (RLAG) of reth member interfaces of the same node is not supported. A reth interface with more than one child interface per node is called RLAG.

Example: Configuring a Chassis Cluster on NFX250 NextGen Devices

IN THIS SECTION

This example shows how to set up chassis clustering on NFX250 NextGen devices.

Requirements

Before you begin:

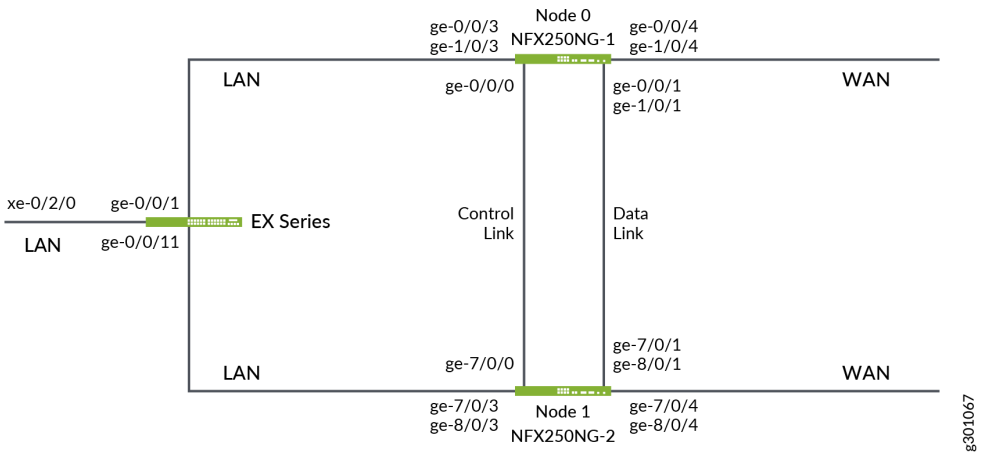
- Physically connect the two devices and ensure that they are the same NFX250 NextGen model.
- Ensure that both devices are running the same Junos OS version
- Remove all interface mapping for the control port ge-0/0/0 on both the nodes.
- Connect the dedicated control port ge-0/0/0 on node 0 to the ge-0/0/0 port on node 1.
- Connect the fabric port on node 0 to the fabric port on node 1.

Overview

Figure 10 on page 143 shows the topology used in this example. This example shows how to set up basic active/passive chassis clustering. One device actively maintains control of the chassis cluster. The other device passively maintains its state for cluster failover capabilities in case the active device becomes inactive.

NOTE: This example does not describe in detail miscellaneous configurations such as how to configure security features. They are essentially the same as they would be for standalone configurations.

Figure 10: NFX250 NextGen Chassis Cluster



Configuration

IN THIS SECTION

Configuring a Chassis Cluster

Step-by-Step Procedure

1. Configure the cluster ID on both the nodes and reboot the devices. A reboot is required to enter into cluster mode after the cluster ID and node ID are set.

   NOTE: You must enter the operational mode to issue the commands on both devices.

   ```
   user@host1> set chassis cluster cluster-id 1 node 0 reboot
   user@host2> set chassis cluster cluster-id 1 node 1 reboot
   ```

   The cluster-id is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster-id is 0 through 255 and setting it to 0 is equivalent to disabling cluster mode.

2. Verify that the chassis cluster is configured successfully:

   - ```
     user@host1> 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      RE  Relinquish monitoring
     ```
     ```

```
Cluster ID: 1
Node   Priority Status               Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 0
node0  1        primary              no      no       None
node1  1        secondary            no      no       None
```

- ```
  root@host1> show chassis cluster information
  node0:
  --------------------------------------------------------------------------
  Redundancy Group Information:

      Redundancy Group 0 , Current State: primary, Weight: 255

          Time            From                To                  Reason
          Mar 15 11:33:47 hold                secondary           Hold timer expired
          Mar 15 11:34:03 secondary           primary             Only node present

  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 15 12:14:49 hold                secondary           Hold timer expired

  Chassis cluster LED information:
      Current LED color: Green
      Last LED change reason: No failures
  ```

After the chassis cluster is set up, you can enter the configuration mode and perform all the configurations on the primary node, node0.
  ```

3. Configure the host names and the out-of-band management IP addresses for nodes 0 and 1:

```
user@host1# set groups node0 system host-name NFX250NG-1
user@host1# set groups node0 interfaces fxp0 unit 0 family inet address 172.16.100.1/24
```

```
user@host2# set groups node1 system host-name NFX250NG-2
user@host2# set groups node1 interfaces fxp0 unit 0 family inet address 172.16.100.2/24
```

If you are accessing the device from a different subnet other than the one configured for the out-of-band management, then set up a static route:

```
user@host1# set routing-options static route 198.51.100.0/24 next-hop 172.16.0.0
user@host1# set routing-options static route 203.0.113.0/24 next-hop 172.16.0.0
```

4. Configure a backup router to access the router from an external network for the out-of-band management

```
user@host1# set groups node0 system backup-router 172.16.0.0
user@host1# set groups node0 system backup-router destination 172.0.0.0/8
user@host1# set groups node0 system backup-router destination 203.0.0.0/8
user@host1# set groups node1 system backup-router 172.16.0.0
user@host1# set groups node1 system backup-router destination 172.0.0.0/8
user@host1# set groups node1 system backup-router destination 203.0.0.0/8
```

Configure Fabric interfaces

Step-by-Step Procedure

The ge-0/0/0 interface is a pre-defined control link. Therefore, you should select any other interface on the device to configure a fabric interface. For example, in the below configuration, ge-0/0/1 is used as the fabric interface.

1. Connect one end of the Ethernet cable to ge-0/0/1 on NFX250NG-1 device and the other end of the cable to ge-0/0/1 on NFX250NG-2 device.

2. Map physical LAN to virtual WAN port:

```
user@host1> set vmhost virtualization-options interfaces ge-8/0/1
user@host1> set vmhost virtualization-options interfaces ge-1/0/1
```

3. Configure front panel (L2) interfaces corresponding to fabric interface:

```
user@host1# set interfaces ge-0/0/1 mtu 9192
user@host1# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
user@host1# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan100
```

```
user@host1# set interfaces sxe-0/0/0 mtu 9192
user@host1# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host1# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host1# set vlans vlan100 vlan-id 100
```

4. Configure L3 interfaces as fabric member:

```
user@host1# set chassis cluster fabric-member ge-1/0/1 vlan-id 100
user@host1# set interfaces fab0 fabric-options member-interfaces ge-1/0/1
user@host1# set groups fab chassis cluster fabric-member ge-1/0/1 vlan-id 100
user@host1# set groups fab chassis cluster fabric-member ge-8/0/1 vlan-id 100
user@host1# set groups fab interfaces fab0 fabric-options member-interfaces ge-1/0/1
user@host1# set groups fab interfaces fab1 fabric-options member-interfaces ge-8/0/1
user@host1# set groups fab vmhost virtualization-options interfaces ge-1/0/1
user@host1# set groups fab vmhost virtualization-options interfaces ge-8/0/1
```

5. Configure data path for fabric interfaces:

```
user@host1# set groups fab interfaces sxe-7/0/0 unit 0 family ethernet-switching vlan members
vlan100
user@host1# set groups fab interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members
vlan100
user@host1# set groups fab interfaces ge-0/0/9 mtu 9000
user@host1# set groups fab interfaces ge-0/0/9 unit 0 family ethernet-switching interface-
mode access
user@host1# set groups fab interfaces ge-0/0/9 unit 0 family ethernet-switching vlan members
vlan100
```

```
user@host1# set groups fab interfaces ge-7/0/9 mtu 9000
user@host1# set groups fab interfaces ge-7/0/9 unit 0 family ethernet-switching interface-
mode access
user@host1# set groups fab interfaces ge-7/0/9 unit 0 family ethernet-switching vlan members
vlan100
user@host1# set groups fab vlan vlan100 vlan-id 100
user@host1# set apply-groups fab
```

6. Configure port peering for fabric and reth members. Port peering ensures that when a LAN interface controlled by the Layer 2 dataplane (FPC0) fails, the corresponding interface on the Layer 3 dataplane (FPC1) is marked down and vice versa. This helps in the failover of the corresponding redundant group to the secondary node.

```
user@host1# set groups node1 chassis cluster redundant-interface ge-8/0/1 mapping-interface
ge-7/0/1
user@host1# set groups node0 chassis cluster redundant-interface ge-1/0/1 mapping-interface
ge-0/0/1
```

7. Enable the system to perform control link recovery automatically. After it determines that the control link is healthy, the system issues an automatic reboot on the node that was disabled when the control link failed. When the disabled node reboots, it rejoins the cluster.

```
user@host1# set chassis cluster control-link-recovery
```

Configure Redundant Groups and Redundant Interfaces

Step-by-Step Procedure

1. Configure redundancy groups 1 and 2. Both `redundancy-group 1` and `redundancy-group 2` control the data plane and include the data plane ports. Each node has interfaces in a redundancy group. As part of redundancy group configuration, you must also define the priority for control plane and data plane—which device is preferred for the control plane, and which device is preferred for the data plane. For chassis clustering, higher priority is preferred. The higher number takes precedence.

In this configuration, `node 0` is the active node as it is associated with `redundancy-group 1`. reth0 is member of `redundancy-group 1` and reth1 is member of `redundancy-group 2`. You must configure all changes in the cluster through node 0. If node 0 fails, then node 1 will be the active node.

```
user@host1# set chassis cluster reth-count 4
user@host1# set chassis cluster redundancy-group 1 node 0 priority 200
user@host1# set chassis cluster redundancy-group 1 node 1 priority 100
user@host1# set chassis cluster redundancy-group 2 node 0 priority 200
user@host1# set chassis cluster redundancy-group 2 node 1 priority 100
user@host1# set chassis cluster redundancy-group 1 preempt
user@host1# set chassis cluster redundancy-group 2 preempt
```

2. Map physical LAN to virtual WAN port for reth members:

```
user@host1# set vmhost virtualization-options interfaces ge-1/0/3
user@host1# set vmhost virtualization-options interfaces ge-1/0/4
user@host1# set vmhost virtualization-options interfaces ge-8/0/3
user@host1# set vmhost virtualization-options interfaces ge-8/0/4
```

3. Configure front panel (L2) interfaces corresponding to reth interface:

```
user@host1# set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode access
user@host1# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan300
```

```
user@host1# set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode access
user@host1# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members vlan400
```

```
user@host1# set interfaces ge-7/0/3 unit 0 family ethernet-switching interface-mode access
user@host1# set interfaces ge-7/0/3 unit 0 family ethernet-switching vlan members vlan300
```

```
user@host1# set interfaces ge-7/0/4 unit 0 family ethernet-switching interface-mode access
user@host1# set interfaces ge-7/0/4 unit 0 family ethernet-switching vlan members vlan400
```

```
user@host1# set interfaces sxe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host1# set interfaces sxe-0/0/1 unit 0 family ethernet-switching vlan members vlan300
user@host1# set interfaces sxe-0/0/1 unit 0 family ethernet-switching vlan members vlan400
```

```
user@host1# set interfaces sxe-7/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host1# set interfaces sxe-7/0/1 unit 0 family ethernet-switching vlan members vlan300
user@host1# set interfaces sxe-7/0/1 unit 0 family ethernet-switching vlan members vlan400
```

```
user@host1# set vlans vlan300 vlan-id 300
user@host1# set vlans vlan400 vlan-id 400
```

