

# How to Configure the NFX150

Published  
2020-03-29

Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

*How to Configure the NFX150*

Copyright © 2020 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

## YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

## END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

## About the Documentation | x

Documentation and Release Notes | x

Using the Examples in This Manual | x

    Merging a Full Example | xi

    Merging a Snippet | xii

Documentation Conventions | xii

Documentation Feedback | xv

Requesting Technical Support | xv

    Self-Help Online Tools and Resources | xvi

    Creating a Service Request with JTAC | xvi

## 1

## Overview

### NFX150 System Overview | 18

    Overview | 18

    NFX150 Models | 19

    Benefits and Uses of NFX150 | 22

### NFX150 Feature Overview | 22

    Software Architecture | 23

    Interfaces | 24

        Physical Interfaces | 25

        Virtual Interfaces | 25

        LTE Interface | 26

        Interface Mapping | 26

    Supported Features | 29

    Performance Modes | 30

Licensing | 30

**Junos OS Releases Supported on NFX Series Hardware | 32**

**NFX Product Compatibility | 33**

Hardware Compatibility | 34

Hardware Compatibility Tool | 34

Software Version Compatibility | 34

NFX250 Software Version Compatibility | 34

## 2

### **Initial Configuration**

**Initial Configuration on NFX150 Devices | 37**

Factory-Default Settings | 37

Enabling Basic Connectivity | 38

Establishing the Connection | 40

**Zero Touch Provisioning on NFX Series Devices | 41**

Understanding Zero Touch Provisioning | 41

Pre-staging an NFX Series Device | 42

Provisioning an NFX Series Device | 44

Provisioning an NFX Series Device Using Sky Enterprise | 45

**Configuring the LTE Module on NFX Devices | 46**

Configuring the LTE Module for Primary Mode | 47

Configuring the LTE Module for Dial-on-Demand Mode | 49

Configuring the LTE Module for Backup Mode | 52

**Upgrading the Modem Firmware on NFX Devices Through Over-the-Air (OTA) | 55**

## 3

### **Configuring Interfaces**

**Mapping Interfaces on NFX150 Devices | 59**

Mapping Physical Interfaces to Virtual Interfaces on NFX150 Devices | 59

Mapping Physical Ports to VNF Interfaces Through SR-IOV | 60

Mapping Layer 3 Dataplane Interfaces to OVS | 60

## Configuring the In-Band Management Interface | 61

## ADSL2 and ADSL2+ Interfaces on NFX150 Devices | 62

ADSL Interface Overview | 62

ADSL2 and ADSL2+ | 63

Configuring ADSL SFP Interface Using VLANs on NFX150 Network Services Platform | 63

Configuring ADSL SFP Interface Without Using VLANs on NFX150 Network Services Platform | 65

## VDSL2 Interfaces on NFX150 Devices | 67

VDSL Interface Overview | 67

VDSL2 Vectoring Overview | 68

VDSL2 Network Deployment Topology | 68

VDSL2 Interface Support on NFX Series Devices | 70

VDSL2 Interface Compatibility with ADSL Interfaces | 71

VDSL2 Interfaces Supported Profiles | 71

VDSL2 Interfaces Supported Features | 72

Configuring VDSL SFP Interface Using VLANs on NFX150 Network Services Platform | 72

Configuring VDSL SFP Interface Without Using VLANs on NFX150 Network Services Platform | 74

# 4

## Configuring Security

### IP Security on NFX Devices | 78

Overview | 78

Configuring Security | 79

Configuring Interfaces | 80

Configuring Routing Options | 81

Configuring Security IKE | 81

Configuring Security IPsec | 84

Configuring Security Policies | 86

Configuring Security Zones | 86

### UTM on NFX Devices | 87

### Application Security on NFX Devices | 88

### Intrusion Detection and Prevention on NFX Devices | 89

**Integrated User Firewall Support on NFX Devices | 89**

## 5

### **Configuring VNFs**

**Prerequisites to Onboard Virtual Network Functions on NFX150 Devices | 92**

Prerequisites for VNFs | 92

**Configuring VNFs on NFX150 Devices | 92**

Load the VNF Image | 93

Prepare the Bootstrap Configuration | 93

Allocate CPUs for a VNF | 95

Allocate Memory for a VNF | 96

(Optional) Attach a Config Drive to the VNF | 97

Configuring VNF Interfaces and VLANs | 99

Configuring VNF Storage Devices | 101

Instantiating a VNF | 102

Verify that the VNF Instantiated Successfully | 103

Instantiating a VNF Using an XML Descriptor File | 104

**Managing VNFs on NFX Series Devices | 104**

Managing VNF States | 105

Managing VNF MAC Addresses | 106

Managing the MTU of a VNF Interface | 107

Accessing a VNF from the JCP | 107

Viewing the List of VNFs | 108

Displaying the Details of a VNF | 108

Deleting a VNF | 109

**Configuring Analyzer VNF and Port-mirroring | 110**

## 6

### **Configuring High Availability**

**Chassis Cluster on NFX150 Devices | 112**

NFX150 Chassis Cluster Overview | 112

Chassis Cluster Modes | 113

Chassis Cluster Interfaces | 113

Chassis Cluster Limitation | 114

Example: Configuring a Chassis Cluster on NFX150 Devices | 114

## Upgrading or Disabling a Chassis Cluster on NFX150 Devices | 125

Upgrading Individual Devices in a Chassis Cluster Separately | 125

Disabling a Chassis Cluster | 126

## 7

## Configuring Service Chaining

### Service Chaining on NFX150 Devices | 128

Understanding Service Chaining | 128

Configuring Service Chaining Using VLANs | 128

Configuring Service Chaining Using DHCP Services on VLANs | 129

### Example: Configuring Service Chaining Using VLANs on NFX150 Network Services Platform | 130

### Example: Configuring Service Chaining Using SR-IOV on NFX150 Network Services Platform | 137

### Example: Configuring Service Chaining Using a Custom Bridge | 144

### Example: Configuring Service Chaining for LAN-WAN Routing | 151

### Example: Configuring Cross Connect on NFX150 Devices | 156

### Example: Configuring Service Chaining for LAN Routing | 166

### Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices | 169

## 8

## Troubleshooting

### Recovering the Root Password for NFX150, NFX250 NextGen, and NFX350 Devices | 180

### Troubleshooting Interfaces on NFX Devices | 183

Monitoring Interface Status and Traffic on NFX Series Devices | 184

## 9

## Operational Commands

request chassis cluster failover node | 190

request chassis cluster failover redundancy-group | 192

request chassis cluster failover reset | 194

request chassis fpc | 195

request vmhost cleanup | 197

request vmhost file-copy | 198

request vmhost halt | 200

request vmhost mode | 202

request vmhost power-off | 204

request vmhost reboot | 206

request vmhost software add | 208

show chassis cluster control-plane statistics | 212

show chassis cluster data-plane interfaces | 215

show chassis cluster data-plane statistics | 217

show chassis cluster information | 220

show chassis cluster interfaces | 226

show chassis cluster port-peering | 232

show chassis cluster statistics | 234

show chassis cluster status | 240

show interfaces | 244

show system visibility cpu | 247

show system visibility host | 251

show system visibility memory | 262

show system visibility network | 265

show system visibility vnf | 271

show vmhost connections | 278

show vmhost control-plane | 280

show vmhost crash | 281



`show vmhost forwarding-options analyzer` | 283

`show vmhost memory` | 285

`show vmhost mode` | 286

`show vmhost status` | 290

`show vmhost storage` | 292

`show vmhost uptime` | 294

`show vmhost version` | 296

`show vmhost vlans` | 298

# About the Documentation

## IN THIS SECTION

- Documentation and Release Notes | x
- Using the Examples in This Manual | x
- Documentation Conventions | xii
- Documentation Feedback | xv
- Requesting Technical Support | xv

Use this guide to perform initial provisioning, configure Junos OS features, chain multiple virtualized network functions, monitor, and manage the NFX150 Series devices.

## Documentation and Release Notes

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

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

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

## Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

## Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xsl;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

## Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {  
    file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