4. Configure WAN (L3) interfaces as reth member:

```
user@host1# set interfaces ge-1/0/3 gigether-options redundant-parent reth0
user@host1# set interfaces ge-8/0/3 gigether-options redundant-parent reth0
user@host1# set interfaces ge-1/0/4 gigether-options redundant-parent reth1
user@host1# set interfaces ge-8/0/4 gigether-options redundant-parent reth1
```

5. Configure reth interfaces:

- Configure reth0:

```
user@host1# set interfaces reth0 vlan-tagging
user@host1# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host1# set interfaces reth0 unit 0 vlan-id 300
user@host1# set interfaces reth0 unit 0 family inet address 192.0.2.0/24
```

- Configure reth1:

```
user@host1# set interfaces reth1 vlan-tagging
user@host1# set interfaces reth1 redundant-ether-options redundancy-group 2
user@host1# set interfaces reth1 unit 0 vlan-id 400
user@host1# set interfaces reth1 unit 0 family inet address 198.51.100.0/24
```

6. Configure interface monitoring for reth interfaces members:

```
user@host1# set chassis cluster redundancy-group 1 interface-monitor ge-1/0/3 weight 255
user@host1# set chassis cluster redundancy-group 1 interface-monitor ge-8/0/3 weight 255
user@host1# set chassis cluster redundancy-group 2 interface-monitor ge-1/0/4 weight 255
user@host1# set chassis cluster redundancy-group 2 interface-monitor ge-8/0/4 weight 255
```

7. Configure port peering for reth interface members:

```
user@host1# set groups node1 chassis cluster redundant-interface ge-8/0/3 mapping-interface
ge-7/0/3
user@host1# set groups node1 chassis cluster redundant-interface ge-8/0/4 mapping-interface
ge-7/0/4
user@host1# set groups node0 chassis cluster redundant-interface ge-1/0/3 mapping-interface
ge-0/0/3
user@host1# set groups node0 chassis cluster redundant-interface ge-1/0/4 mapping-interface
ge-0/0/4
```

8. Configure security policies to allow traffic from LAN to WAN, and from WAN to LAN:

```
user@host1# set security policies default-policy permit-all
user@host1# set security zones security-zone trust host-inbound-traffic system-services all
```

```
user@host1# set security zones security-zone trust host-inbound-traffic protocols all
user@host1# set security zones security-zone trust interfaces all
```

Verification

IN THIS SECTION

Verifying Chassis Cluster Status

Purpose

Verify the status of the chassis cluster and its interfaces.

Action

From operational mode, issue the following commands:

- Verify the status of the cluster:

```
root@host1> 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      RE  Relinquish monitoring
    IS  IRQ storm

Cluster ID: 1
Node   Priority Status               Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  1        primary              no      no       None
node1  1        secondary            no      no       None
```

```
Redundancy group: 1 , Failover count: 1
node0  200      primary              yes     no       None
node1  100      secondary            yes     no       None

Redundancy group: 2 , Failover count: 1
node0  200      primary              yes     no       None
node1  100      secondary            yes     no       None
```

- Verify the status of the redundancy groups:

```
root@host1> show chassis cluster information
node0:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: primary, Weight: 255

        Time            From                To                  Reason
        Jun  8 11:24:14 hold                secondary           Hold timer expired
        Jun  8 11:24:30 secondary           primary             Only node present

    Redundancy Group 1 , Current State: primary, Weight: 255

        Time            From                To                  Reason
        Jun  8 11:24:14 hold                secondary           Hold timer expired
        Jun  8 11:24:30 secondary           primary             Only node present

    Redundancy Group 2 , Current State: primary, Weight: 255

        Time            From                To                  Reason
        Jun  8 11:24:14 hold                secondary           Hold timer expired
        Jun  8 11:24:30 secondary           primary             Only node present

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
        Jun  8 11:25:24 hold                secondary           Hold timer expired

    Redundancy Group 1 , Current State: secondary, Weight: 255

        Time            From                To                  Reason
        Jun  8 11:25:24 hold                secondary           Hold timer expired

    Redundancy Group 2 , Current State: secondary, Weight: 255

        Time            From                To                  Reason
        Jun  8 11:25:23 hold                secondary           Hold timer expired

Chassis cluster LED information:
    Current LED color: Green
    Last LED change reason: No failures
```

- Verify the status of the interfaces:

```
root@host1> show chassis cluster interfaces
Control link status: Up

Control interfaces:
    Index   Interface   Monitored-Status   Internal-SA   Security
    0       em1         Up                 Disabled      Disabled

Fabric link status: Up

Fabric interfaces:
    Name    Child-interface    Status                   Security
                               (Physical/Monitored)
    fab0    ge-1/0/1           Up   / Up                Disabled
    fab0
    fab1    ge-8/0/1           Up   / Up                Disabled
    fab1

Redundant-ethernet Information:
    Name         Status      Redundancy-group
    reth0        Up          1
    reth1        Up          2
    reth2        Down        Not configured
```

```
    reth3        Down        Not configured

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    lo0          Up          0

Interface Monitoring:
    Interface         Weight    Status                    Redundancy-group
                                (Physical/Monitored)
    ge-8/0/3          255       Up  /  Up                 1
    ge-1/0/3          255       Up  /  Up                 1
    ge-8/0/4          255       Up  /  Up                 2
    ge-1/0/4          255       Up  /  Up                 2
```

- Verify the status of the port-peering interfaces:

```
root@host1> show chassis cluster port-peering
node0:
--------------------------------------------------------------------------

Port peering interfaces:
    Backend L3                Mapped Peer L2
    Interface  Status         Interface  Status
    ge-1/0/3    Up            ge-0/0/3   Up
    ge-1/0/4    Up            ge-0/0/4   Up
    ge-1/0/1    Up            ge-0/0/1   Up

node1:
--------------------------------------------------------------------------

Port peering interfaces:
    Backend L3                Mapped Peer L2
    Interface  Status         Interface  Status
    ge-8/0/3    Up            ge-7/0/3   Up
    ge-8/0/4    Up            ge-7/0/4   Up
    ge-8/0/1    Up            ge-7/0/1   Up
```

RELATED DOCUMENTATION

Monitoring of Global-Level Objects in a Chassis Cluster

Upgrading or Disabling a Chassis Cluster on NFX250 NextGen Devices

IN THIS SECTION

Upgrading Individual Devices in a Chassis Cluster Separately

Devices in a chassis cluster can be upgraded separately one at a time.

NOTE: During this type of chassis cluster upgrade, a service disruption of about 3 to 5 minutes occurs.

To upgrade each device in a chassis cluster separately:

1. Load the new image file on node 0.
2. Perform the image upgrade without rebooting the node by entering:

```
user@host> request vmhost software add image_name
```

3. Load the new image file on node 1.
4. Repeat Step 2.
5. Reboot both nodes simultaneously.

Disabling a Chassis Cluster

If you want to operate the device as a standalone device or remove a node from a chassis cluster, you must disable the chassis cluster.

To disable a chassis cluster, enter the following command:

```
{primary:node1}
user@host> set chassis cluster disable reboot
```

After the system reboots, the chassis cluster is disabled.

NOTE: You can also disable the chassis cluster by setting the cluster-id to zero on both the nodes:

```
user@host>set chassis cluster cluster-id 0 node 0 reboot
user@host>set chassis cluster cluster-id 0 node 1 reboot
```

CHAPTER 10

Configuring Service Chaining

Example: Configuring Service Chaining Using VLANs on NFX250 NextGen Devices

IN THIS SECTION

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

Requirements

This example uses an NFX250 NextGen device running Junos OS Release 19.1R1.

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

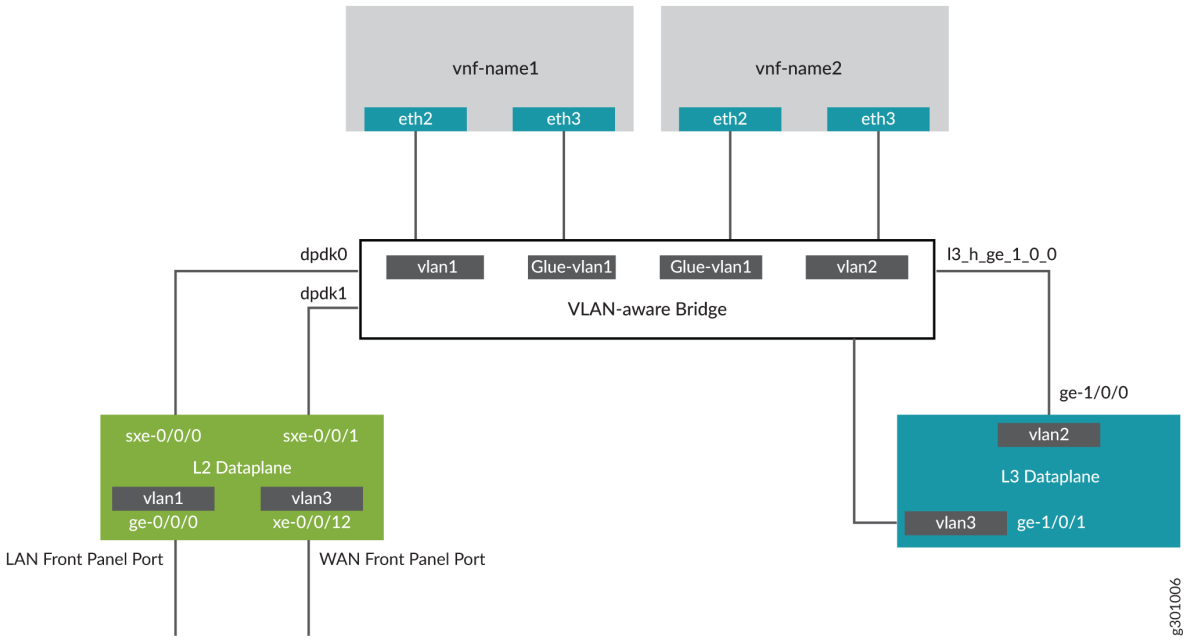
IN THIS SECTION

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 NFX250 NextGen device running Junos OS, as shown in Figure 11 on page 160.

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 vlans vlan3 vlan-id 1178
```

9. Configure VLAN tagging on the WAN-side external-facing interface and assign an IP address.