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

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]  
user@host# load merge relative /var/tmp/ex-script-snippet.conf  
load complete
```

For more information about the **load** command, see [CLI Explorer](#).

## Documentation Conventions

[Table 1 on page xiii](#) defines notice icons used in this guide.

Table 1: Notice Icons





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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  user@host> <b>configure</b>
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> <b>show chassis alarms</b>  No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies guide names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS CLI User Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>

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

Convention	Description	Examples
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name:  [edit] root@# <b>set system domain-name</b> <i>domain-name</i>
<b>Text like this</b>	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"><li>• To configure a stub area, include the <b>stub</b> statement at the [edit <b>protocols ospf area area-id</b>] hierarchy level.</li><li>• The console port is labeled <b>CONSOLE</b>.</li></ul>
< > (angle brackets)	Encloses optional keywords or variables.	<b>stub &lt;default-metric <i>metric</i>&gt;;</b>
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	<b>broadcast   multicast</b>  <b>(<i>string1</i>   <i>string2</i>   <i>string3</i>)</b>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	<b>community name members [ <i>community-ids</i> ]</b>
Indentation and braces ( { } )	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		

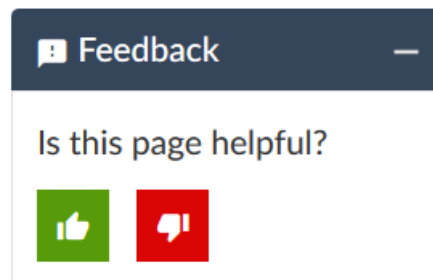
Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
<b>Bold text like this</b>	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .

## Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

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



- Click the thumbs-up icon if the information on the page was helpful to you.
- Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.
- E-mail—Send your comments to [techpubs-comments@juniper.net](mailto:techpubs-comments@juniper.net). Include the document or topic name, URL or page number, and software version (if applicable).

## Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active Juniper Care or Partner Support Services support contract, or are

covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

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

## Self-Help Online Tools and Resources

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

- Find CSC offerings: <https://www.juniper.net/customers/support/>
- Search for known bugs: <https://prsearch.juniper.net/>
- Find product documentation: <https://www.juniper.net/documentation/>
- Find solutions and answer questions using our Knowledge Base: <https://kb.juniper.net/>
- Download the latest versions of software and review release notes: <https://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <https://www.juniper.net/company/communities/>
- Create a service request online: <https://myjuniper.juniper.net>

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

## Creating a Service Request with JTAC

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

- Visit <https://myjuniper.juniper.net>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <https://support.juniper.net/support/requesting-support/>.



# 1

CHAPTER

## Overview

---

NFX150 System Overview | **18**

NFX150 Feature Overview | **22**

Junos OS Releases Supported on NFX Series Hardware | **32**

NFX Product Compatibility | **33**

---

# NFX150 System Overview

## IN THIS SECTION

- Overview | 18
- NFX150 Models | 19
- Benefits and Uses of NFX150 | 22

## Overview

The Juniper Networks NFX150 Network Services Platform is a secure, automated, software-driven customer premises equipment (CPE) platform that delivers virtualized network and security services on demand. NFX150 is part of the Juniper Cloud CPE solution, which leverages Network Functions Virtualization (NFV). It enables service providers to deploy and chain multiple, secure, high-performance virtualized network functions (VNFs) on a single device.

The NFX150 architecture integrates routing, switching, and security functions on a single platform that optimizes the usage of system resources. The architecture enables unified management of all the components through a single CLI. Key components in the NFX150 software include the Junos Control Plane (JCP), Juniper Device Manager (JDM), Layer 2 dataplane, Layer 3 dataplane, and Virtual Network Functions (VNFs).

The NFX150 is suited for small to medium-sized enterprises. With key security features and NFV, the NFX150 can be used in secure SD-WAN and secure router deployments.

The NFX150 portfolio consists of a compact desktop model and a rack-mount model. Both the models are available with or without LTE support. [Figure 1 on page 18](#) shows the compact model without LTE support and [Figure 2 on page 19](#) shows the rack-mount model.

**Figure 1: NFX150 Compact Model (Without LTE Support)**

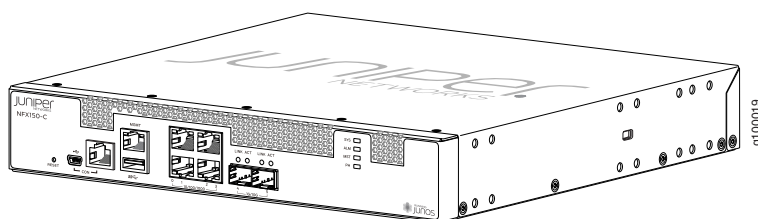
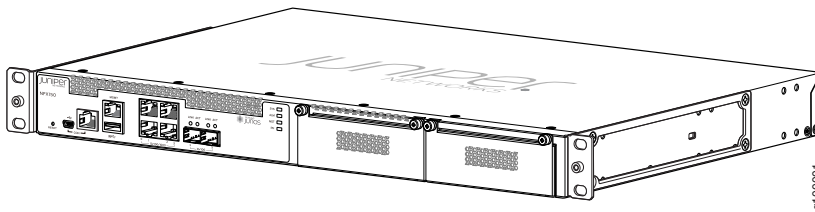


Figure 2: NFX150 Rack-Mount Model



For details on the various models, see [“NFX150 Models” on page 19](#).

## NFX150 Models

The NFX150 is available in seven models. [Table 3 on page 19](#) lists the NFX150-C models and [Table 4 on page 20](#) lists the NFX150-S1 models.

Table 3: NFX150-C (Compact Models)

	NFX150-C-S1	NFX150-C-S1-AE	NFX150-C-S1-AA	NFX150-C-S1E-AE	NFX150-C-S1E-AA
CPU	2.2-GHz 4-core Intel CPU	2.2-GHz 4-core Intel CPU	2.2-GHz 4-core Intel CPU	2.2-GHz 4-core Intel CPU	2.2-GHz 4-core Intel CPU
RAM	8 GB	8 GB	8 GB	16 GB	16 GB
Storage	100 GB SSD	100 GB SSD	100 GB SSD	100 GB SSD	100 GB SSD
Form Factor	Desktop	Desktop	Desktop	Desktop	Desktop

Table 3: NFX150-C (Compact Models) (continued)

	NFX150-C-S1	NFX150-C-S1-AE	NFX150-C-S1-AA	NFX150-C-S1E-AE	NFX150-C-S1E-AA
Ports	Four 10/100/1000BASE-T RJ-45 ports which can be used as either access ports or uplinks	Four 10/100/1000BASE-T RJ-45 ports which can be used as either access ports or uplinks	Four 10/100/1000BASE-T RJ-45 ports which can be used as either access ports or uplinks	Four 10/100/1000BASE-T RJ-45 ports which can be used as either access ports or uplinks	Four 10/100/1000BASE-T RJ-45 ports which can be used as either access ports or uplinks
	Two 1-Gigabit Ethernet /10-Gigabit Ethernet SFP+ ports	Two 1-Gigabit Ethernet /10-Gigabit Ethernet SFP+ ports	Two 1-Gigabit Ethernet /10-Gigabit Ethernet SFP+ ports	Two 1-Gigabit Ethernet /10-Gigabit Ethernet SFP+ ports	Two 1-Gigabit Ethernet /10-Gigabit Ethernet SFP+ 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	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)	Console ports (RJ-45 and mini-USB)	Console ports (RJ-45 and mini-USB)
	One USB 3.0 port	One USB 3.0 port	One USB 3.0 port	One USB 3.0 port	One USB 3.0 port
Expansion module support	No	No	No	No	No
LTE support	No	Yes (integrated LTE modem for Europe and North America)	Yes (integrated LTE modem for Asia, Australia, and New Zealand)	Yes (integrated LTE modem for Europe and North America)	Yes (integrated LTE modem for Asia, Australia, and New Zealand)

Table 4: NFX150-S1 Models

	NFX150-S1	NFX150-S1E
CPU	2.2 GHz 8-core Intel CPU	2.2 GHz 8-core Intel CPU

Table 4: NFX150-S1 Models (*continued*)

	NFX150-S1	NFX150-S1E
RAM	16 GB	32 GB
Storage	200 GB SSD	200 GB SSD
Form Factor	1 RU	1 RU
Ports	Four 10/100/ 1000BASE-T RJ-45 ports which can be used as either access ports or uplinks	Four 10/100/ 1000BASE-T RJ-45 ports which can be used as either access ports or uplinks
	Two 1-Gigabit Ethernet/10-Gigabit Ethernet SFP+ ports	Two 1-Gigabit Ethernet/10-Gigabit Ethernet SFP+ ports
	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)
	One USB 3.0 port	One USB 3.0 port
LTE support	Yes (LTE expansion module)	Yes (LTE expansion module)
Expansion module support	Yes	Yes
Supported expansion modules  <b>NOTE:</b> You can install only one expansion module on the NFX150-S1 devices. The expansion module must be installed in the first slot, which is next to the chassis LEDs.	<ul style="list-style-type: none"> <li>• NFX-EM-6T2SFP— Expansion module with six 1-Gigabit Ethernet RJ-45 ports and two 1-Gigabit Ethernet SFP ports</li> <li>• NFX-LTE-AE—Expansion module with a LTE modem supporting the frequency bands in Europe and North America.</li> <li>• NFX-LTE-AA—Expansion module with a LTE modem supporting the frequency bands in Asia, Australia, and New Zealand.</li> </ul>	<ul style="list-style-type: none"> <li>• NFX-EM-6T2SFP— Expansion module with six 1-Gigabit Ethernet RJ-45 ports and two 1-Gigabit Ethernet SFP ports</li> <li>• NFX-LTE-AE—Expansion module with a LTE modem supporting the frequency bands in Europe and North America.</li> <li>• NFX-LTE-AA—Expansion module with a LTE modem supporting the frequency bands in Asia, Australia, and New Zealand.</li> </ul>

For more information on the NFX150 models and the expansion modules, see the [NFX150 Hardware Guide](#).

## Benefits and Uses of NFX150

The NFX150 Network Services Platform provides these benefits:

- Highly scalable, supporting multiple Juniper 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.
- Supports a variety of flexible deployments. A distributed services deployment model ensures high availability, performance, and compliance. The NFX150 provides an open framework that supports industry standards, protocols, and seamless API integration.
- In addition to Ethernet connections, Wireless WAN support through the LTE module provides more flexibility in deployments.
- Supports advanced security features such as IPsec connectivity, applications detection, and filtering for malicious traffic.
- The 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.
- Support of Open vSwitch (OVS) integrated with Data Plane Development Kit (DPDK) on NFX150 device offers better network packet throughput and lower latencies.

### RELATED DOCUMENTATION

[NFX150 Feature Overview](#) | 22

## NFX150 Feature Overview

### IN THIS SECTION

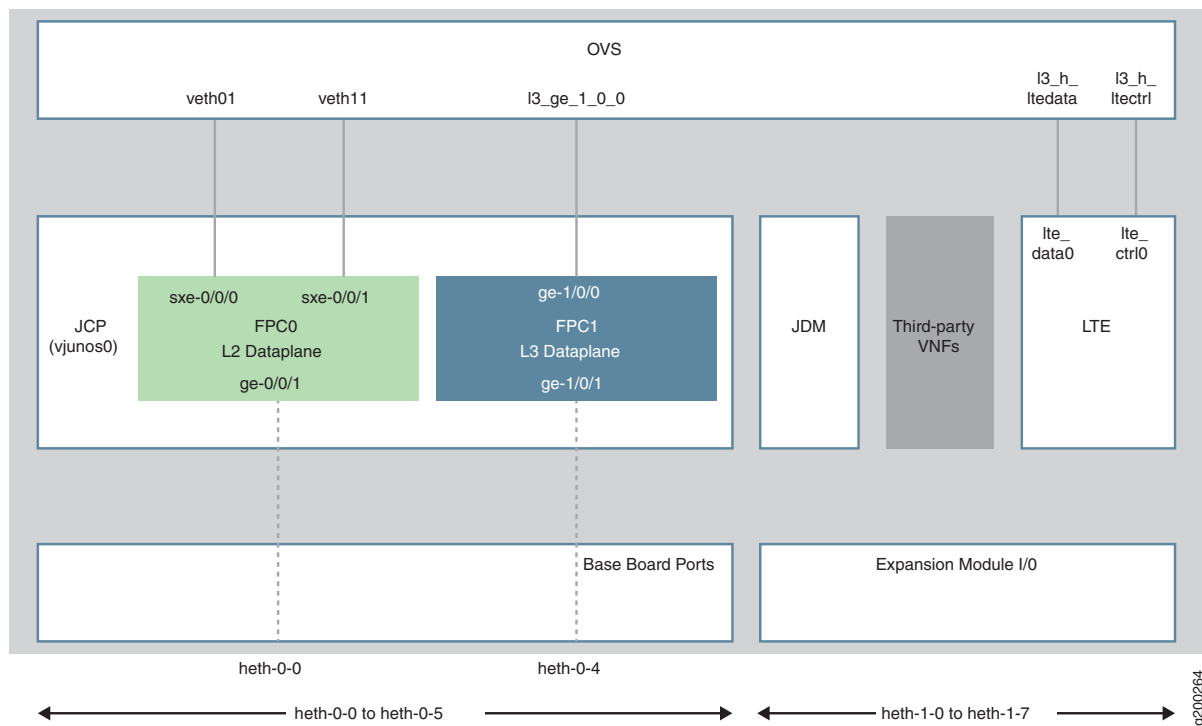
- [Software Architecture](#) | 23
- [Interfaces](#) | 24
- [Supported Features](#) | 29

## Software Architecture

The software architecture for the NFX150 is designed to provide a unified control plane that functions as a single management point.

Figure 3 on page 23 illustrates the architecture of the NFX150.

Figure 3: NFX150 Architecture



Key components of the system software include:

- VNF—A VNF is a consolidated offering that contains all the components required for supporting a fully virtualized networking environment. You can configure and use third-party VNFs in service chains.
- Junos Control Plane (JCP)—The JCP is the Junos VM running on the host OS, Wind River Linux. The JCP functions as the single point of management for all the components. The JCP controls the Layer 2

dataplane, which provide the Layer 2 services and the Layer 3 dataplane, which provides the Layer 3 to Layer 7 services.

In addition to chassis management, JCP enables:

- Configuration of advanced security features.
  - Management of guest virtualized network functions (VNFs) during their life cycle.
  - Installation of third-party VNFs.
  - Creation of VNF service chains.
  - Management of guest VNF images (their binary files).
  - Management of the system inventory and resource usage.
  - Management of the LTE interface.
- Juniper Device Manager (JDM)—An application container that manages VNFs and provides infrastructure services. The JDM functions in the background and users cannot access JDM directly.
  - L2 Dataplane—The Layer 2 dataplane that manages the Layer 2 traffic. The Layer 2 dataplane forwards the LAN traffic to the NFV backplane, Open vSwitch (OVS). The Layer 2 dataplane is mapped to the virtual FPC0 on the JCP. By default, all the 1-Gigabit Ethernet physical ports are mapped to the virtual interfaces on the Layer 2 dataplane.
  - L3 Dataplane—The Layer 3 dataplane that provides datapath functions for the Layer 3 to Layer 7 services. The Layer 3 dataplane is mapped to the virtual FPC1 on the JCP. By default, the two SFP+ ports on the NFX150 chassis are mapped to the virtual interfaces on the Layer 3 dataplane.
  - Linux—The host OS, WindRiver Linux. In Junos OS Release 18.1R1, the WindRiver Linux version is 8.
  - Open vSwitch (OVS) bridge—The OVS bridge is a VLAN-aware system bridge, which acts as the NFV backplane to which the VNFs and FPCs connect. Additionally, you can create custom OVS bridges to isolate connectivity between different VNFs.
  - LTE—A containerized driver that provides 4G LTE connectivity management. The LTE container is bound to the FPC1 for management.

## Interfaces

The interfaces on the NFX150 devices comprise of physical interfaces, virtual interfaces, and the LTE interface.



## Physical Interfaces

The physical interfaces represent the physical ports on the NFX150 chassis and expansion module. The physical interfaces comprise of network and management ports:

- Network ports—Four 1-Gigabit Ethernet ports and two 10-Gigabit Ethernet SFP+ ports function as network ports on the NFX150 chassis. The expansion modules consists of six 1-Gigabit Ethernet ports and two 1-Gigabit Ethernet SFP ports.

The network ports follow the naming convention **heth-slot number-port number**, where:

- heth denotes host Ethernet
- *slot number* is 0 for the chassis ports and 1 for the expansion module ports. The ports on the chassis are named as heth-0-x and the ports on the expansion module are named heth-1-x.
- *port number* is the number of the port on the chassis or expansion module

Each physical port has four virtual functions (VFs) enabled by default.

**NOTE:** You cannot map a VF from a port which is mapped to the Layer 2 dataplane.

- Management port—The NFX150 device has a dedicated management port labeled **MGMT** (fxp0), which functions as the out-of-band management interface. The fxp0 interface is assigned an IP address in the 192.168.1.1/24 network.

## Virtual Interfaces

The virtual FPCs running within the JCP, contain the virtual interfaces. The virtual interfaces on the NFX150 devices are categorized as follows:

- Virtual Layer 2 interfaces (FPC0)—Denoted as ge-0/0/x, where the value of x ranges from:
  - 0 to 3 for NFX150 devices without an expansion module
  - 0 to 11 for NFX150 devices with an expansion module

These interfaces are used to configure the following Ethernet switching features:

- Layer 2 switching of traffic, including support for both trunk and access ports
- Link Layer Discovery Protocol (LLDP)
- IGMP snooping
- Port Security features (MAC limiting, Persistent MAC learning)
- MVRP
- Ethernet OAM, CFM, and LFM

All the 1-Gigabit Ethernet physical ports (heth ports) are mapped to FPC0, by default.

- Virtual Layer 3 interfaces (FPC1)—Denoted as ge-1/0/x, where value of x ranges from 0 to 9. These interfaces are used to configure Layer 3 features such as routing protocols and QoS.

In an NFX150 device, you can configure any of the ge-1/0/x interfaces as in-band management interfaces. In in-band management, you configure a network interface as a management interface and connect it to the management device. You can configure any number of interfaces for in-band management by assigning an IPv4 or IPv6 address to each of the ports, and an in-band management VLAN.

**NOTE:** The NFX150 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.

- Virtual SXE Interfaces—Two static interfaces, sxe-0/0/0 and sxe-0/0/1, connect the FPC0 (Layer 2 dataplane) to the OVS backplane.

## LTE Interface

The NFX150 device models with LTE support can be configured for wireless WAN connectivity over 3G or 4G networks. The LTE physical interface uses the name cl-1/1/0. The dialer interface, dl0, is a logical interface, which is used to trigger calls.

## Interface Mapping

[Table 5 on page 27](#) summarizes the interfaces on the NFX150.

Table 5: Interfaces on the NFX150

Interface Name	Description
heth-0-0 to heth-0-5	<p>Physical ports on the front panel of the NFX150 device, which can be mapped to Layer 2 or Layer 3 interfaces, or VNFs.</p> <p>Ports heth-0-0 to heth-0-3 are 10 Mbps/100 Mbps/1 Gbps tri-speed copper ports.</p> <p>Ports heth-0-4 and heth-0-5 are 10 Gbps SFP+ ports</p> <p><b>For Junos OS Releases 18.1, 18.2 R1, and 18.3 R1:</b></p> <ul style="list-style-type: none"> <li>• Ports heth-0-0 to heth-0-3 are mapped to the LAN ports ge-0/0/0 to ge-0/0/3, respectively.</li> <li>• Ports heth-0-4 and heth-0-5 are mapped to the WAN ports ge-1/0/1 and ge-1/0/2, respectively.</li> </ul> <p><b>For Junos OS Release 18.2 R2</b></p> <ul style="list-style-type: none"> <li>• Ports heth-0-0, heth-0-1, and heth-0-2 are mapped to the LAN ports ge-0/0/0 to ge-0/0/2, respectively.</li> <li>• Port heth-0-4 is mapped to the LAN port ge-0/0/3.</li> </ul> <p>Ports heth-0-3 and heth-0-5 are mapped to the WAN ports ge-1/0/1 and ge-1/0/2, respectively.</p>
heth-1-0 to heth-1-7	<p>Physical ports on the expansion module of the NFX150-S1 device. These ports are mapped to the ge-0/0/n ports by default.</p> <p>Ports heth-1-0 to heth-1-5 are 10 Mbps/100 Mbps/1 Gbps tri-speed copper ports mapped to the LAN ports ge-0/0/4 to ge-0/0/9, respectively.</p> <p>Ports heth-1-6 and heth-1-7 are 1 Gbps SFP ports mapped to the LAN ports ge-0/0/10 and ge-0/0/11 respectively.</p>
ge-0/0/x	<p>Logical Layer 2 interfaces, which can be used for LAN connectivity. The values of x ranges from:</p> <ul style="list-style-type: none"> <li>• 0 to 3 for NFX150 devices without an expansion module</li> <li>• 0 to 11 for NFX150 devices with an expansion module</li> </ul>
ge-1/0/x	<p>A set of up to 10 logical Layer 3 interfaces. Each of these interfaces can have 4k sub-interfaces. The value of x ranges from 0 to 9.</p>
cl-1/1/0	<p>The LTE cellular interface, which carries the physical layer attributes.</p>
dl0	<p>The LTE dialer interface, which carries Layer 3 and security services. The security flow session contains the dl0 interface as the ingress or egress interface.</p>
st0	<p>Secure tunnel interface used for IPsec VPNs.</p>

**Table 5: Interfaces on the NFX150 (continued)**

Interface Name	Description
fxp0	The out-of-band management interface.

The list of supported transceivers for the NFX150 is located at <https://pathfinder.juniper.net/hct/product/>.

Table 7 on page 28 illustrates the default mapping between the physical and virtual interfaces on a NFX150 device.

**Table 6: Default Mapping of Physical Ports to Virtual Ports on NFX150 (for Junos OS Releases 18.1, 18.2 R1, and 18.3 R1)**

Physical Port	Virtual Interface (Layer 2 dataplane)	Virtual Interface (Layer 3 dataplane)
heth-0-0	ge-0/0/0	NA
heth-0-1	ge-0/0/1	NA
heth-0-2	ge-0/0/2	NA
heth-0-3	ge-0/0/3	NA
heth-0-4	NA	ge-1/0/1
heth-0-5	NA	ge-1/0/2

**Table 7: Default Mapping of Physical Ports to Virtual Ports on NFX150 (for Junos OS Releases 18.2 R2)**

Physical Port	Virtual Interface (Layer 2 dataplane)	Virtual Interface (Layer 3 dataplane)
heth-0-0	ge-0/0/0	NA
heth-0-1	ge-0/0/1	NA
heth-0-2	ge-0/0/2	NA
heth-0-3	NA	ge-1/0/1
heth-0-4	ge-0/0/3	NA
heth-0-5	NA	ge-1/0/2

Table 8 on page 29 illustrates the default mapping between the physical ports on the expansion module and the virtual interfaces.

Table 8: Default Mapping of Physical Ports to Virtual Ports for the Expansion Module

Physical Port	Virtual Port (Layer 2 dataplane)
heth-1-0	ge-0/0/4
heth-1-1	ge-0/0/5
heth-1-2	ge-0/0/6
heth-1-3	ge-0/0/7
heth-1-4	ge-0/0/8
heth-1-5	ge-0/0/9
heth-1-6	ge-0/0/10
heth-1-7	ge-0/0/11

**NOTE:** The expansion module ports are mapped to the Layer 2 dataplane interfaces by default. You can change the mapping to suit your requirement. Any of the ports on the chassis and expansion module can be mapped to the ge-1/0/x or ge-0/0/x interfaces. Any change in port mapping configuration will automatically reset the affected FPC.

## Supported Features

Table 9 on page 30 lists the Junos features supported on NFX150.

Table 9: Features Supported on NFX150

Junos OS Release	Routing	Security	Switching
18.1R1	<ul style="list-style-type: none"> <li>• BGP, OSPF, RIP, IS-IS,</li> <li>• MVRP</li> </ul>	<ul style="list-style-type: none"> <li>• NAT</li> <li>• ALG</li> <li>• IPSec</li> <li>• IPv6 NTP</li> <li>• IPv6 TACACS</li> <li>• CoS</li> <li>• Firewall filters</li> </ul>	<ul style="list-style-type: none"> <li>• LLDP</li> <li>• Port mirroring</li> <li>• IGMP/MLD snooping</li> <li>• MLD snooping</li> <li>• Persistent MAC learning</li> <li>• L2Rewrite</li> <li>• Native VLAN</li> </ul>
18.2 R1		<ul style="list-style-type: none"> <li>• Application Security</li> <li>• IDP</li> <li>• Integrated User Firewall</li> <li>• UTM</li> </ul>	

For more details on supported features, see [Feature Explorer](#).

## Performance Modes

Starting in Junos OS Release 19.1R1, NFX150 devices provide the following operational modes:

- Throughput mode—Provides maximum resources (CPU and memory) for Junos software and remaining resources, if any, for third-party VNFs. The default mode is throughput mode.
- 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.

## Licensing

For features or scaling levels that require a license, you must install and properly configure the license to meet the requirements for using the licensable feature or scale level. The device enables you to commit a configuration that specifies a licensable feature or scale without a license for a 30-day grace period. The grace period is a short-term grant that enables you to start using features in the pack or scale up to the system limits (regardless of the license key limit) without a license key installed. The grace period begins when the licensable feature or scaling level is actually used by the device (not when it is first committed).

In other words, you can commit licensable features or scaling limits to the device configuration, but the grace period does not begin until the device uses the licensable feature or exceeds a licensable scaling level.

For information about how to purchase software licenses, contact your Juniper Networks sales representative. Junos OS software implements an honor-based licensing structure and provides you with a 30-day grace period to use the feature without a license key installed. The grace period begins when you configure the feature and your device uses the licensed feature for the first time, but not necessarily when you install the license. After the grace period expires, the system generates system log messages saying that the feature requires a license. To clear the error message and use the licensed feature properly, you must install and verify the required license.

**NOTE:** Configurations might include both licensed and nonlicensed features. For these situations, the license is enforced up to the point where the license can be clearly distinguished. For example, an authentication-order configuration is shared by both Authentication, Authorization, and Accounting (AAA), which is licensed, and by Layer 2 Tunneling Protocol (L2TP), which is not licensed. When the configuration is committed, the device does not issue any license warnings, because it is not yet known whether AAA or L2TP is using the configuration. However, at runtime, the device checks for a license when AAA authenticates clients, but does not check when L2TP authenticates clients.

The device reports any license breach as a warning log message whenever a configuration is committed that contains a feature or scale limit usage that requires a license. Following the 30-day grace period, the device periodically reports the breach to syslog messages until a license is installed and properly configured on the device to resolve the breach.

**NOTE:** Successful commitment of a licensable feature or scaling configuration does not imply that the required licenses are installed or not required. If a required license is not present, the system issues a warning message after it commits the configuration.

**Table 10: NFX150 Junos Software Licenses**

License	Features	License SKU	Device Model
Base software (STD)	Layer 2 services, Layer 3 services, NAT, IPsec, stateful firewall	NFX150-C-STD	NFX150-C-S1 and NFX150-C-S1E
		NFX150-S-STD	NFX150-S1 and NFX150-S1E

Table 10: NFX150 Junos Software Licenses (*continued*)

License	Features	License SKU	Device Model
Advanced software (ADV)	Features in the base software plus AppFW, AppID, AppTrack, AppRoute	NFX150-C-ADV	NFX150-C-S1 and NFX150-C-S1E
		NFX150-S-ADV	NFX150-S1 and NFX150-S1E

## RELATED DOCUMENTATION

[Initial Configuration on NFX150 Devices | 37](#)
[Mapping Interfaces on NFX150 Devices | 59](#)

## Junos OS Releases Supported on NFX Series Hardware

The [Table 11 on page 32](#) provides details of Junos OS software releases supported on the NFX Series devices.

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

Table 11: 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	<a href="#">NFX150 Software Download Page</a>



Table 11: Supported Junos OS Releases on NFX Series Devices (continued)

NFX Series Platform	Supported Junos OS Release	Software Package	Software Downloads Page
NFX250	15.1X53-D45, 15.1X53-D47, 15.1X53-D470, and 15.1X53-D471	nfx-2  jinstall-host-nfx-2-flex-x86-64-<release-number>-secure-signed.tgz  install-media-host-usb-nfx-2-flex-x86-64-<release-number>-secure.img	<a href="#">NFX250 Software Download Page</a>
	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	
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	<a href="#">NFX350 Software Download Page</a>

RELATED DOCUMENTATION

| [NFX250 Overview](#)

# NFX Product Compatibility

IN THIS SECTION

- [Hardware Compatibility | 34](#)
- [Software Version Compatibility | 34](#)

## 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 12 on page 34](#).
- 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 12: 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

Table 12: Software Compatibility Details with vSRX and Cloud CPE Solution (*continued*)

NFX250 Junos OS Release	vSRX	Cloud CPE Solution
15.1X53-D47.4	15.1X49-D100.6	Cloud CPE Solution 3.0.1
15.1X53-D490	15.1X49-D143	Cloud CPE Solution 4.0
15.1X53-D495	15.1X49-D160	Cloud CPE Solution 4.1
15.1X53-D496	15.1X49-D170	Cloud CPE Solution 4.1
15.1X53-D45.3	15.1X49-D61	Not applicable
17.2R1	15.1X49-D78.3	Not applicable
17.3R1	15.1X49-D78.3	Not applicable
17.4R1	15.1X49-D78.3	Not applicable
15.1X53-D471	15.1X49-D143	Not applicable
18.1R1	18.1R1	Not applicable
18.1R2	18.1R2	Not applicable
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 NFX150 Devices | **37**

Zero Touch Provisioning on NFX Series Devices | **41**

Configuring the LTE Module on NFX Devices | **46**

Upgrading the Modem Firmware on NFX Devices Through Over-the-Air (OTA) | **55**

---

# Initial Configuration on NFX150 Devices

IN THIS SECTION

- [Factory-Default Settings | 37](#)
- [Enabling Basic Connectivity | 38](#)
- [Establishing the Connection | 40](#)

## Factory-Default Settings

The NFX150 device is shipped with the following factory-default settings:

Table 13: Security Policies

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

Table 14: Interface Mapping (for Junos OS Releases 18.1, 18.2 R1, and 18.3 R1)

Port Label	Interface	Virtual Interface	Security Zone	DHCP State	IP Address
0/0 to 0/3	heth-0-0 to heth-0-3	ge-0/0/0 to ge-0/0/3	trust	Server	192.168.2.1/24
0/4	heth-0-4	ge-1/0/1	untrust	Client	ISP assigned
0/5	heth-0-5	ge-1/0/2	untrust	Client	ISP assigned
MGMT	fxp0	N/A	N/A	N/A	192.168.1.1/24

Table 15: Interface Mapping (for Junos OS Releases 18.2 R2 and 18.4R1)

Port Label	Interface	Virtual Interface	Security Zone	DHCP State	IP Address
0/0 to 0/2	heth-0-0 to heth-0-2	ge-0/0/0 to ge-0/0/2	trust	Server	192.168.2.1/24

Table 15: Interface Mapping (for Junos OS Releases 18.2 R2 and 18.4R1) (continued)

Port Label	Interface	Virtual Interface	Security Zone	DHCP State	IP Address
0/3	heth-0-3	ge-1/0/1	untrust	Client	ISP assigned
0/4	heth-0-4	ge-0/0/3	trust	Server	192.168.2.1/24
0/5	heth-0-5	ge-1/0/2	untrust	Client	ISP assigned
MGMT	fxp0	N/A	N/A	N/A	192.168.1.1/24

Table 16: LTE Interfaces

Interface	Security Zone	IP Address
cl-1/1/0	N/A	N/A
dl0 (logical)	untrust	ISP assigned

The NFX150 device is shipped with the following services enabled by default: DHCP, HTTPS, and TFTP.

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

## Enabling Basic Connectivity

1. Ensure that the NFX150 device is powered on.
2. Connect to the console port:
  - a. Plug one end of the Ethernet cable into the console port on your NFX150 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:  
Baud rate—9600; Parity—N; 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 it 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** and enter the password **juniper123**. If the software completes booting before you connect to the console, you might need to press the **Enter** key for the prompt to appear:

**NOTE:** Starting with Junos OS Release 18.1R2 or later, the root password is not configured for initial configuration of the NFX150 devices.