```
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 192.0.2.1/24
```

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 160.

Example: Configuring Service Chaining Using SR-IOV on NFX250 NextGen 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 NFX250 NextGen device running Junos OS Release 19.1R1.

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

Overview

IN THIS SECTION

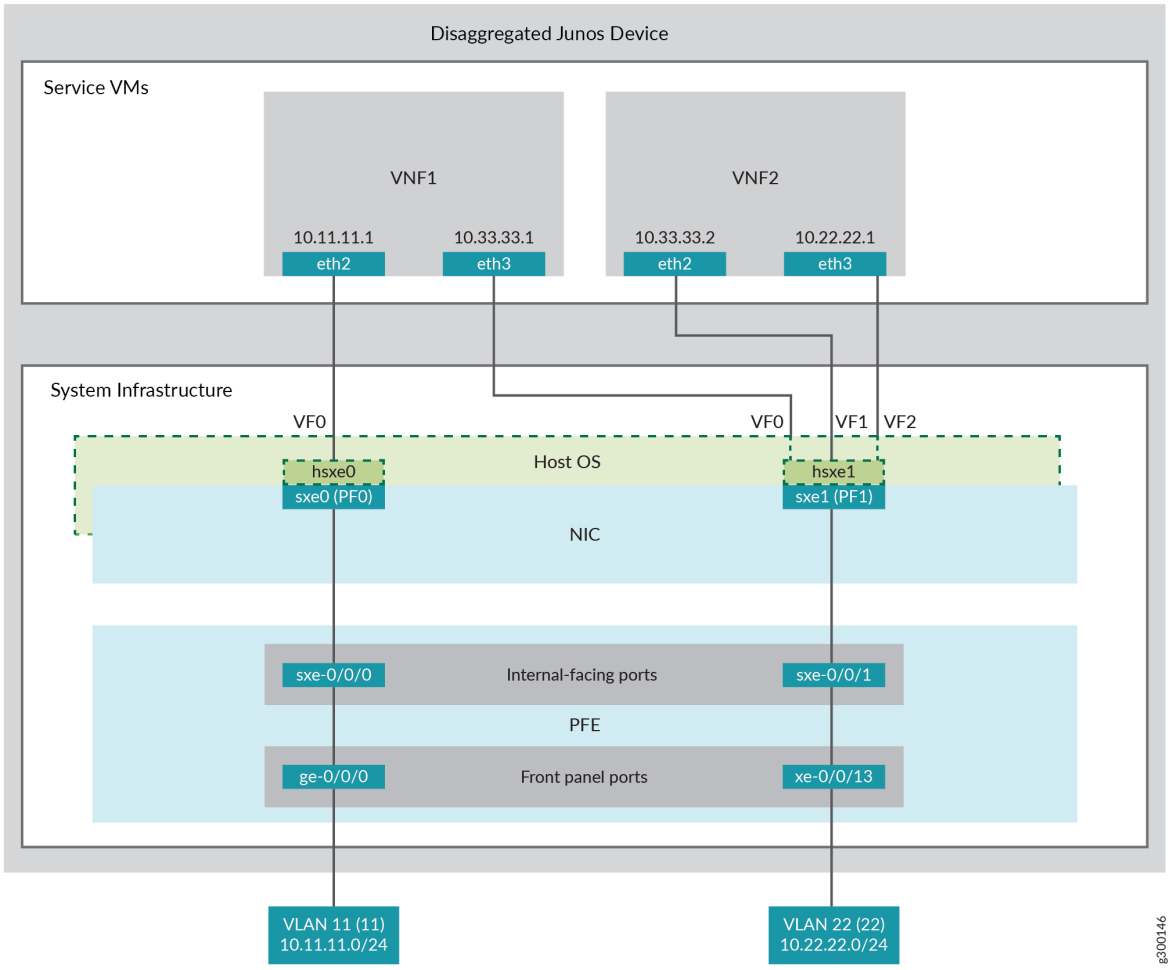
This example uses the front panel ports ge-0/0/0 and xe-0/0/13 associated with the PFE, and its internal-facing ports, sxe-0/0/0 and sxe-0/0/1. The internal NIC ports, sxe0 and sxe1, are not configured directly; instead, they are abstracted at the host OS layer and configured as interfaces hsxe0 and hsxe1. The VNFs use two interfaces, eth2 and eth3. These elements are generally separated into a LAN side and a WAN side.

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 168 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

This example describes:

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-0/0/13.0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/13.0 family ethernet-switching vlan member Vlan22
```

```
set interfaces sxe-0/0/1.0 family ethernet-switching interface-mode trunk
set interfaces sxe-0/0/1.0 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-0/0/13.0 family ethernet-switching interface-mode trunk
   user@host# set interfaces xe-0/0/13.0 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-0/0/1.0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/1.0 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-0/0/13
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-0/0/1
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 interface hsxe0 virtual-
   function
   ```

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

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

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

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

4. 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 NFX250 NextGen Devices

IN THIS SECTION

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

Requirements

This example uses an NFX250 NextGen device running Junos OS Release 19.1R1.

Overview

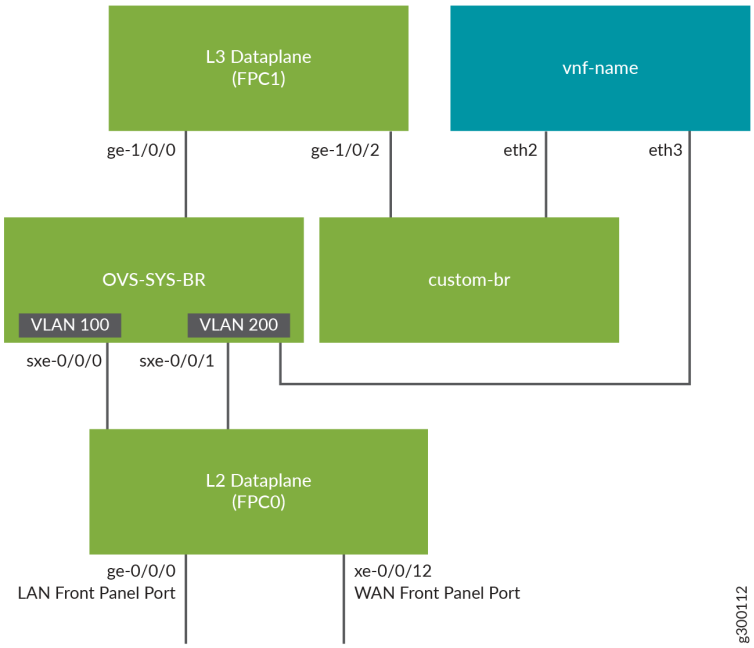
IN THIS SECTION

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 175.

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
```

NOTE: When a VNF interface is mapped to a custom bridge, you should restart the VNF for the mapping to take effect.

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 Cross-Connect on NFX250 NextGen Devices

IN THIS SECTION

This example shows how to configure the cross-connect feature on NFX250 NextGen devices.

Requirements

This example uses an NFX250 NextGen device running Junos OS Release 19.1R1.

Overview

IN THIS SECTION

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.

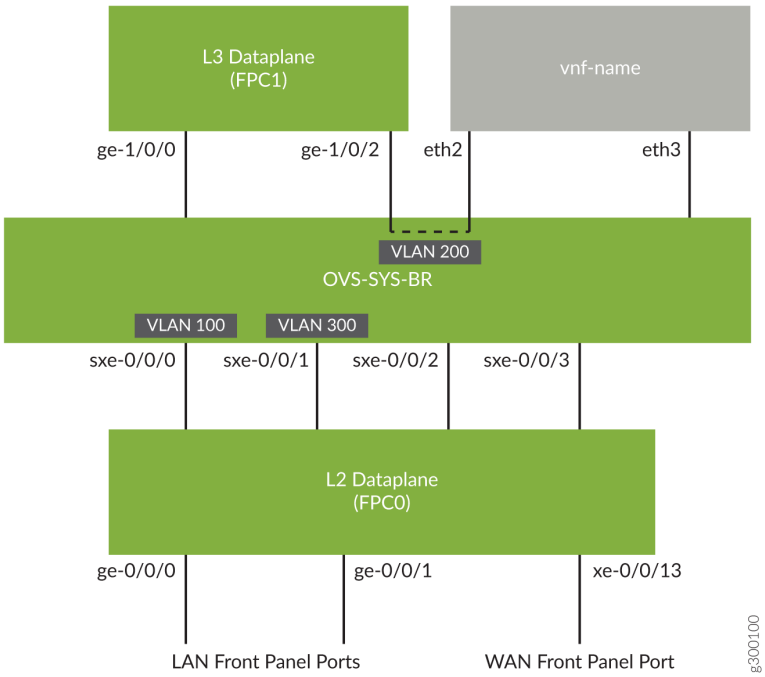
The cross-connect feature supports the following:

- 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 14 on page 186.

Figure 14: Configuring Cross-Connect



Configuration

IN THIS SECTION

Configuring 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
```

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

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
```

Configuring 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
```