```
login: root
password: juniper123
```

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 the Internet connection for 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
```

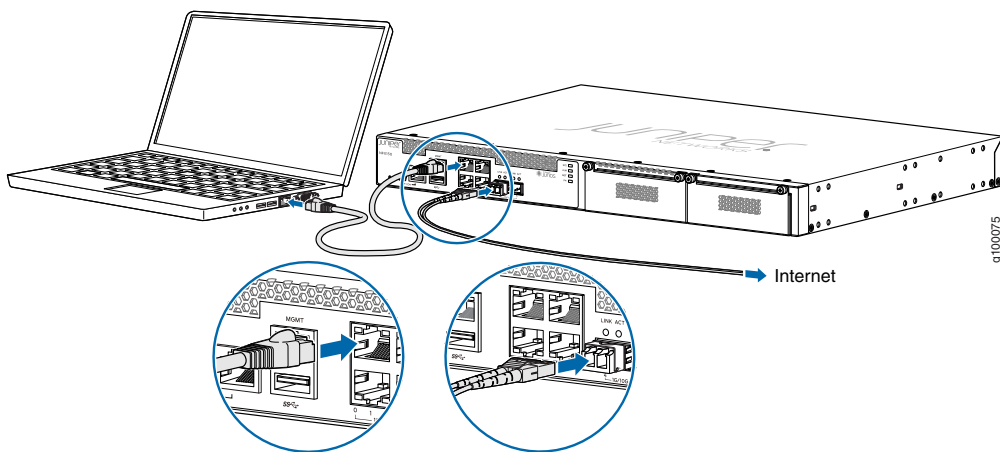
## Establishing the Connection

1. Connect the device to the Internet Service Provider (ISP) by using the following step:

**NOTE:** For information on interface mapping, see [Table 14 on page 37](#) and [Table 16 on page 38](#).

Connect one of the WAN ports to the ISP. The device is assigned an IP address by the ISP through DHCP.

Figure 4: Connecting the Interfaces on an NFX150-S1 Device



Optionally, you can obtain a SIM card from the ISP and connect the device through LTE.



**NOTE:** The LTE expansion module must be purchased separately.

2. Connect the laptop to one of the front panel LAN ports. The laptop is assigned an IP address by the DHCP server running on the interface.
3. Open a browser on your laptop, navigate to <https://www.juniper.net>, and verify your connectivity.

#### RELATED DOCUMENTATION

| *Installing and Configuring the LTE Module*

## Zero Touch Provisioning on NFX Series Devices

### IN THIS SECTION

- [Understanding Zero Touch Provisioning | 41](#)
- [Pre-staging an NFX Series Device | 42](#)
- [Provisioning an NFX Series Device | 44](#)
- [Provisioning an NFX Series Device Using Sky Enterprise | 45](#)

### 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.

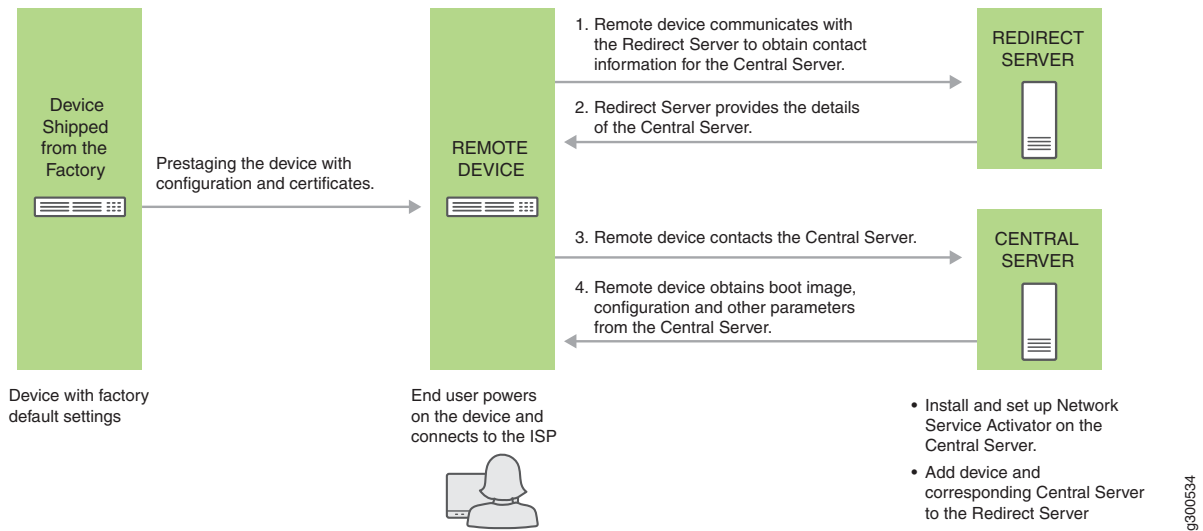
## Pre-staging an NFX Series Device

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 5 on page 43](#) illustrates the workflow of prestaging the NFX Series devices.

Figure 5: 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 NFX350 Devices*.
5. The device applies the prestage configuration and uses the certificates to authenticate the customer specific Redirect Server or Regional Server.
6. The Redirect Server or Regional Server sends the corresponding Central Server information to the device.
7. The device sends a provisioning request to the Central Server. The Central Server responds with the boot image and the configuration that is provisioned on the Central Server for that particular device.
8. The device fetches the boot image and configuration file from the Central Server.
9. The device upgrades to the boot image and applies the configuration to start the services and become operational.

To delete the prestage configuration and certificates, run:

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

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

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

To verify the prestage configuration and certificates, run:

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

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

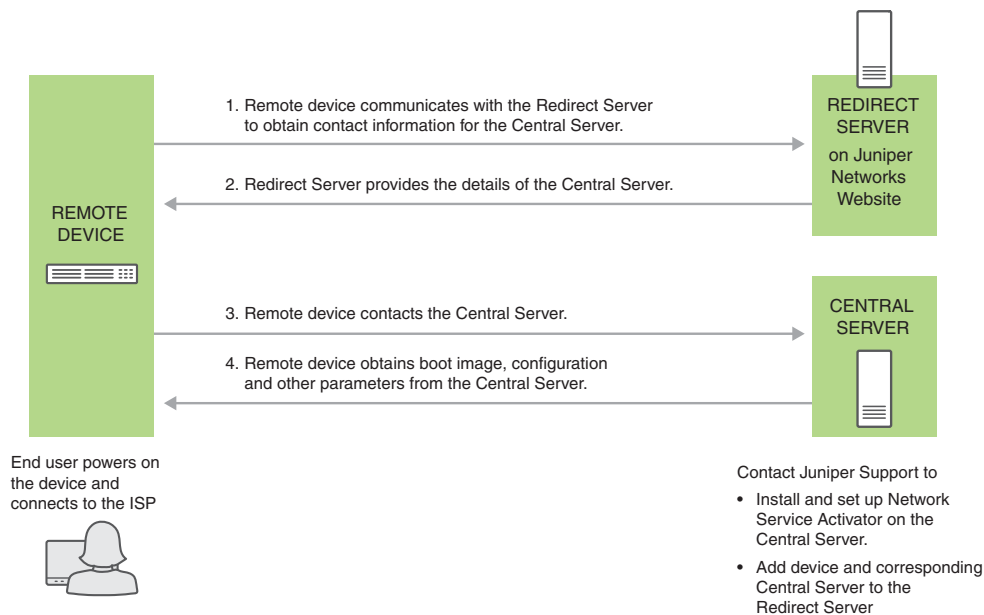
```
user@host>show system phone-home pre-stage
```

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

## Provisioning an NFX Series Device

Figure 6 on page 44 illustrates the workflow of the initial provisioning of NFX Series devices.

Figure 6: 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 6 on page 44](#) 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.

# Configuring the LTE Module on NFX Devices

## IN THIS SECTION

- [Configuring the LTE Module for Primary Mode | 47](#)
- [Configuring the LTE Module for Dial-on-Demand Mode | 49](#)
- [Configuring the LTE Module for Backup Mode | 52](#)

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
```

**NOTE:** To configure the LTE interface in a chassis cluster, configure the **redial-delay** to specify the delay (in seconds) to switch to the secondary cl interface when the active cl interface times out.

```
user@host# set interfaces dl0 unit 0 dialer-options redial-delay time-in-seconds
```

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.

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

To configure LTE interfaces on an NFX150 chassis cluster, specify the priority for the interfaces. The interface with the higher priority becomes the active interface.

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

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 being used as the active interface, run the **show dialer pools** command.

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
```

**NOTE:** To configure the LTE interface in a chassis cluster, configure the **redial-delay** to specify the delay (in seconds) to switch to the secondary cl interface when the active cl interface times out.

```
user@host# set interfaces dl0 unit 0 dialer-options redial-delay time-in-seconds
```

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.

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

To configure LTE interfaces on an NFX150 chassis cluster, specify the priority for the interfaces. The interface with the higher priority becomes the active interface.

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

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 being used as the active interface, run the **show dialer pools** command.

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
```

**NOTE:** To configure the LTE interface in a chassis cluster, configure the **redial-delay** to specify the delay (in seconds) to switch to the secondary cl interface when the active cl interface times out.

```
user@host# set interfaces dl0 unit 0 dialer-options redial-delay time-in-seconds
```

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.

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

To configure LTE interfaces on an NFX150 chassis cluster, specify the priority for the interfaces. The interface with the higher priority becomes the active interface.

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

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 being used as the active interface, run the **show dialer pools** command.

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
```

## RELATED DOCUMENTATION

[Upgrading the Modem Firmware on NFX Devices Through Over-the-Air \(OTA\) | 55](#)

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

Over-the-Air (OTA) firmware upgrade enables automatic and timely upgrade of modem firmware when new firmware versions are available. The OTA upgrade can be enabled or disabled on the LTE module. OTA is disabled by default.

**NOTE:** When upgrading the software on the NFX devices, the LTE firmware is also upgraded if the software contains a newer firmware version.

1. Enable OTA upgrade on the LTE module:

```
user@host > request modem wireless fota cl-1/1/0 enable
```

```
Set FOTA on modem succeeded
```

2. Initiate the firmware upgrade:

```
user@host > request modem wireless upgrade cl-1/1/0
```

```
Launch FOTA upgrade succeeded
```

3. Verify the firmware upgrade status:

```
user@host > show modem wireless firmware cl-1/1/0
```

```
LTE mPIM firmware details
  Product name: Junos LTE mPIM
  Serial number: D23F4349-10FA-41AA-A538-03648DE
  Hardware version: AcceleratedConcepts/porter
  Firmware version: 17.11.13
  MAC: 00:00:5e:00:53:82
  System uptime: 4632 seconds
Wireless modem firmware details
  Modem firmware version:
9999999_9904609_SWI9X30C_02.24.05.06_00_GENERIC_002.026_000
  Modem Firmware build date: 19/05/2017
  Card type: MC7455
  Modem manufacturer: Sierra Wireless, Inc
```

```

Hardware version: 1.0
Power & Temperature: Normal 3368 mV, Normal 29.00 C
OTA status
  State: Enabled
  New firmware available: No
Number of SIM: 1
Slot of active: 1
Status of SIM 1
  SIM state: SIM present
  Modem PIN security status: Disabled
  SIM status: SIM Okay
  SIM user operation needed: No Op
  Retries remaining: 3

```

#### 4. Check the LTE module connection status:

**user@host > show modem wireless network cl-1/1/0**

```

LTE Connection details
  Connected time: 2880
  IP: 10.12.219.210
  Gateway: 10.12.219.209
  DNS: 123.123.123.123
  IPv6: ::
  Gatewayv6: ::
  DNSv6: ::
  Input bps: 0
  Output bps: 0
  Bytes Received: 1952
  Bytes Transferred: 2164
  Packets Received: 10
  Packets Transferred: 20
Wireless Modem Network Info
  Current Modem Status: Connected
  Current Service Status: Normal
  Current Service Type: PS
  Current Service Mode: LTE
  Current Band: B3
  Network: UNICOM
  Mobile Country Code (MCC): 460
  Mobile Network Code (MNC): 1
  Location Area Code (LAC): 65534
  Routing Area Code (RAC): 0

```



```
Cell Identification: 239907605
Access Point Name (APN): 3gnet
Public Land Mobile Network (PLMN): CHN-UNICOM
Physical Cell ID (PCI): 452
International Mobile Subscriber Identification (IMSI): *****
International Mobile Equipment Identification (IMEI/MEID): *****
Integrate Circuit Card Identity (ICCID): 89860117811046631207
Reference Signal Receiving Power (RSRP): -71
Reference Signal Receiving Quality (RSRQ): -8
Signal to Interference-plus-Noise Ratio (SINR): 19
Signal Noise Ratio (SNR): 22
Energy per Chip to Interference (ECIO): 0
```

## RELATED DOCUMENTATION

[Configuring the LTE Module on NFX Devices](#) | 46

# 3

CHAPTER

## Configuring Interfaces

---

Mapping Interfaces on NFX150 Devices | **59**

Configuring the In-Band Management Interface | **61**

ADSL2 and ADSL2+ Interfaces on NFX150 Devices | **62**

VDSL2 Interfaces on NFX150 Devices | **67**

---

# Mapping Interfaces on NFX150 Devices

## IN THIS SECTION

- Mapping Physical Interfaces to Virtual Interfaces on NFX150 Devices | 59
- Mapping Physical Ports to VNF Interfaces Through SR-IOV | 60
- Mapping Layer 3 Dataplane Interfaces to OVS | 60

## Mapping Physical Interfaces to Virtual Interfaces on NFX150 Devices

The NFX150 devices are shipped with the default mapping as described in [“NFX150 Feature Overview” on page 22](#). You can change the mapping to suit your needs. Any of the physical ports can be mapped to the ge-0/0/x or ge-1/0/x interfaces. Any change in port mapping configuration will automatically reset the affected FPC.

**NOTE:** On NFX150 devices, the link does not come up if 1G SFP port is connected from heth-0-4 and heth-0-5 ports to a peer device. As a workaround, disable the auto-negotiation for the interface connected to the NFX150 device on the remote device.

- To map a physical port to a virtual interface:

```
root# set vmhost virtualization-options interfaces virtual-interface-name mapping interface
physical-interface-name
```

For example, to map the physical port heth-0-2 to the Layer 2 dataplane (FPC0) interface ge-0/0/3, the command is as follows:

```
root# set vmhost virtualization-options interfaces ge-0/0/3 mapping interface heth-0-2
```

For example, to map the physical port heth-0-5 to the Layer 3 dataplane (FPC1) interface ge-1/0/2, the command is as follows

```
root# set vmhost virtualization-options interfaces ge-1/0/2 mapping interface heth-0-5
```

- To map a physical port to two FPC1 interfaces:

```
root# set vmhost virtualization-options interfaces virtual-interface-name1 mapping interface
physical-interface-name
root# set vmhost virtualization-options interfaces virtual-interface-name2 mapping interface
physical-interface-name
```

For example, to map the physical port heth-0-1 to the Layer 3 data plane(FPC1) interfaces ge-1/0/1 and ge-1/0/2:

```
root# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface heth-0-1
root# set vmhost virtualization-options interfaces ge-1/0/2 mapping interface heth-0-1
```

## Mapping Physical Ports to VNF Interfaces Through SR-IOV

You can configure Layer 3 services using single-root I/O virtualization (SR-IOV). To map a VNF interface to a physical interface by using the SR-IOV virtual function, use the following command:

```
root# set virtual-network-functions vnf-name interfaces interface-name mapping interface
physical-interface-name virtual-function
```

## Mapping Layer 3 Dataplane Interfaces to OVS

The ge-1/0/0 interface is always mapped to the service chaining backplane (OVS). To map additional Layer 3 dataplane interfaces to the system bridge, use the following command:

```
root# set vmhost virtualization-options interfaces interface-name
```

### RELATED DOCUMENTATION

[Configuring the In-Band Management Interface](#) | 61

# Configuring the In-Band Management Interface

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.

You can configure any of the ge-1/0/x ports, where x ranges from 0 to 9, as in-band management interfaces. In-band management can be configured using either a LAN port (FPC0) or a WAN port (FPC1).

To configure a WAN port for in-band management:

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

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

2. Configure the IP address for the in-band management interface:

```
root@host# set interfaces interface-name unit 0 family inet address address/prefix-length
```

**NOTE:** The ge-1/0/x port selected for configuration must be the same port that is mapped to the physical port (heth) being used for management connectivity.