Verifying 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
```

RELATED DOCUMENTATION

Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices

Example: Configuring Service Chaining for LAN Routing on NFX250 NextGen Devices

IN THIS SECTION

This example shows how to configure service chaining for LAN routing.

Requirements

This example uses an NFX250 NextGen device running Junos OS Release 19.1R1.

Overview

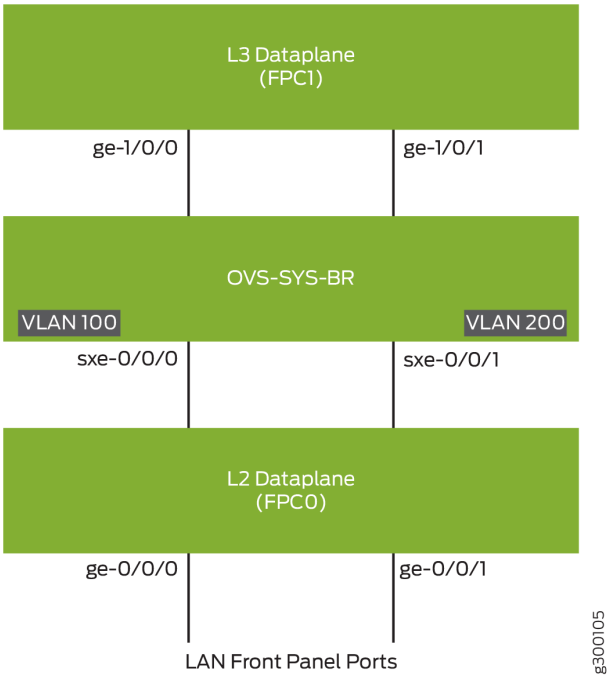
IN THIS SECTION

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 15 on page 197.

Figure 15: 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 NFX250 NextGen Devices

IN THIS SECTION

This example shows how to configure service chaining for LAN to WAN routing.

Requirements

This example uses an NFX250 NextGen device running Junos OS Release 19.1R1.

Overview

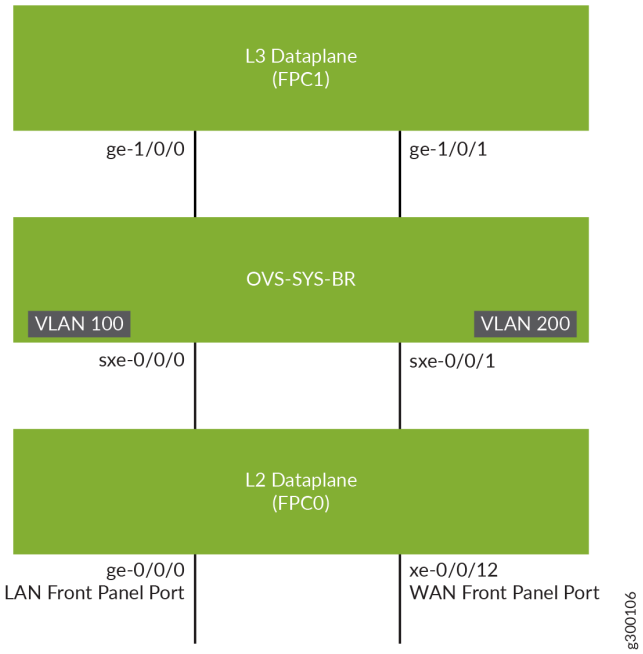
IN THIS SECTION

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 16 on page 201.

Figure 16: 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 NFX250 NextGen Devices

IN THIS SECTION

This example shows how to configure service chaining for LAN to WAN routing through third-party VNFs on NFX250 NextGen devices.

Requirements

This example uses an NFX250 NextGen device running Junos OS Release 19.1R1.

Overview

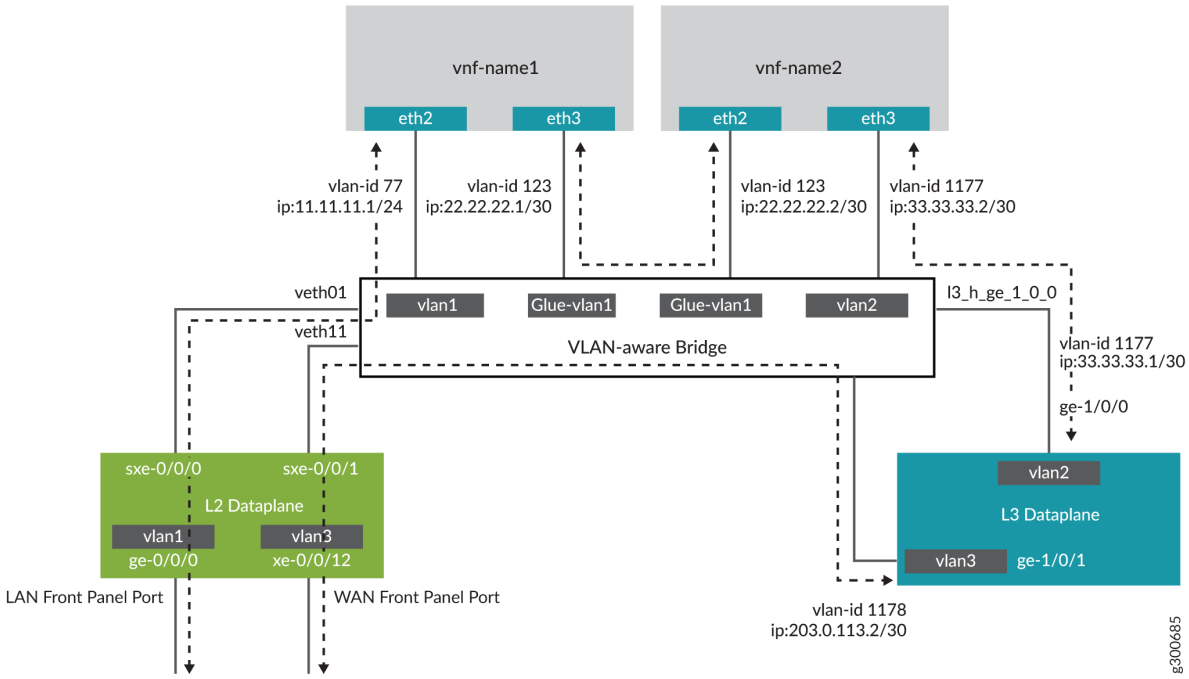
IN THIS SECTION

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 17 on page 206.

Figure 17: 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
```

Configuring the VNF Interfaces for Creating the Service Chain

Step-by-Step Procedure

1. Configure the vmhost instance with the vlans for connecting to the OVS bridge for service chaining:

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

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 glue-vlan1 and and the second interface mapped to VLAN vlan2:

```
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
```

Configuring the Layer 3 Datapath

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
```

Configuring the Layer 2 Datapath (JCP WAN Interfaces)

Step-by-Step Procedure

1. Configure a VLAN for the WAN-side JCP interfaces:

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

2. 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
```

3. 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
```

4. Commit the configuration:

```
user@host# commit
```

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
```

11

CHAPTER

Monitoring and Troubleshooting

Configuring SNMP on NFX150, NFX250 NextGen, and NFX350 Devices

IN THIS SECTION

SNMP monitors network devices from a central location. NFX Series (NFX150, NFX250 NextGen, and NFX350) devices support querying of MIB data by using SNMPv2c and SNMPv3. Separate SNMP agents (known as the SNMP process or snmpd) reside on the vjunos0 and Host OS. The vjunos0 acts as the proxy for the Host OS. The system and chassis related MIB data is available in vjunos0.

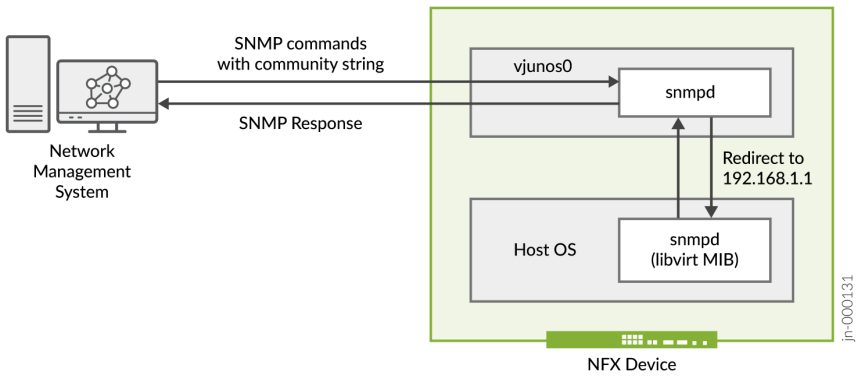
Starting in Junos OS Release 21.4R1, NFX Series devices support LM-SENSORS-MIB, ENTITY-SENSORS-MIB, and libvirt MIB. The LM-SENSORS-MIB and ENTITY-SENSORS-MIB data is available on vjunos0 whereas the libvirt MIB data is available on the Host OS. You can use the libvirt MIB to monitor virtual machines. This topic discusses the SNMP implementation for the libvirt MIB.

How to Configure SNMPv2c to Access Libvirt MIB Data

SNMPv2c uses community strings, which act as passwords when determining the SNMP clients and how clients can access the data in the SNMP agent. The community string is not pre-configured on NFX Series devices. To access MIBs data using SNMPv2c, you must configure a community string and an SNMP proxy for the Host OS. The community string is added to the Host OS.

Figure 18 on page 214 illustrates the communication flow for SNMPv2c on NFX Series devices.

Figure 18: Communication Flow for SNMPv2c on NFX Series Devices



When a user issues SNMP commands like `snmpwalk`, `snmpget` with the community string from the network management server:

- The request goes to the SNMP daemon in vjunos0. The SNMPD reads the community string in the SNMP request and redirects the request to the Host OS using the internal routing instance nfx-host.
- The SNMPD in the Host OS processes the request and sends the response to vjunos0, which then sends it to the network management server.

To configure SNMPv2c:

1. Configure the SNMPv2c community string in the Host OS:

```
root@host# set vmhost snmp v2c community community-name
```

NOTE: Ensure that a community with the same name does not already exist on the device.

2. Configure the proxy in vjunos0:

```
root@host# set snmp proxy snmp-proxy-name device-name 192.168.1.1
root@host# set snmp proxy snmp-proxy-name version-v2c snmp-community community-name
root@host# set snmp proxy snmp-proxy-name nfx-host
```

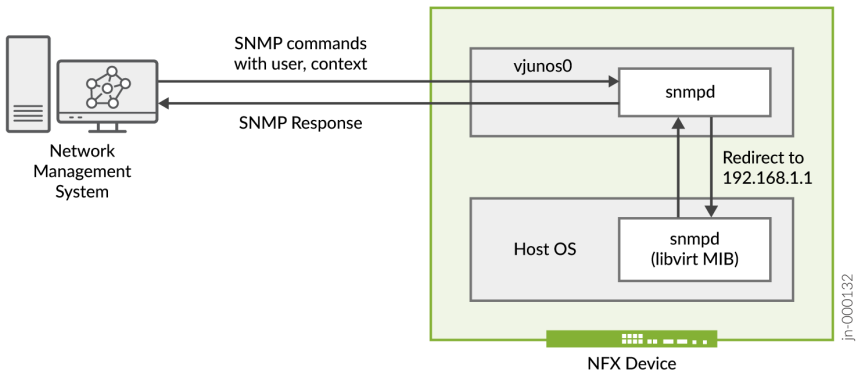
To enable traps, see "How to Enable libvirt SNMPv2c Trap Support" on page 221.

How to Configure SNMPv3 to Access Libvirt MIB Data

SNMPv3 provides a secure way to access MIB data as it supports authentication and encryption. SNMPv3 uses the user-based security model (USM) for message security and the view-based access control model (VACM) for access control. USM specifies authentication and encryption, and VACM specifies access-control rules. Figure 19 on page 215 illustrates the communication flow for SNMPv3 on NFX Series devices. For SNMPv3, you must create:

- An SNMPv3 user under the vmhost hierarchy in Host OS with the authentication type and privacy
- An SNMPv3 proxy with the user name and context

Figure 19: Communication Flow for SNMPv3 on NFX Series Devices



When a user issues SNMP commands like (`snmpwalk`, `snmpget`) with the user name and authentication credentials from the network management server:

- The request goes to the SNMP daemon in vjunos0. The SNMPD reads the context for the Host OS in the SNMP request and redirects the request to the Host OS using the internal routing instance nfx-host.
- The SNMPD in the Host OS processes the request and sends the response to vjunos0, which then sends it to the network management server.

To configure SNMPv3:

1. Configure the local engine information for USM:

```
root@host# set snmp v3 usm local-engine user snmpv3-user-name  authentication-type
authentication-password Authentication_Password
```

```
root@host# set snmp v3 usm local-engine user snmpv3-user-name  privacy-type privacy-password
Privacy_Password
```

2. Configure the remote engine and remote user. You must configure the remote-engine id as 80001f8804686f7374. The remote engine ID is used to compute the security digest for authenticating and encrypting packets sent to a user on the remote host. When sending an inform message, the agent uses the credentials of the user configured on the remote engine (inform target).

```
root@host# set snmp v3 usm remote-engine 80001f8804686f7374 user snmpv3-user-name
authentication-type authentication-password Authentication_Password
root@host# set snmp v3 usm remote-engine 80001f8804686f7374 user snmpv3-user-name  privacy-
type privacy-password Privacy_Password
```

3. Configure VACM:

```
root@host# set snmp v3 vacm security-to-group security-model usm security-name snmpv3-user-
name  group Group-Name
root@host# set snmp v3 vacm access group Group-Name context-prefix snmpv3-context-name
security-model usm security-level authentication read-view iso
```

4. Configure SNMPv3 on the Host OS:

```
root@host# set vmhost snmp v3 user snmpv3-user-name authentication-type authentication-
password Authentication_Password
root@host# set vmhost snmp v3 user snmpv3-user-name privacy-type privacy-password
Privacy_Password
```

5. Configure SNMP v3 proxy:

```
root@host# set snmp proxy snmpv3-proxy-name device-name 192.168.1.1
root@host# set snmp proxy snmpv3-proxy-name version-v3 security-name snmpv3-user-name
root@host# set snmp proxy snmpv3-proxy-name version-v3 context snmpv3-context-name
root@host# set snmp proxy snmpv3-proxy-name nfx-host
```

To enable traps, see "How to Enable libvirt SNMPv3 Trap Support" on page 222.

How to Query Libvirt MIB Data

You can use the **snmpget**, **snmpgetnext**, and **snmpwalk** commands to read the MIB information. Note that you cannot use **snmpset** to configure the libvirt MIB.

The libvirt MIB provides the following information:

- Name of active virtual guest (domain name)
- Current state of the active guest (state of domain)
- Number of virtual CPUs the virtual guest uses (cpu count defined for domain)
- Current amount of memory (in MiB) used by the virtual guest (current allocated memory)
- Memory limit for the domain (the maximum amount of memory (in MiB) that can be used by the virtual guest)
- CPU time used by the virtual guest, in nanoseconds (CPU time)
- Status of the virtual guest (row status)

The following are sample outputs of the **snmpwalk** command when you execute it on NMS:

- SNMPv2c:

```
nms_server# snmpwalk -v2c -Obs -c community-name device_A LIBVIRT-MIB::libvirtMIB
libvirtGuestName.51.145.57.25.141.222.70.137.181.75.129.190.178.93.132.174 = STRING: "centos1"
libvirtGuestName.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = STRING: "vjunos0"
libvirtGuestState.51.145.57.25.141.222.70.137.181.75.129.190.178.93.132.174 = INTEGER:
running(1)
libvirtGuestState.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = INTEGER: running(1)
libvirtGuestCpuCount.51.145.57.25.141.222.70.137.181.75.129.190.178.93.132.174 = Gauge32: 1
libvirtGuestCpuCount.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Gauge32: 1
libvirtGuestMemoryCurrent.51.145.57.25.141.222.70.137.181.75.129.190.178.93.132.174 =
Gauge32: 512
libvirtGuestMemoryCurrent.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Gauge32: 1954
libvirtGuestMemoryLimit.51.145.57.25.141.222.70.137.181.75.129.190.178.93.132.174 = Gauge32:
512
libvirtGuestMemoryLimit.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Gauge32: 1954
libvirtGuestCpuTime.51.145.57.25.141.222.70.137.181.75.129.190.178.93.132.174 = Counter64:
12430000000
libvirtGuestCpuTime.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Counter64:
808830000000
libvirtGuestRowStatus.51.145.57.25.141.222.70.137.181.75.129.190.178.93.132.174 = INTEGER:
```

```
active(1)
libvirtGuestRowStatus.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = INTEGER:
active(1)
libvirtGuestRowStatus.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = No more
variables left in this MIB View (It is past the end of the MIB tree)
```

- SNMPv3:

```
nms-server# snmpwalk -Obs -v3 -a authentication-type -A Authentication_Password -x privacy-
type -X Privacy-Password -l authPriv-level -u snmpv3-user-name -n snmpv3-context-name
device_B LIBVIRT-MIB::libvirtMIB
libvirtGuestName.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = STRING: "vjunos0"
libvirtGuestName.185.178.34.217.212.63.76.189.191.225.54.201.166.88.167.118 = STRING: "vnf0"
libvirtGuestState.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = INTEGER: running(1)
libvirtGuestState.185.178.34.217.212.63.76.189.191.225.54.201.166.88.167.118 = INTEGER:
running(1)
libvirtGuestCpuCount.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Gauge32: 1
libvirtGuestCpuCount.185.178.34.217.212.63.76.189.191.225.54.201.166.88.167.118 = Gauge32: 1
libvirtGuestMemoryCurrent.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Gauge32: 1954
libvirtGuestMemoryCurrent.185.178.34.217.212.63.76.189.191.225.54.201.166.88.167.118 =
Gauge32: 512
libvirtGuestMemoryLimit.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Gauge32: 1954
libvirtGuestMemoryLimit.185.178.34.217.212.63.76.189.191.225.54.201.166.88.167.118 = Gauge32:
512
libvirtGuestCpuTime.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = Counter64:
959760000000
libvirtGuestCpuTime.185.178.34.217.212.63.76.189.191.225.54.201.166.88.167.118 = Counter64:
19830000000
libvirtGuestRowStatus.170.187.204.221.238.255.66.227.0.0.86.74.85.78.79.83 = INTEGER:
active(1)
libvirtGuestRowStatus.185.178.34.217.212.63.76.189.191.225.54.201.166.88.167.118 = INTEGER:
active(1)
```

The following is a sample output of the **snmpwalk** command when you exexute it on the NFX Series device:

```
root@host> show vmhost snmp mib walk LIBVIRT-MIB::libvirtMIB
LIBVIRT-MIB::libvirtGuestName[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = STRING: "centos1"
LIBVIRT-MIB::libvirtGuestName[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = STRING: "vjunos0"
LIBVIRT-MIB::libvirtGuestState[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = INTEGER:
running(1)
```

```
LIBVIRT-MIB::libvirtGuestState[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = INTEGER: running(1)
LIBVIRT-MIB::libvirtGuestCpuCount[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Gauge32: 1
LIBVIRT-MIB::libvirtGuestCpuCount[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Gauge32: 1
LIBVIRT-MIB::libvirtGuestMemoryCurrent[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Gauge32:
512
LIBVIRT-MIB::libvirtGuestMemoryCurrent[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Gauge32: 1954
LIBVIRT-MIB::libvirtGuestMemoryLimit[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Gauge32: 512
LIBVIRT-MIB::libvirtGuestMemoryLimit[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Gauge32: 1954
LIBVIRT-MIB::libvirtGuestCpuTime[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Counter64:
12300000000
LIBVIRT-MIB::libvirtGuestCpuTime[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Counter64:
734900000000
LIBVIRT-MIB::libvirtGuestRowStatus[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = INTEGER:
active(1)
LIBVIRT-MIB::libvirtGuestRowStatus[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = INTEGER:
active(1)
```

Supported Chassis MIBs and Traps

NFX Series devices support the following chassis MIBs:

- jnxFruContentsIndex
- jnxFruL1Index
- jnxFruL2Index
- jnxFruL3Index
- jnxFruName
- jnxFruType
- jnxFruSlot
- jnxFruTemp
- jnxFruOfflineReason
- jnxFruLastPowerOff
- jnxFruLastPowerOn
- jnxFruPowerUpTime

- jnxFruChassisId
- jnxFruChassisDescr
- jnxFruPsdAssignment

NFX Series devices support the following traps:

- jnxFanFailure
- jnxFanOK
- jnxPowerSupplyFailure
- jnxPowerSupplyOK
- jnxOverTemperature
- jnxTemperatureOK
- jnxPowerSupplyRemoved (only for NFX350)

Supported libvirt MIB Traps

IN THIS SECTION

The libvirt MIB monitors the virtual machines and sends asynchronous traps to the network management server. For example, if a domain (VNF) crashes unexpectedly, a notification is sent to the network management server. The traps are generated in the Host OS and sent to the snmptrapd daemon on vjunos0. The snmptrapd daemon forwards the traps to the network management server.

The libvirt trap has the following definition structure:

```
libvirtGuestNotif NOTIFICATION-TYPE
      OBJECTS { libvirtGuestName,
                libvirtGuestUUID,
                libvirtGuestState,
```

```
                libvirtGuestRowStatus }
        STATUS current
        DESCRIPTION
           "Guest lifecycle notification."
        ::= { libvirtNotifications 1 }
```

Here is a sample output of an snmp libvirt trap:

```
SNMPv2-MIB::snmpTrapOID.0 = OID: LIBVIRT-MIB::libvirtGuestNotif,
 LIBVIRT-MIB::libvirtGuestName.0 = STRING: "test1",
 LIBVIRT-MIB::libvirtGuestUUID.1 = STRING: 7ad4bc2a-16db-d8c0-1f5a-6cb777e17cd8,
 LIBVIRT-MIB::libvirtGuestState.2 = INTEGER: running(1),
 LIBVIRT-MIB::libvirtGuestRowStatus.3 = INTEGER: active(1)
```

The current state of the active guest can be one of the following:

- running(1)
- blocked(2)
- paused(3)
- shutdown(4)
- shutoff(5)
- crashed(6)

How to Enable libvirt SNMPv2c Trap Support

To enable SNMPv2c trap support:

1. Configure the community name for the trap:

   ```
   root@host# set vmhost snmp v2c-trap trap-community community-name
   ```

2. Configure the client-address, which is the source address from which the trap originates. If you do not configure the source address, the hypervisor address (192.168.1.1) is used as the client address.

   ```
   root@host# set vmhost snmp client-address client-ip
   ```

3. Configure port-forwarding. You can configure multiple IP adresses.

```
root@host#  set forwarding-options helpers port 162 server IP-address-of-trap-target
```

How to Enable libvirt SNMPv3 Trap Support

To enable SNMPv3 trap support:

1. Configure the user name for the trap:

```
root@host# set vmhost snmp v3-trap trap-user user-name
```

2. Configure the client-address, which is the source address from which the trap originates. If you do not configure the source address, the hypervisor address (192.168.1.1) is used as the client address.

```
root@host# set vmhost snmp client-address client-ip
```

3. Configure the AES and SHA passwords for the user:

```
root@host# set vmhost snmp v3 user user-name authentication-sha authentication-password
password
root@host# set vmhost snmp v3 user user-name privacy-aes128 privacy-password password
```

4. Configure port-forwarding for libVirtMIB trap support. You can configure multiple IP adresses.

```
root@host#  set forwarding-options helpers port 162 server IP-address-of-trap-target
```

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

IN THIS SECTION

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 vmhost cleanup

IN THIS SECTION

Syntax

```
request vmhost cleanup
```

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.

Release Information

Command introduced in Junos OS Release 18.1R1.

request vmhost file-copy

IN THIS SECTION

Syntax

```
request vmhost file-copy (crash|log) from-jnode host file-name to-vjunos host file-name
```

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

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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

request vmhost halt

IN THIS SECTION

Syntax

```
request vmhost halt
```

Description

Stop the host OS and Junos OS running on the device.

Required Privilege Level

maintenance

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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

request vmhost mode

IN THIS SECTION

Syntax

```
request vmhost mode [compute | hybrid | throughput | mode-name]
```

Description

Select the operational mode of the device from a pre-defined list of modes (compute, hybrid, or throughput) or specify a *mode-name* for a custom mode.