3. Configure VLAN tagging:

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

To configure a LAN port for in-band management:

1. Configure the management VLAN:

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

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

```
root@host# set interfaces ge-0/0/x unit 0 family ethernet-switching vlan members mgmt-vlan
root@host# set interfaces sxe-0/0/[01] unit 0 family ethernet-switching vlan members mgmt-vlan
```

3. Configure VLAN tagging:

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

# ADSL2 and ADSL2+ Interfaces on NFX150 Devices

IN THIS SECTION

- [ADSL Interface Overview | 62](#)
- [Configuring ADSL SFP Interface Using VLANs on NFX150 Network Services Platform | 63](#)
- [Configuring ADSL SFP Interface Without Using VLANs on NFX150 Network Services Platform | 65](#)

## ADSL Interface Overview

IN THIS SECTION

- [ADSL2 and ADSL2+ | 63](#)

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

ADSL transmission is asymmetric because the downstream bandwidth is typically greater than the upstream bandwidth. The typical bandwidths of ADSL2 and ADSL2+ circuits are defined in [Table 17 on page 62](#).

Table 17: Standard Bandwidths of DSL Operating Modes

Operating Modes	Upstream	Downstream
ADSL2	1–1.5 Mbps	12–14 Mbps

Table 17: Standard Bandwidths of DSL Operating Modes (*continued*)

Operating Modes	Upstream	Downstream
ADSL2+	1–1.5 Mbps	24–25 Mbps

ADSL2 and ADSL2+ support the following standards:

- LLC SNAP bridged 802.1q
- VC MUX bridged

The ADSL Mini-PIM facilitates a maximum of 10 virtual circuits on supported security devices.

Supported security devices with Mini-PIMs can use PPP over Ethernet over ATM (PPPoEoA) and PPP over ATM (PPPoA) 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.

## Configuring ADSL SFP Interface Using VLANs on NFX150 Network Services Platform

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

To configure ADSL SFP interfaces on NFX150 devices:

1. Connect to the host.

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

2. Configure the WAN side front panel port with vlan-tagging.

```
user@host# set interfaces virtual-interface-name vlan-tagging
```

```
user@host# set interfaces ge-1/0/1 vlan-tagging
```

3. Configure a VLAN for the WAN side front panel port.

```
user@host# set interfaces virtual-interface-name unit 0 vlan-id vlan-id
```

```
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 152
```

4. Configure the WAN side front panel port with an IP address.

```
set interfaces virtual-interface-name unit 0 family inet address ip-address
```

```
user@host# set interfaces ge-1/0/1 unit 0 family inet address 152.1.1.152/24
```

5. Configure the physical (heth) interface with ADSL SFP options.

```
user@host# set vmhost interfaces physical-interface-name dsl-sfp-options adsl-options vpi vpi vci vci
encap encapsulation
```

```
user@host# set vmhost interfaces heth-0-4 dsl-sfp-options adsl-options vpi 8 vci 36 encap
llcsnap-bridged-802.1q
```

**NOTE:**

- The default value for **encap** is **llcsnap-bridged-802.1q**.
- The ADSL SFP interface is enabled only if you configure **vci** and **vpi**. By default, **vci** and **vpi** are disabled.



6. Map the physical (heth) interfaces to the virtual (ge) interfaces.

```
user@host# set vmhost virtualization-options interfaces virtual-interface-name mapping interface
physical-interface-name
```

```
user@host# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface heth-0-4
```

7. Commit the configuration.

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

To verify the configuration, enter the **show interfaces heth-0-4** command.

[edit]

```
user@host# show interfaces heth-0-4
```

```
Physical interface: heth-0-4, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Fiber, MTU: 9192, ADSL2P mode, Speed:
1Gbps, Duplex: Full-duplex, Auto-negotiation: Disabled
  ADSL status:
    Modem status   : Showtime (Adsl2plus)
    DSL mode       :      Auto      Annex A
  Device flags    : Present Running
  Current address: 58:00:bb:ac:c8:51, Hardware address: 58:00:bb:ac:c8:51
```

## Configuring ADSL SFP Interface Without Using VLANs on NFX150 Network Services Platform

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

To configure ADSL SFP interfaces on NFX150 devices:

1. Connect to the host.

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

2. Configure the WAN side front panel port with an IP address.

**set interfaces *virtual-interface-name* unit 0 family inet address *ip-address***

```
user@host# set interfaces ge-1/0/1 unit 0 family inet address 153.1.1.153/24
```

3. Configure the physical (heth) interface with ADSL SFP options.

**user@host# set vmhost interfaces *physical-interface-name* dsl-sfp-options adsl-options vpi *vpi* vci *vci* encap *encapsulation***

```
user@host# set vmhost interfaces heth-0-4 dsl-sfp-options adsl-options vpi 8 vci 36 encap
llcsnap-bridged-802.1q
```

**NOTE:**

- The default value for **encap** is **llcsnap-bridged-802.1q**.
- The ADSL SFP interface is enabled only if you configure **vci** and **vpi**. By default, **vci** and **vpi** are disabled.

4. Map the physical (heth) interfaces to the virtual (ge) interfaces.

**user@host# set vmhost virtualization-options interfaces *virtual-interface-name* mapping interface *physical-interface-name***

```
user@host# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface heth-0-4
```

5. Commit the configuration.

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

To verify the configuration, enter the **show interfaces heth-0-4** command.

[edit]

user@host# **show interfaces heth-0-4**

```
Physical interface: heth-0-4, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Fiber, MTU: 9192, ADSL2P mode, Speed:
1Gbps, Duplex: Full-duplex, Auto-negotiation: Disabled
  ADSL status:
    Modem status   : Showtime (Adsl2plus)
    DSL mode       :      Auto      Annex A
  Device flags    : Present Running
  Current address: 58:00:bb:ac:c8:51, Hardware address: 58:00:bb:ac:c8:51
```

## RELATED DOCUMENTATION

# VDSL2 Interfaces on NFX150 Devices

## IN THIS SECTION

- [VDSL Interface Overview | 67](#)
- [VDSL2 Network Deployment Topology | 68](#)
- [VDSL2 Interface Support on NFX Series Devices | 70](#)
- [Configuring VDSL SFP Interface Using VLANs on NFX150 Network Services Platform | 72](#)
- [Configuring VDSL SFP Interface Without Using VLANs on NFX150 Network Services Platform | 74](#)

## VDSL Interface Overview

## IN THIS SECTION

- [VDSL2 Vectoring Overview | 68](#)

Very-high-bit-rate digital subscriber line (VDSL) technology is part of the xDSL 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:

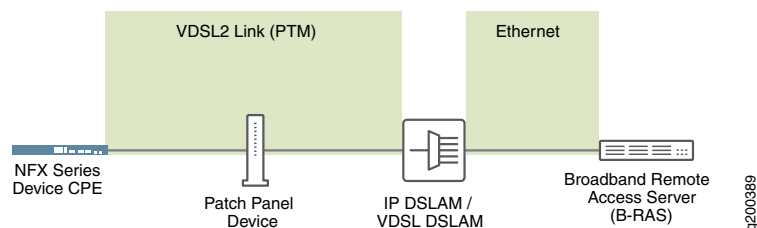
1. Connect an end-user device such as a LAN, hub, or PC through an Ethernet interface to the customer premises equipment (CPE) (for example, an NFX device).
2. Connect the CPE to a DSLAM.
3. The VDSL2 interface uses either Gigabit Ethernet or fiber as second mile to connect to the Broadband Remote Access Server (B-RAS) as shown in [Figure 7 on page 69](#).
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 8 on page 70](#).

**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.

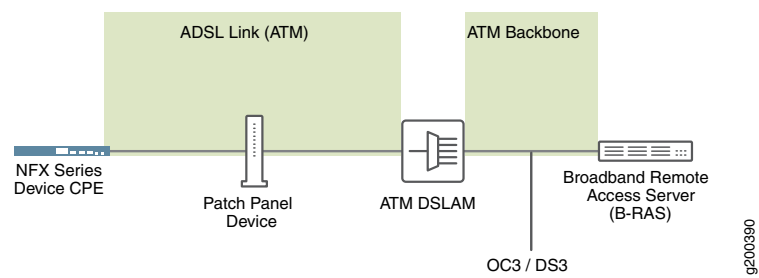
[Figure 7 on page 69](#) shows a typical VDSL2 network topology.

**Figure 7: Typical VDSL2 End-to-End Connectivity and Topology Diagram**



[Figure 8 on page 70](#) shows a backward-compatible ADSL topology using ATM DSLAM.

Figure 8: Backward-Compatible ADSL Topology (ATM DSLAM)



## VDSL2 Interface Support on NFX Series Devices

The VDSL2 interface is supported on the NFX Series devices listed in [Table 18 on page 70](#). (Platform support depends on the Junos OS release in your installation.)

Table 18: VDSL2 Annex A and Annex B Features

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

**NOTE:** Only one JNP-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 heth-0-4 and heth-0-5.

### 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 19 on page 71](#) lists the different profiles supported on the VDSL2 interfaces and their properties.

**Table 19: 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)

## VDSL2 Interfaces Supported Features

The following features are supported on the VDSL2 interfaces:

- ADSL2 and ADSL2+ backward compatibility with Annex A, Annex M support
- PTM or EFM (802.3ah) support
- Operation, Administration, and Maintenance (OAM) support for ADSL2 and ADSL2+ modes
- Multilink Point-to-Point Protocol (MLPPP) (supported only when the VDSL2 Mini-PIM is operating in ADSL2 mode)
- MTU size of 1514 bytes (maximum) in VDSL2 mode and 1496 bytes in ADSL mode.
- Support for maximum of 10 permanent virtual connections (PVCs) (only in ADSL2 and ADSL2+ mode)

## Configuring VDSL SFP Interface Using VLANs on NFX150 Network Services Platform

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

To configure VDSL SFP interfaces on NFX150 devices:

1. Connect to the host.

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

2. Configure the WAN side front panel port with vlan-tagging.

```
user@host# set interfaces virtual-interface-name vlan-tagging
```

```
user@host# set interfaces ge-1/0/1 vlan-tagging
```

3. Configure a VLAN for the WAN side front panel port.

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



```
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 151
```

4. Configure the WAN side front panel port with an IP address.

```
user@host# set interfaces virtual-interface-name unit logical-unit-number family inet address ip-address
```

```
user@host# set interfaces ge-1/0/1 unit 0 family inet address 151.1.1.151/24
```

5. Configure the physical (heth) interface with VDSL SFP options, VDSL profile, and carrier settings on the VDSL SFP interface.

```
user@host# vmhost interfaces physical-interface-name dsl-sfp-options vdsl-options profile profile  
carrier carrier
```

```
user@host# set vmhost interfaces heth-0-4 dsl-sfp-options vdsl-options profile auto carrier auto
```

**NOTE:**

- The default value for **vdsl-options profile** is **auto**. The value **auto** supports all profiles ranging from 8a to 17a.
- The default value for **vdsl-options carrier** is **auto**. The value **auto** includes a43 and b43.

6. Map the physical (heth) interfaces to the virtual (ge) interfaces.

```
user@host# set vmhost virtualization-options interfaces virtual-interface-name mapping interface  
physical-interface-name
```

```
user@host# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface heth-0-4
```

7. Commit the configuration.

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

To verify the configuration, enter the **show interfaces heth-0-4** command.

[edit]

user@host# **show interfaces heth-0-4**

```
Physical interface: heth-0-4, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Fiber, MTU: 9192, VDSL2 mode, Speed:
1Gbps, Duplex: Full-duplex, Auto-negotiation: Disabled
  VDSL status:
    Modem status   : Showtime (Auto)
    VDSL profile   : Profile-8b
  Device flags    : Present Running
  Current address: d8:b1:22:33:5e:51, Hardware address: d8:b1:22:33:5e:51
```

## Configuring VDSL SFP Interface Without Using VLANs on NFX150 Network Services Platform

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

To configure VDSL SFP interfaces on NFX150 devices:

1. Connect to the host.

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

2. Configure the WAN side front panel port with an IP address.

user@host# **set interfaces *virtual-interface-name* unit *logical-unit-number* family inet address *ip-address***

```
user@host# set interfaces ge-1/0/1 unit 0 family inet address 151.1.1.151/24
```

3. Configure the physical (heth) interface with VDSL SFP options, VDSL profile, and carrier settings on the VDSL SFP interface.

```
user@host# vmhost interfaces physical-interface-name dsl-sfp-options vdsl-options profile profile
carrier carrier
```

```
user@host# set vmhost interfaces heth-0-4 dsl-sfp-options vdsl-options profile auto carrier auto
```

**NOTE:**

- The default value for **vdsl-options profile** is **auto**. The value **auto** supports all profiles ranging from 8a to 17a.
- The default value for **vdsl-options carrier** is **auto**. The value **auto** includes a43 and b43.

4. Map the physical (heth) interfaces to the virtual (ge) interfaces.

```
user@host# set vmhost virtualization-options interfaces virtual-interface-name mapping interface
physical-interface-name
```

```
user@host# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface heth-0-4
```

5. Commit the configuration.

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

To verify the configuration, enter the **show interfaces heth-0-4** command.

[edit]

```
user@host# show interfaces heth-0-4
```

```
Physical interface: heth-0-4, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Fiber, MTU: 9192, VDSL2 mode, Speed:
1Gbps, Duplex: Full-duplex, Auto-negotiation: Disabled
  VDSL status:
    Modem status   : Showtime (Auto)
    VDSL profile   : Profile-8b
  Device flags    : Present Running
  Current address: d8:b1:22:33:5e:51, Hardware address: d8:b1:22:33:5e:51
```

## RELATED DOCUMENTATION

.

# 4

CHAPTER

## Configuring Security

---

IP Security on NFX Devices | **78**

UTM on NFX Devices | **87**

Application Security on NFX Devices | **88**

Intrusion Detection and Prevention on NFX Devices | **89**

Integrated User Firewall Support on NFX Devices | **89**

---

# IP Security on NFX Devices

## IN THIS SECTION

- [Overview | 78](#)
- [Configuring Security | 79](#)

## 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 20 on page 79](#) lists the IPsec features supported on NFX Series devices.