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

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)
```

Release Information

The `request vmhost mode` command is introduced in Junos OS Release 19.1R1.

The `request vmhost mode` *`mode-name`* command is introduced in Junos OS Release 21.1R1 for NFX250 NextGen and NFX350 devices.

request vmhost power-off

IN THIS SECTION

Syntax

```
request vmhost power-off
```

Description

Shut down the Junos OS software and the host OS.

Required Privilege Level

maintenance

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
...
...
```

Release Information

Command introduced in Junos OS Release 18.1R1.

NOTE: `request vmhost power-on` is not supported on NFX150 and NFX250 (NG) devices.

request vmhost reboot

IN THIS SECTION

Syntax

```
request vmhost reboot [disk1 | disk2] [primary | alternate]
```

Description

Reboot the Junos OS software and the host OS from the specified disk and the partition within the disk.

Required Privilege Level

maintenance

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!
```

Release Information

Command introduced in Junos OS Release 18.1R1.

request vmhost software add

IN THIS SECTION

Syntax

```
request vmhost software add package-name <in>| <no-validate>| <reboot>| <set>| <unlink>|
<upgrade-to-model model-number>
```

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

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!
```

Release Information

Command introduced in Junos OS Release 18.1R1.

request vmhost storage

IN THIS SECTION

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]
```

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

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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show system inventory hardware cpu

IN THIS SECTION

Syntax

```
show system inventory hardware cpu
```

Description

Display system CPU statistics.

Required Privilege Level

view

Output Fields

Table 14 on page 253 lists the output fields for the `show system inventory hardware cpu` command. Output fields are listed in the approximate order in which they appear.

NOTE: Starting in Junos OS Release 22.1R1, the CPU allocations are displayed as a number range (for example, 4-7) instead of individual numbers (for example, 4,5,6,7).

Table 14: show system inventory hardware cpu Output Fields

Field Name	Field Description
Fields for Inventory CPU Information	
No of CPUs	Total number of CPUs.
No of Logical CPUs/Hyper threads	Total number of hyper threads.
No of CPU sockets	Total number of CPU sockets.
No of Cores per socket	Total number of cores per socket.
No of Hyper threads per core	Total number of hyper threads per core.
CPU Vendor	The name of the CPU vendor.

Table 14: show system inventory hardware cpu Output Fields *(Continued)*

Field Name	Field Description
CPU Model	The CPU model.
CPU Architecture	The type of CPU architecture.
CPU Speed	The CPU speed, in GHz.
CPU Cache	The size of the CPU cache.
No of Max vCPUs	The maximum number of allowed virtual CPUs.
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.
`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 CPU Pinning Information	
`Virtual Machine`	The name of the virtual machine.
`vCPU`	The ID of virtual CPUs used by the virtual machine.

Table 14: show system inventory hardware cpu Output Fields *(Continued)*

Field Name	Field Description
CPU	The ID of CPUs used by the virtual machine.
Fields for Emulator Thread Pinning Information (This section is displayed starting in Junos OS Release 22.1R1.)	
Virtual Machine	The name of 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.
Quota	The CPU quota configured using the `cpu colocation quota` parameter. **NOTE**: This parameter is displayed starting in Junos OS Release 22.1R1.

Sample Output

show system inventory hardware cpu (NFX150)

```
user@host> show system inventory hardware cpu
```

```
Inventory CPU Information
-------------------------
No of CPUs:                       4
No of Logical CPUs/Hyper threads: 4
No of CPU sockets:                1
No of Cores per socket:           4
No of Hyper threads per core:     1
No of Max VCPUs:                  4
CPU Vendor:                       GenuineIntel
```

```
CPU Model:                          Intel(R) Atom(TM) CPU C3558 @ 2.20GHz
CPU Architecture:                   x86_64
CPU Speed:                          2.2000 GHz
CPU Cache:                          2048 KB

CPU Statistics (Time in sec)
----------------------------
User Time:              1044
System Time:            90
Idle Time:              5605
I/O Wait Time:          21
Nice Time:              0
Interrupt Service Time: 0

CPU Pinning Information
---------------------------------------
Virtual Machine             vCPU    CPU
--------------------------- ------- ---
vjunos0                     0       0
testvnf1                    0       2-3
```

show system inventory hardware cpu (NFX250 (NG))

```
user@host> show system inventory hardware cpu
```

```
Inventory CPU Information
-------------------------
No of CPUs:                         4
No of Logical CPUs/Hyper threads: 8
No of CPU sockets:                  1
No of Cores per socket:             4
No of Hyper threads per core:     2
No of Max VCPUs:                    8
CPU Vendor:                         GenuineIntel
CPU Model:                          Intel(R) Pentium(R) CPU D1517 @ 1.60GHz
CPU Architecture:                   x86_64
CPU Speed:                          1.6000 GHz
CPU Cache:                          6144 KB

CPU Statistics (Time in sec)
```

```
---------------------------
User Time:              88347
System Time:            2121
Idle Time:              241343
I/O Wait Time:          73
Nice Time:              0
Interrupt Service Time: 0

CPU Pinning Information
---------------------------------------
Virtual Machine             vCPU    CPU
--------------------------- ------- ---
vjunos0                     0       0
testvnf1                    0       2
testvnf1                    1       6-7

Emulator Thread Pinning Information
-------------------------------
Virtual Machine             CPU
--------------------------- ---
vjunos0                     0
testvnf1                    2-3


System Component                CPUs     Quota
------------------------------- -------- -----
ovs-vswitchd                    0,4
srxpfe                          1,5
jdm
```

show system inventory hardware cpu (NFX350)

```
user@host> show system inventory hardware cpu
```

```
Inventory CPU Information
-------------------------
No of CPUs:                       8
No of Logical CPUs/Hyper threads: 16
No of CPU sockets:                1
No of Cores per socket:           8
```

```
No of Hyper threads per core:     2
No of Max VCPUs:                  14
CPU Vendor:                       GenuineIntel
CPU Model:                        Intel(R) Xeon(R) D-2146NT CPU @ 2.30GHz
CPU Architecture:                 x86_64
CPU Speed:                        2.3000 GHz
CPU Cache:                        11264 KB

CPU Statistics (Time in sec)
----------------------------
User Time:              4738
System Time:            115
Idle Time:              19399
I/O Wait Time:          17
Nice Time:              0
Interrupt Service Time: 0

CPU Pinning Information
---------------------------------------
Virtual Machine             vCPU    CPU
--------------------------- ------- ---
vjunos0                     0       8
testvnf1                    0       3-4
testvnf1                    1       5-7

Emulator Thread Pinning Information
-------------------------------
Virtual Machine             CPU
--------------------------- ---
vjunos0                     8
testvnf1                    11-15


System Component                CPUs     Quota
------------------------------- -------- -----
ovs-vswitchd                    1,8
srxpfe                          0,2
jdm                             8
```

Release Information

Command introduced in Junos OS Release 15.1X53-D40.

show system visibility cpu

IN THIS SECTION

Syntax

```
show system visibility cpu
```

Description

Display details such as per CPU statistics, per CPU usage, and CPU pinning for a Junos OS platform.

Required Privilege Level

view

Output Fields

Table 15 on page 260 lists the output fields for the `show system visibility cpu` command. Output fields are listed in the approximate order in which they appear.

> **NOTE**: Starting in Junos OS Release 22.1R1, the CPU allocations are displayed as a number range (for example, 4-7) instead of individual numbers (for example, 4,5,6,7).

Table 15: 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.
`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

Table 15: show system visibility cpu Output Fields *(Continued)*

Field Name	Field Description
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.
Fields for Emulator Thread Pinning Information (This section is displayed starting in Junos OS Release 22.1R1.)	
Virtual Machine	The name of 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.
Quota	The CPU quota configured using the `cpu colocation quota` parameter. **NOTE**: This parameter is displayed starting in Junos OS Release 22.1R1.

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      542       392         879       0         5           0
1      1747      1           97        0         0           0
2      29        40          1742      0         0           0
3      1361      0           483       0         0           0
4      213       27          1594      0         5           0
5      0         0           1844      0         0           0
6      17        3           1821      0         1           0
7      17        5           1821      0         0           0

CPU Usages
----------------
CPU Id CPU Usage
------ ---------
0      34.600000000000001
1      100.0
2      7.0
3      100.0
4      0.0
5      0.0
6      0.0
7      0.0

CPU Pinning Information
---------------------------------------
Virtual Machine             vCPU    CPU
--------------------------- ------- ---
vjunos0                     0       0

Emulator Thread Pinning Information
-------------------------------
Virtual Machine             CPU
--------------------------- ---
vjunos0                     0


System Component                CPUs     Quota
------------------------------- -------- -----
ovs-vswitchd                    0-1
riot                            0,2
```

```
srxpfe                          0,3
jdm                             0
```

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      40084     1324        270       0         2           0
1      203       395         41009     0         10          0
2      209       21          41435     0         22          0
3      0         0           41680     0         21          0
4      41659     0           64        0         0           0
5      6150      376         33392     0         0           0
6      0         0           41712     0         8           0
7      0         0           41714     0         6           0

CPU Usages
----------------
CPU Id CPU Usage
------ ---------
0      100.0
1      1.0
2      0.0
3      0.0
4      100.0
5      100.0
6      0.0
7      0.0

CPU Pinning Information
---------------------------------------
Virtual Machine             vCPU    CPU
--------------------------- ------- ---
vjunos0                     0       0
testvnf1                    0       2
testvnf1                    1       6-7

Emulator Thread Pinning Information
```

```
------------------------------
Virtual Machine             CPU
--------------------------- ---
vjunos0                     0
testvnf1                    2-3


System Component                CPUs     Quota
------------------------------- -------- -----
ovs-vswitchd                    0,4
srxpfe                          1,5
jdm                             1
```

show system visibility cpu (NFX350)

```
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      1463      76          148       0         3           0
1      1616      3           78        0         0           0
2      1300      1           395       0         0           0
3      112       13          1567      0         2           0
4      1         0           1693      0         1           0
5      0         0           1695      0         2           0
6      0         0           1697      0         0           0
7      0         0           1696      0         0           0
8      188       18          1475      0         4           0
11     0         0           1698      0         0           0
12     0         0           1698      0         0           0
13     0         0           1698      0         0           0
14     0         0           1698      0         0           0
15     0         0           1698      0         0           0

CPU Usages
----------------
CPU Id CPU Usage
------ ---------
0      100.0
1      100.0
```

```
2      100.0
3      0.0
4      0.0
5      1.0
6      0.0
7      0.0
8      23.0
11     0.0
12     0.0
13     0.0
14     0.0
15     0.0

CPU Pinning Information
----------------------------------------
Virtual Machine             vCPU    CPU
--------------------------- ------- ---
vjunos0                     0       8
testvnf1                    0       3-4
testvnf1                    1       5-7

Emulator Thread Pinning Information
-------------------------------
Virtual Machine             CPU
--------------------------- ---
vjunos0                     8
testvnf1                    11-15


System Component                CPUs     Quota
------------------------------- -------- -----
ovs-vswitchd                    1,8
srxpfe                          0,2
jdm                             8
```

Release Information

Command introduced in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

- *show system visibility host*
- *show system visibility memory*
- *show system visibility network*
- *show system visibility vnf*

show system visibility host

IN THIS SECTION

Syntax

```
show system visibility host
```

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

Output Fields

Table 16 on page 267 lists the output fields for the `show system visibility host` command. Output fields are listed in the approximate order in which they appear.

Table 16: 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	
`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.

Table 16: show system visibility host 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.
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).
Percentage Used	The percentage of used disk space.
Fields for Host Disk I/O Information	

Table 16: show system visibility host Output Fields *(Continued)*

Field Name	Field Description
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.

Table 16: show system visibility host Output Fields *(Continued)*

Field Name	Field Description
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.
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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

- *show system visibility cpu*
- *show system visibility memory*
- *show system visibility network*
- *show system visibility vnf*

show system visibility memory

IN THIS SECTION

Syntax

```
show system visibility memory
```

Description

Display the details about virtual memory and shared memory for a Junos OS platform.

Required Privilege Level

view

Output Fields

Table 17 on page 277 lists the output fields for the `show system visibility memory` command. Output fields are listed in the approximate order in which they appear.

Table 17: 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).
`Free`	The total amount of free virtual memory, in kibibytes (KiBs).

Table 17: show system visibility memory Output Fields *(Continued)*

Field Name	Field Description
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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

- *show system visibility cpu*
- *show system visibility host*
- *show system visibility network*
- *show system visibility vnf*

show system visibility network

IN THIS SECTION

Syntax

```
show system visibility network
```

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

Output Fields

Table 18 on page 281 lists the output fields for the `show system visibility network` command. Output fields are listed in the approximate order in which they appear.

Table 18: 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.
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	

Table 18: show system visibility network Output Fields *(Continued)*

Field Name	Field Description
PF	The names of the Physical Functions available.
VF	The names of the Virtual Functions available for each Physical Function.
Reserved For	The owner type for the Virtual Functions.
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           Reserved For
--------- ------------ ----------------
heth-0-0  0000:02:10.5 fpc,vnf
heth-0-0  0000:02:11.1 fpc,vnf
heth-0-0  0000:02:11.5 fpc,vnf
heth-0-0  0000:05:10.2 fpc,vnf
```

```
heth-0-0  0000:05:10.6 fpc,vnf
heth-0-0  0000:05:10.4 fpc,vnf
heth-0-0  0000:02:10.1 fpc,vnf

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           Reserved For
--------- ------------ ----------------
hsxe0     0000:03:10.0 vnf
hsxe0     0000:03:11.4 vnf
hsxe0     0000:03:10.2 vnf
hsxe0     0000:03:11.6 vnf
hsxe0     0000:03:10.4 vnf
hsxe0     0000:03:11.0 vnf
hsxe0     0000:03:10.6 vnf
hsxe0     0000:03:11.2 vnf
hsxe0     0000:03:12.2 vnf
hsxe0     0000:03:12.0 vnf
hsxe0     0000:03:13.4 vnf
hsxe0     0000:03:12.6 vnf
hsxe0     0000:03:13.2 vnf
hsxe0     0000:03:12.4 vnf
hsxe0     0000:03:13.0 vnf
hsxe1     0000:03:11.5 vnf
hsxe1     0000:03:10.1 vnf
hsxe1     0000:03:11.7 vnf
hsxe1     0000:03:10.3 vnf
hsxe1     0000:03:11.1 vnf
hsxe1     0000:03:10.5 vnf
hsxe1     0000:03:11.3 vnf
hsxe1     0000:03:10.7 vnf
hsxe1     0000:03:12.3 vnf
hsxe1     0000:03:13.5 vnf
hsxe1     0000:03:12.1 vnf
hsxe1     0000:03:13.3 vnf
hsxe1     0000:03:12.7 vnf
hsxe1     0000:03:13.1 vnf
hsxe1     0000:03:12.5 vnf

OVS Interfaces
------------------------
NAME              MTU
----------------- ------
dpdk1             1500
ovs-sys-br        1500
```

```
l3_h_ge_1_0_0     1500
dpdk0             1500
```

show system visibility network (NFX350)

```
user@host> show system visibility network
VNF MAC Addresses
------------------------------------------------------------
VNF                                     MAC
---------------------------------------- -----------------
new_ethdef0                             78:4F:9B:2B:2E:4B
new_ethdef1                             78:4F:9B:2B:2E:4C

VNF Internal IP Addresses
------------------------------------------------------------
VNF                                     IP
---------------------------------------- ---------------
new                                     192.0.2.100

VNF Virtual Functions
------------------------------------------------------------------
VNF                                     PF        VF
---------------------------------------- --------- ------------
new                                     hsxe0     0000:b7:02.1
new                                     hsxe0     0000:b7:02.2

Free Virtual Functions
---------------------------------------
PF        VF           Reserved For
--------- ------------ ----------------
hsxe0     0000:b6:02.3 vnf
hsxe0     0000:b6:03.5 fpc
hsxe0     0000:b6:02.1 vnf
hsxe0     0000:b6:02.2 vnf
hsxe0     0000:b6:03.6 fpc
hsxe0     0000:b6:02.4 vnf
hsxe0     0000:b6:02.5 vnf
hsxe0     0000:b6:02.6 vnf
hsxe0     0000:b6:02.7 vnf
hsxe1     0000:b6:06.4 vnf
hsxe1     0000:b6:06.5 vnf
```

```
hsxe1     0000:b6:06.6 vnf
hsxe1     0000:b6:06.7 vnf
hsxe1     0000:b6:06.1 vnf
hsxe1     0000:b6:06.2 vnf
hsxe1     0000:b6:06.3 vnf
hsxe1     0000:b6:07.5 fpc
hsxe1     0000:b6:07.6 fpc
hsxe2     0000:b6:0b.5 fpc
hsxe2     0000:b6:0b.6 fpc
hsxe2     0000:b6:0a.7 vnf
hsxe2     0000:b6:0a.6 vnf
hsxe2     0000:b6:0a.5 vnf
hsxe2     0000:b6:0a.4 vnf
hsxe2     0000:b6:0a.3 vnf
hsxe2     0000:b6:0a.2 vnf
hsxe2     0000:b6:0a.1 vnf
hsxe3     0000:b6:0e.3 vnf
hsxe3     0000:b6:0e.2 vnf
hsxe3     0000:b6:0e.1 vnf
hsxe3     0000:b6:0e.7 vnf
hsxe3     0000:b6:0e.6 vnf
hsxe3     0000:b6:0e.5 vnf
hsxe3     0000:b6:0e.4 vnf
hsxe3     0000:b6:0f.5 fpc
hsxe3     0000:b6:0f.6 fpc

VNF Interfaces
-------------------------------------------------------------------------------------------
VNF                  Interface Type      Source       Model      MAC                VLAN-ID
-------------------- --------- --------- ------------ ---------- ------------------ -------
new                  --        hostdev   --           --         78:4f:9b:2b:2e:4b  97
new                  --        hostdev   --           --         78:4f:9b:2b:2e:4c  4000

OVS Interfaces
------------------------
NAME              MTU
----------------- ------
dpdk0             9216
xdsl_eth0         9192
ovs-sys-br        9192
dpdk2             9216
dpdk1             9216
```

```
dpdk3              9216
l3_h_ge_1_0_0      9216
```

Release Information

Command introduced in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

- *show system visibility cpu*
- *show system visibility host*
- *show system visibility memory*
- *show system visibility vnf*

show system visibility vnf

IN THIS SECTION

Syntax

```
show system visibility vnf vnf name
```

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

Output Fields

Table 19 on page 289 lists the output fields for the `show system visibility vnf` command. Output fields are listed in the approximate order in which they appear.

Table 19: 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.

Table 19: show system visibility vnf Output Fields *(Continued)*

Field Name	Field Description
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.
Fields for List of Virtual Functions per VNF	

Table 19: show system visibility vnf Output Fields *(Continued)*

Field Name	Field Description
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.

Table 19: show system visibility vnf Output Fields *(Continued)*

Field Name	Field Description
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.
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.

Table 19: show system visibility vnf Output Fields *(Continued)*

Field Name	Field Description
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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

RELATED DOCUMENTATION

- *show system visibility cpu*
- *show system visibility host*
- *show system visibility memory*
- *show system visibility network*

show vmhost connections

IN THIS SECTION

Syntax

```
show vmhost connections
```

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

Output Fields

Table 20 on page 297 lists the output fields for the `show vmhost connections` command. Output fields are listed in the approximate order in which they appear.

Table 20: 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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost control-plane

IN THIS SECTION

Syntax

```
show vmhost control-plane
```

Description

Display the status of the JCP, JDM, Layer 2 dataplane, Layer 3 dataplane, and LTE.

Required Privilege Level

view

Sample Output

NOTE: Starting in Junos OS Release 22.1R1, the output of the `show vmhost control-plane` command is enhanced to display the following information:

- Any system component that is currently offline. If you disabled a system component in the currently active custom mode configuration, then the status of that component is displayed as OFFLINE.
- Status and state of the NFV backplane

show vmhost control-plane (NFX150)

```
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       NOT_RUNNING          OFFLINE
Layer 3 Infrastructure       RUNNING              OK
NFV Backplane                RUNNING              OK
```

show vmhost control-plane (NFX250 (NG))

```
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
NFV Backplane                RUNNING              OK
```

show vmhost control-plane (NFX350)

```
user@host> show vmhost control-planeVmhost 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
NFV Backplane                RUNNING              OK
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost crash

IN THIS SECTION

Syntax

```
show vmhost crash
```

Description

Display host OS crash information.

Required Privilege Level

view

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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost forwarding-options analyzer

IN THIS SECTION

Syntax

```
show vmhost forwarding-options analyzer analyzer-name
```

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

Output Fields

Table 21 on page 303 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 21: 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.

Table 21: show vmhost forwarding-options analyzer Output Fields *(Continued)*

Field Name	Field Description
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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost memory

IN THIS SECTION

Syntax

```
show vmhost memory
```

Description

Display the memory information for the host OS.

Required Privilege Level

view

Output Fields

Sample Output

show vmhost memory

```
user@host> show vmhost memory
Memory Controller Information
```

```
-----------------------------

Id :MC0
correctable-error                  :0
uncorrectable-error                :0
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost mode

IN THIS SECTION

Syntax

```
show vmhost mode
show vmhost mode mode-name
```

Description

The `show vmhost mode` command displays the CPU and memory allocations for various components in the current mode of the device. The `show vmhost mode` *mode-name* command displays the CPU and memory allocations for various components for a specific mode of the device.

NOTE: Starting in Junos OS Release 22.1R1, the output of the `show vmhost mode` command displays the maximum memory that can be configured and the currently configured memory for VNFs. In addition, the CPU allocations are displayed as a number range (for example, 4-7) instead of individual numbers (for example, 4,5,6,7).