**Table 20: IPsec Features Supported on NFX Series Devices**

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

## Configuring Security

### IN THIS SECTION

- [Configuring Interfaces | 80](#)
- [Configuring Routing Options | 81](#)

- [Configuring Security IKE | 81](#)
- [Configuring Security IPsec | 84](#)
- [Configuring Security Policies | 86](#)
- [Configuring Security Zones | 86](#)

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

## Configuring Interfaces

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

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

To configure interfaces, complete the following steps:

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

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

2. Enable VLAN tagging support on the logical interface:

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

3. Assign a VLAN ID to the logical interface:

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

4. Assign an IPv4 address to the logical interface:

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

5. Assign an IPv6 address to the logical interface:



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

## Configuring Routing Options

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

Protocol-independent routing properties include:

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

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

1. Configure RIB and static route:

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

2. Configure static route:

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

## Configuring Security IKE

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

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

To configure IKE traceoptions, complete the following steps:

1. Specify the maximum size of the trace file:

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

2. Specify the parameters to trace information for IKE:

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

3. Specify the level of trace information for IKE:

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

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

To configure IKE proposal, complete the following steps:

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

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

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

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

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

3. Configure an authentication algorithm for the IKE proposal:

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

4. Define an encryption algorithm for the IKE proposal:

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

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

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

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

To configure IKE policy, complete the following steps:

1. Define an IKE policy with first phase mode:

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

2. Define a set of IKE proposals:

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

3. Define a pre-shared key for IKE:

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

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

To configure IKE gateway, complete the following steps:

1. Configure an IKE gateway with an IKE policy:

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

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

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

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

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

4. Configure the local IKE identity:

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

5. Configure the remote IKE identity:

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

6. Configure an external interface for IKE negotiations:

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

7. Configure username of the client:

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

8. Configure password of the client:

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

## Configuring Security IPsec

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

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

To configure an IPsec proposal, complete the following steps:

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

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

2. Define an authentication algorithm for the IPsec proposal:

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

3. Define an encryption algorithm for the IPsec proposal:

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

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

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

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

To configure IPsec policies, complete the following steps:

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

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

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

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

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

To configure IPsec VPN, complete the following steps:

1. Define an IKE gateway for the IPsec VPN:

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

2. Define an IPsec policy for the IPsec VPN:

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

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

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

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

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

5. Define a criteria to establish IPsec VPN tunnels:

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

## Configuring Security Policies

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

To configure security policies, complete the following steps:

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

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

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

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

3. Configure security policy application:

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

4. Set security policy match criteria:

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

## Configuring Security Zones

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

To configure security zones, complete the following steps:

1. Configure security zones with system services:

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

2. Define protocols for security zones:

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

### 3. Configure interfaces for security zones:

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

## 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*.
- **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

## 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



# 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](#) | 87

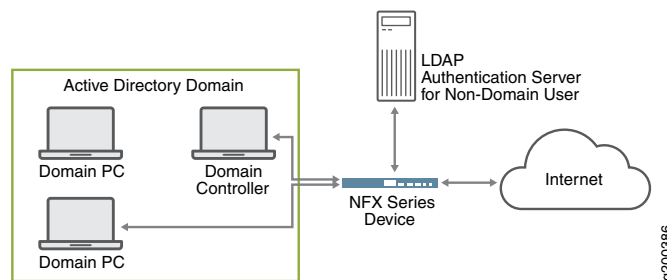
# Integrated User Firewall Support on NFX Devices

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 9 on page 90 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 9: 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 | 87

# 5

CHAPTER

## Configuring VNFs

---

Prerequisites to Onboard Virtual Network Functions on NFX150 Devices | **92**

Configuring VNFs on NFX150 Devices | **92**

Managing VNFs on NFX Series Devices | **104**

Configuring Analyzer VNF and Port-mirroring | **110**

---

# Prerequisites to Onboard Virtual Network Functions on NFX150 Devices

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

The number of VNFs that you can onboard on the device depends on the availability of system resources such as the number of CPUs and system memory.

Before you onboard the VNFs, it is recommended to check the available system resources such as CPUs, memory, and storage for VNFs. For more information, see [“Configuring VNFs on NFX150 Devices” on page 92](#).

## Prerequisites for VNFs

To instantiate VNFs, the NFX device supports:

- KVM based hypervisor deployment
- OVS or Virtio interface drivers
- raw or qcow2 VNF file types
- (Optional) SR-IOV
- (Optional) CD-ROM and USB configuration drives
- (Optional) Hugepages for memory requirements

## Configuring VNFs on NFX150 Devices

### IN THIS SECTION

- [Load the VNF Image | 93](#)
- [Prepare the Bootstrap Configuration | 93](#)
- [Allocate CPUs for a VNF | 95](#)
- [Allocate Memory for a VNF | 96](#)

- (Optional) Attach a Config Drive to the VNF | 97
- Configuring VNF Interfaces and VLANs | 99
- Configuring VNF Storage Devices | 101
- Instantiating a VNF | 102
- Verify that the VNF Instantiated Successfully | 103
- Instantiating a VNF Using an XML Descriptor File | 104

The NFX150 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.

To configure a VNF, log in to the JCP and perform the following tasks:

## Load the VNF Image

To load a VNF image on the device from a remote location, use the **file-copy** command. Alternatively, you can use the NETCONF command **file-put**, to load a VNF image.

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

## Prepare the Bootstrap Configuration

You can bootstrap a VNF using an attached config drive that contains a bootstrap-config 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 97](#).

**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 21 on page 95 lists the CPUs available for VNF usage for the NFX150 models.

**Table 21: CPUs Available for VNF Usage (Junos OS 19.1R1 Release)**

Model	CPUs Available for VNF Usage		
	Throughput Mode	Hybrid Mode	Compute Mode
NFX150-C-S1	0	1	2
NFX150-C-S1-AE	0	1	2
NFX150-C-S1-AA	0	1	2
NFX150-C-S1E-AE	0	1	2
NFX150-C-S1E-AA	0	1	2
NFX150-S1	0	2	4
NFX150-S1E	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.

To check the CPU availability and its status, use the following command:

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

To specify the number of virtual CPUs that are required for a VNF, use the following commands:

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

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

2. Pin 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 either be a number or a range. By default, a VNF is allocated with one virtual CPU that is not pinned to any physical CPU.

**NOTE:** You cannot change the CPU configuration of a VNF when the VNF is in **running** state. Restart the VNF for changes to take effect.

To enable hardware-virtualization or hardware-acceleration for VNF CPUs, type the following command:

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

## Allocate Memory for a VNF

On NFX150 devices running Junos OS Release 18.1R1, enabling hugepages for VNFs and pre-reserving of hugepages are not supported.

[Table 22 on page 97](#) lists the possible memory availability for VNF usage for the NFX150 models.



Table 22: Memory Availability for VNF Usage

Model	Memory Availability for VNF Usage (Junos OS 19.1R1 Release)
NFX150-C-S1	1 GB
NFX150-C-S1-AE	1 GB
NFX150-C-S1-AA	1 GB
NFX150-C-S1E-AE	9 GB
NFX150-C-S1E-AA	9 GB
NFX150-S1	7 GB
NFX150-S1E	23 GB

To specify the maximum primary memory that the VNF can use, enter the following command:

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

**NOTE:** You cannot change the memory configuration of a VNF if the VNF is in the running state. Restart the VNF for changes to take effect.

## (Optional) Attach a Config Drive to the VNF

Add files and template to the config drive.

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

```
user@host# set virtual-network-functions vnf-name config-data source file source-file1
user@host# set virtual-network-functions vnf-name config-data source file source-file2
```

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

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

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

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

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

3. Specify the device name and type to add in the config drive:

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

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

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

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

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

4. Commit the configuration:

```
user@host# commit
```

Add a directory to the config drive.

1. Specify the source directory to add in the config drive:

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

2. Specify the device name and type to add in the config drive:

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

3. 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](#) command output message.

## Configuring VNF Interfaces and VLANs

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

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

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

*vlan-id* is optional and it is the port VLAN ID.

- To create a VLAN:

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

- To attach a VNF interface to a VLAN:

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

### NOTE:

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

Starting in Junos OS Release 19.2R1, changes to the default MAC flooding behavior of the VNF interfaces improve the performance of multicast traffic. If a VNF interface is not attached to a VLAN, drop flow is not configured. The interface functions as a trunk port that can receive and forward the VLAN traffic. If the destination MAC address is known, the interface forwards the traffic to the destined port. If the MAC address is unknown, or if it is broadcast or multicast traffic, the interface forwards the traffic to all the ports in the same VLAN and to the ports that do not have a VLAN assigned.

In earlier releases, if a VNF interface is not attached to a VLAN, drop flow is configured and the VNF interface drops the outgoing traffic.

**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 delete a VNF interface:

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

**NOTE:**

- To delete an interface, you must stop the VNF, delete the interface, and start the VNF.
- After attaching or detaching a virtual function, you must restart the VNF for changes to take effect.
- eth0 and eth1 are reserved for default VNF interfaces that are connected to the internal network and 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 convention. VNF interfaces that are configured in 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 JCP and name them accordingly.

- Starting in Junos OS Release 19.2R1, you can manually disable the VNF interfaces (eth0 through eth9) on the OVS or custom bridge by issuing the following command:

```
user@host# set virtual-network-functions vnf-name interfaces interface-name link disable
```

**NOTE:**

- If a link in a cross-connect configuration is down, then the cross-connect will also be down.
- You cannot manually disable the VF interfaces on the VNF.
- The eth0 and eth1 interfaces, which function as management interfaces, can be disabled only if the **no-default-interfaces** option is configured.

To identify a disabled link, issue the following command:

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

For example, the following output shows that the eth2 link on the centos VNF is disabled. Note that the output is truncated to provide only the details relevant to the disabled link.

```
Network Name : ovs-sys-br

Interface : centos_eth2
Type : virtual ethernet, Link type : Full-Duplex, MAC : fe:b6:c2:cc:66:a0
MTU : [], Link State :down, Admin State : down
Native Vlan ID : None, Vlan mode : Access, Vlan Members : None
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
Rx-packets :      0
Rx-drops    :      0
Rx-errors   :      0
Tx-packets  :    348
Tx-drops    :   42948
Tx-errors   :      0
```

## Configuring VNF Storage Devices

The NFX150 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, enter the following command:

```
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 hdx or sdx or vdx. For example, hdb, sdc, vdb and so on.

To add a virtual USB storage device, enter the following command:

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

To attach an additional hard disk, enter the following command:

```
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 a hard disk from the VNF, enter the following command:

```
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 changes to take effect. The CD detach operation fails if the device is in use within the VNF.
- 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 in **running** state.
- You must save the source file in the **/var/public** directory and the file must have read and write permission for all users.

## Instantiating a VNF

You can instantiate a VNF by configuring the VNF name, and specifying either the path to an XML descriptor file or to an image.

While instantiating a VNF with an image, two VNF interfaces are added by default. These interfaces are required for management and internal network. The target Peripheral Component Interconnect (PCI) addresses, such as 0000:00:03:0 and 0000:00:04:0 are reserved for these interfaces.

- To instantiate a VNF using an image:

**NOTE:** Only Qcow2 and Raw image types are supported.

```
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 configuring VNFs, do not use VNF names in the format `vnfn`—for example, `vnf1`, `vnf2`, and so on. Configurations containing 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, and it is recommended to allow the system to allocate a UUID for the VNF.

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

## Verify that the VNF Instantiated Successfully

Verify that the VNF instantiated successfully by using the following command:

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

ID	Name	State	Liveliness
1	vjunos0	Running	alive

2	centos1	Running	alive
3	centos2	Running	alive
11057	LTE	Running	alive

The **Liveliness** output field of a VNF indicates whether the IP address of the VNF is reachable or not reachable over the internal management network. The default IP address of the liveliness bridge is 192.0.2.1/24.

## Instantiating a VNF Using an XML Descriptor File

You can also instantiate a VNF by using an XML descriptor file. You must place the XML file in the /var/public/ directory.

```
user@host# set virtual-network-functions vnf-name init-descriptor file-path
```

```
user@host# commit
```

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

## Managing VNFs on NFX Series Devices

### IN THIS SECTION

- [Managing VNF States | 105](#)
- [Managing VNF MAC Addresses | 106](#)
- [Managing the MTU of a VNF Interface | 107](#)
- [Accessing a VNF from the JCP | 107](#)
- [Viewing the List of VNFs | 108](#)



- [Displaying the Details of a VNF | 108](#)
- [Deleting a VNF | 109](#)

## 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
```

# 6

CHAPTER

## Configuring High Availability

---

Chassis Cluster on NFX150 Devices | **112**

Upgrading or Disabling a Chassis Cluster on NFX150 Devices | **125**

---

# Chassis Cluster on NFX150 Devices

## IN THIS SECTION

- [NFX150 Chassis Cluster Overview | 112](#)
- [Chassis Cluster Interfaces | 113](#)
- [Chassis Cluster Limitation | 114](#)
- [Example: Configuring a Chassis Cluster on NFX150 Devices | 114](#)

A chassis cluster, where two devices operate as a single device, provides high availability on NFX150 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.

## NFX150 Chassis Cluster Overview

You can configure NFX150 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 heth-0-0 port on one node to the heth-0-0 port on the second node.

**NOTE:** You can use only the heth-0-0 port 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 port except the heth-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 master 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 NFX150 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 heth-0-0 port is configured as the dedicated control interface on NFX150 devices. Therefore, you cannot map the heth-0-0 port to any other virtual interface if the device is part of a chassis cluster.

- **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 NFX150 Devices

### IN THIS SECTION

- [Requirements | 114](#)
- [Overview | 115](#)
- [Configuration | 115](#)
- [Verification | 121](#)

This example shows how to set up chassis clustering on NFX150 devices.

### Requirements

Before you begin:

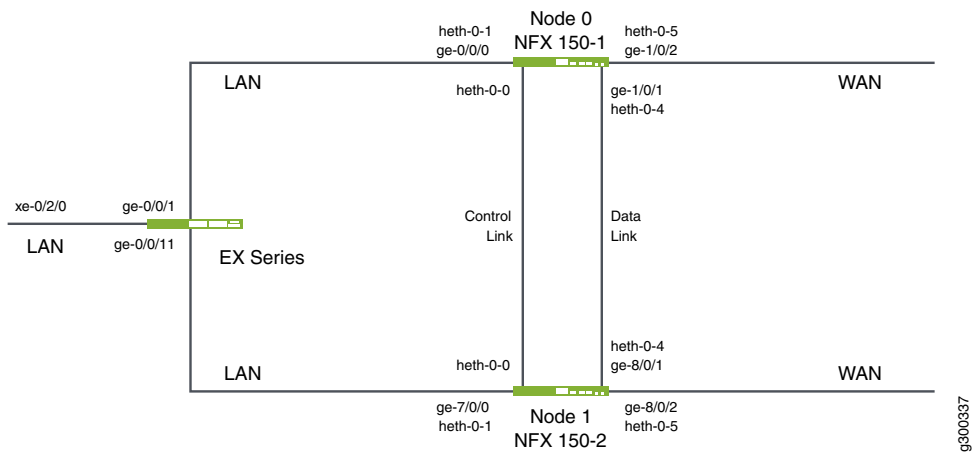
- Physically connect the two devices and ensure that they are the same NFX150 model.
- Ensure that both devices are running the same Junos OS version
- Remove all interface mapping for the control port heth-0-0 on both the nodes.
- Connect the dedicated control port heth-0-0 on node 0 to the heth-0-0 port on node 1.
- Connect the fabric port on node 0 to the fabric port on node 1.

Overview

Figure 10 on page 115 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: NFX150 Chassis Cluster



Configuration

IN THIS SECTION

- [Configuring a Chassis Cluster | 115](#)
- [Configuring Redundant Groups and Redundant Interfaces | 119](#)

*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@NFX150-1> **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 NFX150-1
user@host1# set groups node0 interfaces fxp0 unit 0 family inet address 10.204.41.80/20

user@host1# set groups node1 system host-name NFX150-2
user@host1# set groups node1 interfaces fxp0 unit 0 family inet address 10.204.41.90/20