Required Privilege Level

view

Sample Output (NFX150)

show vmhost mode

```
user@host> show vmhost mode

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

CPU Allocations:
Name                                Configured                              Used
-------------------------------------------------------------------------------------------------
--------------------
Junos Control Plane                 0                                       0,2
Juniper Device Manager              0                                       0
LTE                                 0                                       -
NFV Backplane Control Path          0                                       0
NFV Backplane Data Path             -                                       -
```

```
Layer 2 Control Path                0                                       0
Layer 2 Data Path                   1                                       0-1
Layer 3 Control Path                0                                       0
Layer 3 Data Path                   1                                       1
CPUs available for VNFs             2-3                                     2-3
CPUs turned off                     -                                       -

Memory Allocations:
Name                                Configured                              Used
-------------------------------------------------------------------------------------------------
---------------------
Junos Control Plane (mB)            2048                                    1895
NFV Backplane 1G hugepages          -                                       0
NFV Backplane 2M hugepages          -                                       0
Layer 2 1G hugepages                -                                       0
Layer 2 2M hugepages                128                                     128
Layer 3 1G hugepages                1                                       1
Layer 3 2M hugepages                401                                     400
VNF max memory limit (gB) (approx)  1                                       1.000
```

show vmhost mode hybrid

```
user@host> show vmhost mode hybrid

Mode: hybrid

CPU Allocations:
Name                                Configured
-------------------------------------------------------------------------------
Junos Control Plane                 0
LTE                                 0
Juniper Device Manager              0
NFV Backplane Control Path          0
NFV Backplane Data Path             -
Layer 2 Control Path                0
Layer 2 Data Path                   1
Layer 3 Control Path                0
Layer 3 Data Path                   2
CPUs turned off                     -
CPUs available for VNFs             3
```

```
Memory Allocations:
Name                                Configured
-------------------------------------------------------------------------------
Junos Control Plane (mB)            2048
NFV Backplane 1G hugepages          -
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                128
Layer 3 1G hugepages                1
Layer 3 2M hugepages                401
VNF max memory limit (gB) (approx)  1
```

show vmhost mode throughput

```
user@host> show vmhost mode throughput

Mode: throughput

CPU Allocations:
Name                                Configured
-------------------------------------------------------------------------------
Junos Control Plane                 0
LTE                                 0
Juniper Device Manager              0
NFV Backplane Control Path          0
NFV Backplane Data Path             -
Layer 2 Control Path                0
Layer 2 Data Path                   1
Layer 3 Control Path                0
Layer 3 Data Path                   2-3
CPUs turned off                     -
CPUs available for VNFs             -

Memory Allocations:
Name                                Configured
-------------------------------------------------------------------------------
Junos Control Plane (mB)            2048
NFV Backplane 1G hugepages          -
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                128
```

```
Layer 3 1G hugepages                1
Layer 3 2M hugepages                401
VNF max memory limit (gB) (approx)  -
```

show vmhost mode flex

```
user@host> show vmhost mode flex

Mode: flex

CPU Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane                 0
LTE                                 0
Juniper Device Manager              0
NFV Backplane Control Path          2
NFV Backplane Data Path             -
Layer 2 Control Path                0
Layer 2 Data Path                   1
Layer 3 Control Path                0
Layer 3 Data Path                   2
CPUs turned off                     -
CPUs available for VNFs             3

Memory Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane (mB)            1536
NFV Backplane 1G hugepages          -
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                128
Layer 3 1G hugepages                -
Layer 3 2M hugepages                282
VNF max memory limit (gB) (approx)  2
```

Sample Output (NFX250 (NG))

show vmhost mode

```
user@host> show vmhost mode

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

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

Memory Allocations:
Name                                Configured                          Used
-------------------------------------------------------------------------------------------------
--------------------
Junos Control Plane (mB)            2048                                2020
NFV Backplane 1G hugepages          1                                   2
NFV Backplane 2M hugepages          -                                   0
Layer 2 1G hugepages                -                                   -
Layer 2 2M hugepages                -                                   -
Layer 3 1G hugepages                1                                   1
Layer 3 2M hugepages                1376                                1375
VNF max memory limit (gB) (approx)  6                                   2.000
```

show vmhost mode hybrid

```
user@host> show vmhost mode hybrid

Mode: hybrid

CPU Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane                 0
LTE                                 0
Juniper Device Manager              1
NFV Backplane Control Path          0
NFV Backplane Data Path             4
Layer 2 Control Path                -
Layer 2 Data Path                   -
Layer 3 Control Path                1
Layer 3 Data Path                   2,5
CPUs turned off                     6
CPUs available for VNFs             3,7

Memory Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane (mB)            2048
NFV Backplane 1G hugepages          1
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                -
Layer 3 1G hugepages                1
Layer 3 2M hugepages                1376
VNF max memory limit (gB) (approx)  6
```

show vmhost mode throughput

```
user@host> show vmhost mode throughput

Mode: throughput

CPU Allocations:
```

```
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane                 4
LTE                                 4
Juniper Device Manager              4
NFV Backplane Control Path          4
NFV Backplane Data Path             -
Layer 2 Control Path                -
Layer 2 Data Path                   -
Layer 3 Control Path                0
Layer 3 Data Path                   2-3
CPUs turned off                     1,5-7
CPUs available for VNFs             -

Memory Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane (mB)            2048
NFV Backplane 1G hugepages          -
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                -
Layer 3 1G hugepages                1
Layer 3 2M hugepages                1376
VNF max memory limit (gB) (approx)  -
```

show vmhost mode flex

```
user@host> show vmhost mode flex

Mode: flex

CPU Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane                 0-1
LTE                                 0
Juniper Device Manager              1
NFV Backplane Control Path          7
NFV Backplane Data Path             6
```

```
Layer 2 Control Path                -
Layer 2 Data Path                   -
Layer 3 Control Path                1
Layer 3 Data Path                   7
CPUs turned off                     -
CPUs available for VNFs             2-5,8-11

Memory Allocations:
Name                                Configured
-------------------------------------------------------------------------
Junos Control Plane (mB)            1536
NFV Backplane 1G hugepages          1
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                -
Layer 3 1G hugepages                -
Layer 3 2M hugepages                282
VNF max memory limit (gB) (approx)  24
```

Sample Output (NFX350)

show vmhost mode

```
user@host> show vmhost mode

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              3-7,11-15                            4-5,11,15
CPUs turned off                      9-10                                 -

Memory Allocations:
Name                                 Configured                           Used
-------------------------------------------------------------------------------------------
--------------------
Junos Control Plane (mB)             2048                                 1926
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
VNF max memory limit (gB) (approx)   7                                    2.000
```

show vmhost mode hybrid

```
user@host> show vmhost mode hybrid

Mode: hybrid

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

Memory Allocations:
```

```
Name                                Configured
-------------------------------------------------------------------------
Junos Control Plane (mB)            2048
NFV Backplane 1G hugepages          4
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                -
Layer 3 1G hugepages                4
Layer 3 2M hugepages                5633
VNF max memory limit (gB) (approx)  7
```

show vmhost mode throughput

```
user@host> show vmhost mode throughput

Mode: throughput

CPU Allocations:
Name                                Configured
-------------------------------------------------------------------------
Junos Control Plane                 8
LTE                                 8
Juniper Device Manager              8
NFV Backplane Control Path          8
NFV Backplane Data Path             -
Layer 2 Control Path                -
Layer 2 Data Path                   -
Layer 3 Control Path                0
Layer 3 Data Path                   2-7
CPUs turned off                     1,9-15
CPUs available for VNFs             -

Memory Allocations:
Name                                Configured
-------------------------------------------------------------------------
Junos Control Plane (mB)            2048
NFV Backplane 1G hugepages          -
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                -
Layer 3 1G hugepages                4
```

```
Layer 3 2M hugepages                5633
VNF max memory limit (gB) (approx)  -
```

show vmhost mode flex

```
user@host> show vmhost mode flex

Mode: flex
CPU Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane                 0,1
LTE                                 0
Juniper Device Manager              1
NFV Backplane Control Path          1
NFV Backplane Data Path             4
Layer 2 Control Path                -
Layer 2 Data Path                   -
Layer 3 Control Path                1
Layer 3 Data Path                   5
CPUs turned off                     -
CPUs available for VNFs             2,3,6,7



Memory Allocations:
Name                                Configured
------------------------------------------------------------------------------
Junos Control Plane (mB)            1536
NFV Backplane 1G hugepages          1
NFV Backplane 2M hugepages          -
Layer 2 1G hugepages                -
Layer 2 2M hugepages                -
Layer 3 1G hugepages                1
Layer 3 2M hugepages                200
```

Release Information

Command introduced in Junos OS Release 19.1R1.

show vmhost snmp mib walk

IN THIS SECTION

Syntax

```
show vmhost snmp mib walk mib-name
```

Description

The `show vmhost snmp mib walk` *mib-name* command displays the libvirt SNMP object values that are associated with the libvirt MIB.

NOTE: To view information for other MIBs, use the `show snmp mib walk` command.

Required Privilege Level

view

Sample Output

show vmhost snmp mib walk

```
root@host> show vmhost snmp mib walk LIBVIRT-MIB::libvirtMIB
LIBVIRT-MIB::libvirtGuestName[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = STRING: "centos1"
LIBVIRT-MIB::libvirtGuestName[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = STRING: "vjunos0"
LIBVIRT-MIB::libvirtGuestState[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = INTEGER:
running(1)
LIBVIRT-MIB::libvirtGuestState[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = INTEGER: running(1)
LIBVIRT-MIB::libvirtGuestCpuCount[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Gauge32: 1
LIBVIRT-MIB::libvirtGuestCpuCount[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Gauge32: 1
LIBVIRT-MIB::libvirtGuestMemoryCurrent[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Gauge32:
512
LIBVIRT-MIB::libvirtGuestMemoryCurrent[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Gauge32: 1954
LIBVIRT-MIB::libvirtGuestMemoryLimit[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Gauge32: 512
LIBVIRT-MIB::libvirtGuestMemoryLimit[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Gauge32: 1954
LIBVIRT-MIB::libvirtGuestCpuTime[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = Counter64:
12300000000
LIBVIRT-MIB::libvirtGuestCpuTime[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = Counter64:
734900000000
LIBVIRT-MIB::libvirtGuestRowStatus[STRING: 33913919-8dde-4689-b54b-81beb25d84ae] = INTEGER:
active(1)
LIBVIRT-MIB::libvirtGuestRowStatus[STRING: aabbccdd-eeff-42e3-0-564a554e4f53] = INTEGER:
active(1)
```

Release Information

Command introduced in Junos OS Release 21.4R1.

show vmhost status

IN THIS SECTION

Syntax

```
show vmhost status
```

Description

Display the virtualization status and status of all the CPUs.

Required Privilege Level

view

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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost storage

IN THIS SECTION

Syntax

```
show vmhost storage
```

Description

Display the vmhost storage information.

Required Privilege Level

view

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     : 0000601543968B1000058
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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost uptime

IN THIS SECTION

Syntax

```
show vmhost uptime
```

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

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 330.

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost version

IN THIS SECTION

Syntax

```
show vmhost version detail
```

Description

Display host version information including Linux host kernel version and host software version.

Required Privilege Level

view

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
```

Release Information

Command introduced in Junos OS Release 18.1R1.

show vmhost vlans

IN THIS SECTION

Syntax

```
show vmhost vlans
```

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

Output Fields

Table 22 on page 335 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 22: 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
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

----
```

Release Information

Command introduced in Junos OS Release 18.1R1.