```

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 groups node0 routing-options static route 0.0.0.0/0 next-hop 10.204.47.254
user@host1# set groups node1 routing-options static route 0.0.0.0/0 next-hop 10.204.47.254

```

4. Map the physical LAN port to the virtual LAN interface on FPC0:

```

user@host1# set groups node0 vmhost virtualization-options interfaces ge-0/0/0 mapping interface
heth-0-1

```

```
user@host1# set groups node1 vmhost virtualization-options interfaces ge-7/0/0 mapping interface heth-0-2
```

**NOTE:** In a chassis cluster, the FPC1 ports on the secondary node are denoted as ge-8/0/x, and the FPC0 ports are denoted as ge-7/0/x.

5. Map the physical WAN port to the virtual WAN interface on FPC1:

```
user@host1# set groups node0 vmhost virtualization-options interfaces ge-1/0/2 mapping interface heth-0-5
```

```
user@host1# set groups node1 vmhost virtualization-options interfaces ge-8/0/2 mapping interface heth-0-5
```

6. Configure port peering between the FPC0 and FPC1 on nodes 0 and 1. 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@host# set groups node0 chassis cluster redundant-interface ge-1/0/0 mapping-interface ge-0/0/0
```

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

7. Configure the fabric ports:

```
user@host1# set groups node0 vmhost virtualization-options interfaces ge-1/0/1 mapping interface heth-0-4
```

```
user@host1# set groups node1 vmhost virtualization-options interfaces ge-8/0/1 mapping interface heth-0-4
```

```
user@host1# set interfaces fab0 fabric-options member-interfaces ge-1/0/1
```

```
user@host1# set interfaces fab1 fabric-options member-interfaces ge-8/0/1
```

8. Apply the node-specific configurations on nodes 0 and 1:

```
user@host1# set apply-groups "${node}"
```

9. 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
```

10. Verify the interfaces:

```
user@host1# run show chassis cluster interfaces
```

### ***Configuring Redundant Groups and Redundant Interfaces***

A redundancy group (RG) defines failover properties for a group of devices. When a failure occurs, a RG enables the control plane to fail over to the secondary node. RETH is the redundant interface consisting of two Ethernet links, one from each device in the cluster to form a single logical interface. All logical configurations are tied to the reth interface through the RGs.

### **Step-by-Step Procedure**

1. Configure redundancy groups 0 and 1. RG 0 controls the control plane and it determines the primary node. RG 1 controls the data plane and includes 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 RG0. 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 redundancy-group 1 node 0 priority 100
```

```
user@host1# set chassis cluster redundancy-group 1 node 1 priority 100
```

```
user@host1# set chassis cluster redundancy-group 2 node 0 priority 100
```

```
user@host1# set chassis cluster redundancy-group 2 node 1 priority 100
```

2. Enable preempt for RG1.

```
user@host1# set chassis cluster redundancy-group 1 preempt
```

**NOTE:** If preempt is added to a redundancy group configuration, the device with the higher priority in the group can initiate a failover to become master. By default, preemption is disabled.

3. Configure the interfaces that the redundancy groups need to monitor to determine whether an interface is up or down.

By default, redundancy groups have a threshold tolerance value of 255. When an interface monitored by a redundancy group becomes unavailable, its weight is subtracted from the redundancy group's threshold. When a redundancy group's threshold reaches 0, it fails over to the other node.

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

4. Configure the data interfaces so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly.

Define the following parameters:

- The maximum number of reth interfaces for the cluster, so that the system can allocate the appropriate resources for them.

```
user@host1# set chassis cluster reth-count 5
```

- The heartbeat interval and threshold, which define the wait time before failover is triggered in the chassis cluster.

```
user@host1# set chassis cluster heartbeat-interval 1000
user@host1# set chassis cluster heartbeat-threshold 5
```

- Membership information of the member interfaces to reth interfaces.

```
user@host# set interfaces ge-1/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-1/0/2 gigether-options redundant-parent reth2
user@host# set interfaces ge-8/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-8/0/2 gigether-options redundant-parent reth2
```

5. Configure the reth interfaces:

- Configure reth1:

```
user@host1# set interfaces reth1 vlan-tagging
user@host1# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host1# set interfaces reth1 unit 0 vlan-id 100
user@host1# set interfaces reth1 unit 0 family inet address 192.0.3.1/24
user@host1# set interfaces reth1 unit 0 family inet6 address 2001:db8:1:1::/64
```

- Configure reth2:



```

user@host1# set interfaces reth2 vlan-tagging
user@host1# set interfaces reth2 redundant-ether-options redundancy-group 2
user@host1# set interfaces reth2 unit 0 vlan-id 200
user@host1# set interfaces reth2 unit 0 family inet address • 203.0.113.1/24
user@host1# set interfaces reth2 unit 0 family inet6 address 2001:db8:2:1::/64

```

6. 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

### 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@NFX150-1> 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

```

Redundancy group: 1 , Failover count: 0
node0 100      primary      yes      no      None
node1 100      secondary    yes      no      None

Redundancy group: 2 , Failover count: 0
node0 100      primary      no       no      None
node1 100      secondary    no       no      None

```

- Verify the status of the redundancy groups:

```
root@NFX150-1> show chassis cluster information
```

```

node0:
-----
Redundancy Group Information:

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

        Time           From           To           Reason
        Mar 21 12:06:18 hold           secondary    Hold timer
expired
        Mar 21 12:18:46 secondary      primary      Remote node
reboot

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

        Time           From           To           Reason
        Mar 21 12:06:19 hold           secondary    Hold timer
expired
        Mar 21 12:08:51 secondary      primary      Remote is in
secondary hold

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

        Time           From           To           Reason
        Mar 21 12:06:19 hold           secondary    Hold timer
expired
        Mar 21 12:18:46 secondary      primary      Remote node
reboot

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 21 13:02:05 hold           secondary    Hold timer
expired

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

        Time           From           To           Reason
        Mar 21 13:02:05 hold           secondary    Hold timer
expired

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

        Time           From           To           Reason
        Mar 21 13:02:06 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@NFX150-1> 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
           ge-1/0/1   Up / Up                  Disabled
fab0
fab0

```

```
fab1    ge-8/0/1      Up    /  Up      Disabled
fab1
```

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Down	Not configured
reth1	Down	1
reth2	Down	2
reth3	Down	Not configured
reth4	Down	Not configured

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

Interface Monitoring:

Interface	Weight	Status (Physical/Monitored)	Redundancy-group
ge-8/0/0	255	Up / Up	1
ge-1/0/0	255	Up / Up	1
ge-8/0/2	255	Up / Up	2
ge-1/0/2	255	Up / Up	2

- Verify the status of the port-peering interfaces:

```
root@NFX150-1> show chassis cluster port-peering
```

```
node0:
```

```
-----
```

Port peering interfaces:

Backend L3		Mapped Peer L2	
Interface	Status	Interface	Status
ge-1/0/0	Up	ge-0/0/0	Up

```
node1:
```

```
-----
```

Port peering interfaces:

Backend L3		Mapped Peer L2	
Interface	Status	Interface	Status
ge-8/0/0	Up	ge-7/0/0	Up

## RELATED DOCUMENTATION

[Monitoring of Global-Level Objects in a Chassis Cluster](#)[Monitoring Chassis Cluster Interfaces](#)[Monitoring IP Addresses on a Chassis Cluster](#)[Configuring Cluster Failover Parameters](#)[Chassis Cluster Redundancy Group Failover](#)

## Upgrading or Disabling a Chassis Cluster on NFX150 Devices

### IN THIS SECTION

- [Upgrading Individual Devices in a Chassis Cluster Separately | 125](#)
- [Disabling a Chassis Cluster | 126](#)

### 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
```

# 7

CHAPTER

## Configuring Service Chaining

---

Service Chaining on NFX150 Devices | **128**

Example: Configuring Service Chaining Using VLANs on NFX150 Network Services Platform | **130**

Example: Configuring Service Chaining Using SR-IOV on NFX150 Network Services Platform | **137**

Example: Configuring Service Chaining Using a Custom Bridge | **144**

Example: Configuring Service Chaining for LAN-WAN Routing | **151**

Example: Configuring Cross Connect on NFX150 Devices | **156**

Example: Configuring Service Chaining for LAN Routing | **166**

Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices | **169**

---

# Service Chaining on NFX150 Devices

## IN THIS SECTION

- [Understanding Service Chaining | 128](#)
- [Configuring Service Chaining Using VLANs | 128](#)
- [Configuring Service Chaining Using DHCP Services on VLANs | 129](#)

## Understanding Service Chaining

In many network environments, it is common for traffic to flow through several *network services* on the way to its destination. These services—firewalls, Network Address Translators (NAT), load balancers, and so on—are generally spread across multiple network elements. Each device is a separate piece of hardware, providing a different service, and requiring separate operation and management. This method of linking together multiple network functions could be thought of as *physical service chaining*.

A more efficient model for service chaining is to virtualize and consolidate network functions onto a single device.

Virtualized service chaining is supported on NFX150 devices starting with Junos OS Release 18.1. Virtual network functions (VNFs) can be installed and linked together to provide L2, L3, and L4-L7 services for traffic flowing through the device.

## Configuring Service Chaining Using VLANs

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

To configure service chaining:

1. Create a VLAN. Use one of the following commands:

- Create a VLAN without a VLAN ID. You can add only access ports to this VLAN:

```
set vmhost vlans vlan-name vlan-id none
```



- Create a VLAN with a VLAN ID:

```
set vmhost vlans vlan-name vlan-id vlan-id
```

- Create a VLAN using a list of VLAN IDs:

```
set vmhost vlans vlan-name vlan-id-list vlan-id range | comma-separated list
```

2. Attach an interface on the VNF to the VLAN:

```
set virtual-network-functions vnf-name interfaces ethx mapping vlan mode [access|trunk]
```

```
set virtual-network-functions vnf-name interfaces ethx mapping vlan members list
```

3. Attach a native VLAN ID to the VNF interface:

```
set virtual-network-functions vnf-name interfaces ethx mapping vlan native-vlan-id vlan-id
```

## Configuring Service Chaining Using DHCP Services on VLANs

Using DHCP services, you need not manually configure the IP addresses on the VNF interfaces to achieve service-chaining. Enable DHCP clients on the glue bridge interfaces within the VNF for an IP address to be assigned from the DHCP pool.

To configure service chaining:

1. Create a VLAN with a VLAN ID **none**.

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

**NOTE:** To use the DHCP pooling feature, the VLAN ID must be set to **none**.

2. Specify the IP address pool to be used:

```
user@host# set access address-assignment pool p4 family inet network network-address
```

```
user@host# set access address-assignment pool p4 family inet range r4 low start-IP-address
```

```
user@host# set access address-assignment pool p4 family inet range r4 high end-IP-address
```

3. Attach an interface from FPC1 to the VLAN:

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

4. Configure an IP address on the interface and enable dhcp-server on it.

```
user@host# set interface ge-1/0/x unit 0 family inet address address/prefix-length
user@host# set system services dhcp-local-server group grp1 interface ge-1/0/x
```

5. Attach an interface on the VNF to the VLAN to complete the service chain:

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

6. Enable the DHCP client on the VNF.

To check the assigned IP address, use the `show system visibility vnf` command.

## Example: Configuring Service Chaining Using VLANs on NFX150 Network Services Platform

### IN THIS SECTION

- [Requirements | 131](#)
- [Overview | 131](#)
- [Configuration | 132](#)

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

## Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

Before you configure service chaining, be sure you have:

- Installed and launched the relevant VNFs, assigned the corresponding interfaces, and configured the resources.

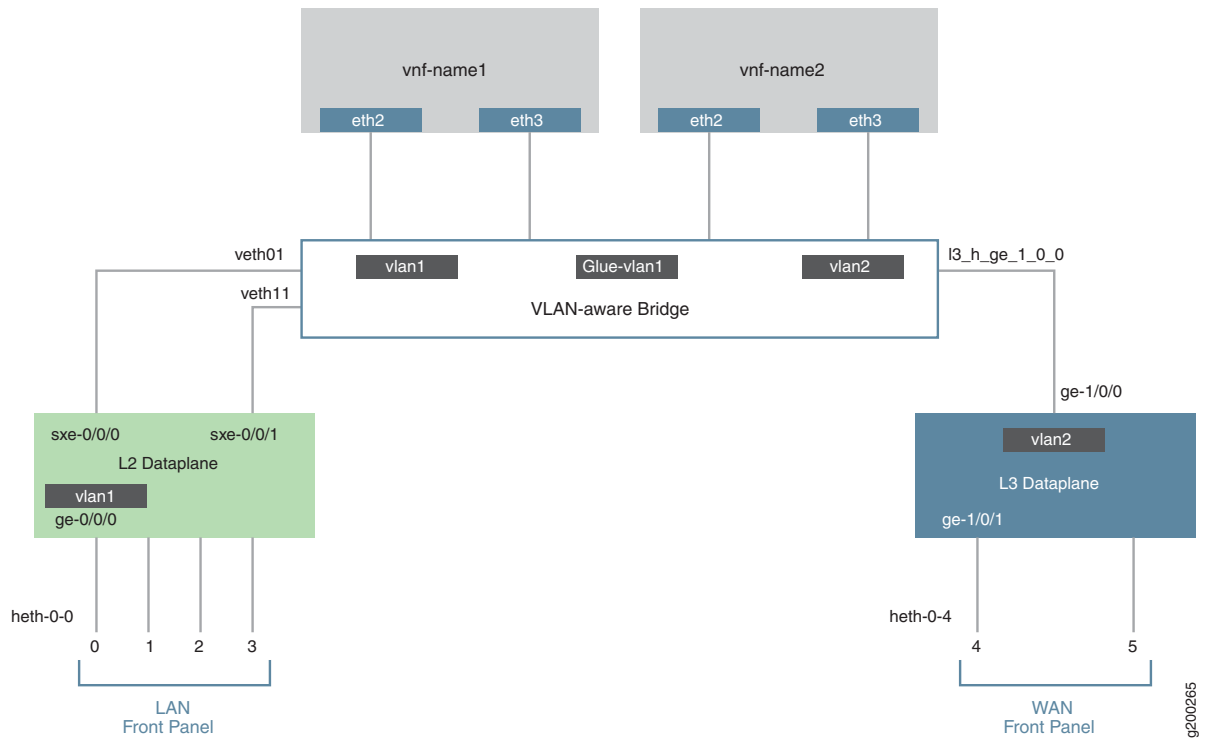
## Overview

Service chaining on a device running the disaggregated Junos OS allows multiple services, or virtual network functions (VNFs), to be applied to traffic as it 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 device running the disaggregated Junos OS, as shown in [Figure 11 on page 132](#).

Figure 11: Service Chaining Using VLANs



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

- The front panel ports.
- The internal-facing ports.
- The VM interfaces. VNF interfaces must use the format eth#, where # is from 0 through 9.
- VLANs, to provide bridging between the sxe and VM interfaces.

## Configuration

### IN THIS SECTION

- [Configuring the Interfaces | 133](#)
- [Configuring the VNF Interfaces and Creating the Service Chain | 136](#)

## Configuring the Interfaces

### Step-by-Step Procedure

To configure the interfaces:

1. Connect to the JCP.

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

2. Map the physical (heth) interfaces to the virtual (ge) interfaces.

```
user@jcp# set vmhost virtualization-options interfaces ge-0/0/2 mapping interface heth-0-2
user@jcp# set vmhost virtualization-options interfaces ge-1/0/1 mapping interface heth-0-4
```

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

```
user@jcp# 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 could be a trunk port if appropriate.

```
user@jcp# set interfaces ge-0/0/2.0 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@jcp# set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
user@jcp# set interfaces sxe-0/0/0.0 family ethernet-switching vlan members vlan1
```

6. Configure the WAN side front panel port with vlan-tagging and an IP address:

```
user@jcp# set interfaces ge-1/0/1 vlan-tagging
```

```
user@jcp# set interfaces ge-1/0/1.0 vlan-id 1178
user@jcp# set interfaces ge-1/0/1.0 family inet address 192.0.2.1/24
```

7. Configure the WAN side internal-facing interface as a vlan-tagged interface and assign an IP address to it:

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

8. Commit the configuration.

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

## Results

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

[edit]

```
user@host# show interfaces ge-0/0/2
```

```
mtu 9192;
unit 0 {
    family ethernet-switching {
        vlan {
            members [ default 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 vlans**

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

## Configuring the VNF Interfaces and Creating the Service Chain

### Step-by-Step Procedure

Once you have completed the configuration on JCP, you need to:

1. Configure the vmhost instance with either with LAN, WAN, or glue-vlan to be used for service chaining

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

2. Bring up the VNF (vnf-name1) with one virtio interface mapped to VLAN, and another interface mapped to glue-vlan.

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

3. Similarly bring up the second VNF (vnf-name2) with one interface mapped to VLAN2, and the second interface mapped to the same glue-vlan.

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

4. Finally, configure the IP addresses and static routes for each interface of the VNFs as shown in [Figure 11 on page 132](#).

### RELATED DOCUMENTATION

*Understanding Service Chaining on Disaggregated Junos OS Platforms*

*Disaggregated Junos OS VMs*

*Virtio and SR-IOV Usage*



# Example: Configuring Service Chaining Using SR-IOV on NFX150 Network Services Platform

## IN THIS SECTION

- [Requirements | 137](#)
- [Overview | 137](#)
- [Configuration | 139](#)

This example shows how to configure service chaining using SR-IOV on NFX150 platforms.

## Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

Before you configure service chaining, be sure you have:

- Installed and launched the relevant VNFs

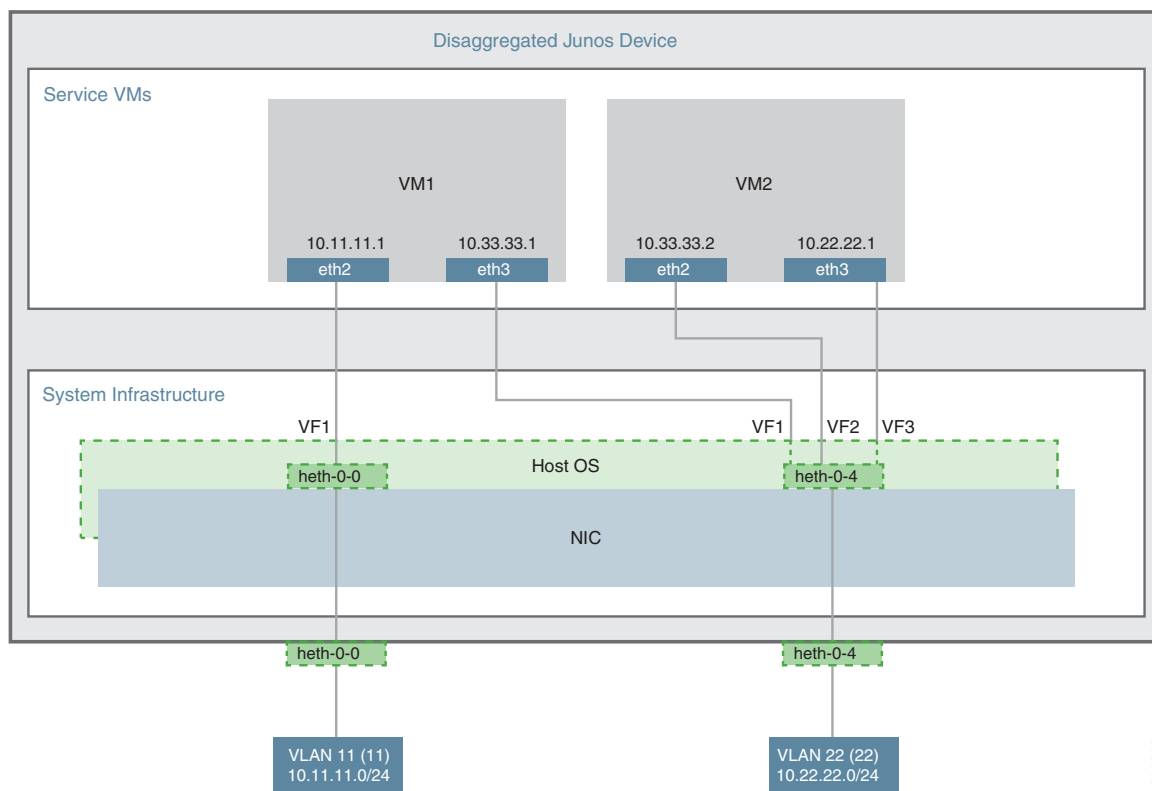
## Overview

Service chaining on a device running the disaggregated Junos OS allows multiple services, or virtual network functions (VNFs), to be applied to traffic as it 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 device running the disaggregated Junos OS, as shown in [Figure 12 on page 138](#).

Figure 12: Service Chaining Using SR-IOV—Device Infrastructure



This example uses the front panel ports heth-0-0 and heth-0-4. The VMs use two interfaces each, eth2 and eth3.

These elements are generally separated into two parts: a *LAN side* and a *WAN side*.

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

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

- The front panel ports, heth-0-0 and heth-0-4.
- The internal-facing ports, sxe-0/0/0 and sxe-0/0/1.
- The VNF interfaces. VNF interfaces must use the format eth#, where # is from 0 through 9.
- The virtual function setting, to indicate SR-IOV is being used to provide direct access between the sxe and VNF interfaces.

## Configuration

### IN THIS SECTION

- [Configuring the Packet Forwarding Engine Interfaces | 139](#)
- [Creating the Service Chain | 142](#)

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@jcp#
```

```
set vlans Vlan11 vlan-id 11
```

```
set interfaces ge-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 ge-1/0/1 family ethernet-switching interface-mode trunk
```

```
set interfaces ge-1/0/1 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. Connect to the JCP.

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

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

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

3. Configure the Packet Forwarding Engine's LAN-side front panel port, and add it to the LAN-side VLAN.  
The LAN-side port is typically an access port, but could be a trunk port if appropriate.

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

4. Configure the Packet Forwarding Engine's 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@jcp# set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
user@jcp# set interfaces sxe-0/0/0.0 family ethernet-switching vlan member Vlan11
```

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

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

6. Configure the Packet Forwarding Engine's 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@jcp# set interfaces ge-1/0/1 family ethernet-switching interface-mode trunk
user@jcp# set interfaces ge-1/0/1 family ethernet-switching vlan members Vlan22
```

7. Configure the Packet Forwarding Engine's WAN-side internal-facing interface as a trunk port, and add it 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@jcp# set interfaces sxe-0/0/1.0 family ethernet-switching interface-mode trunk
user@jcp# set interfaces sxe-0/0/1.0 family ethernet-switching vlan members Vlan22
```

8. Commit the configuration.

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

## Results

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

```
[edit]
user@jcp# show interfaces ge-0/0/0
unit 0 {
  family ethernet-switching {
    vlan {
      members Vlan11;
    }
  }
}

[edit]
user@jcp# show interfaces ge-1/0/1
unit 0 {
```

```

family ethernet-switching {
    interface-mode trunk;
    vlan {
        members Vlan22;
    }
}

[edit]
user@jcp# show interfaces sxe-0/0/0
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members Vlan11;
        }
    }
}

[edit]
user@jcp# show interfaces sxe-0/0/1
unit 0 {
    family ethernet-switching {
        interface-mode trunk;
        vlan {
            members Vlan22;
        }
    }
}

[edit]
user@jcp# show vlans
Vlan11 {
    vlan-id 11;
}
Vlan22 {
    vlan-id 22;
}

```

## Creating the Service Chain

### Step-by-Step Procedure

To configure the VNF interfaces and create the service chain:

1. Configure *vnf-name1*'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. VNF must use the interfaces from eth0 through eth9.

The heth interface is the configurable representation of the related NIC interface.

```
user@jcp> configure
[edit]
user@jcp# set virtual-network-functions vnf-name1 interfaces eth2 mapping heth-0-0
virtual-function
```

2. Configure the *vnf-name1*'s WAN-side interface from the eth3 VNF interface as shown in [Figure 12 on page 138](#).

```
user@jcp# set virtual-network-functions vnf-name1 interfaces eth3 mapping heth-0-4
virtual-function
```

3. Similarly bring up *vnf-name2* with both interfaces eth2 and eth3 on heth-0-4.

```
user@jcp# set virtual-network-functions vnf-name2 interfaces eth2 mapping heth-0-4
virtual-function
user@jcp# set virtual-network-functions vnf-name2 interfaces eth3 mapping heth-0-4
virtual-function
```

4. Finally, configure the IP addresses and static routes for each interface of the VNFs, and add routes to achieve the 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

## IN THIS SECTION

- [Requirements | 144](#)
- [Overview | 144](#)
- [Configuration | 145](#)
- [Verifying the Configuration | 148](#)

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

## Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

## Overview

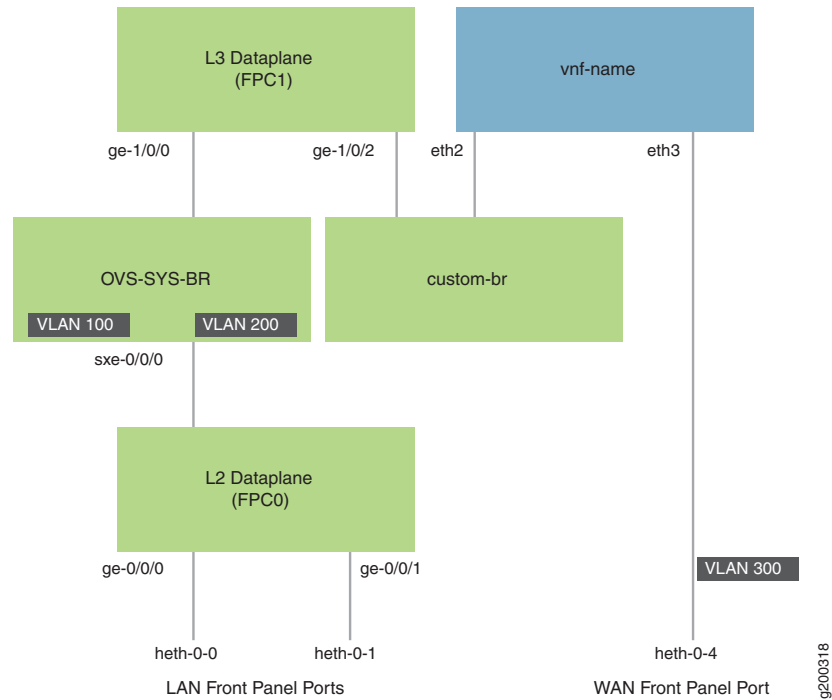
The default system bridge is OVS. The OVS bridge is a VLAN-aware system bridge, which acts as the NFV backplane to which the VNFs and FPCs connect. However, you can choose to create a custom bridge based on your requirement. This example explains how to configure service chaining using a custom bridge.

## Topology

This example uses the topology shown in [Figure 13 on page 145](#).



Figure 13: Service Chaining Using a Custom Bridge



## Configuration

### IN THIS SECTION

- [Create VLANs and the Custom Bridge | 145](#)
- [Map the Interfaces | 146](#)
- [Configure the Layer 2 Datapath | 146](#)
- [Configure the Layer 3 Datapath | 147](#)
- [Configure the VNF | 147](#)

### Create VLANs and 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
```

## Map the Interfaces

### Step-by-Step Procedure

1. Map the heth-0-0 physical port to the FPC0 interface.

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

2. Map the heth-0-1 physical port to the FPC0 interface.

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

3. Map the FPC1 interface ge-1/0/1 to the system bridge OVS.

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

4. Map the FPC1 interface ge-1/0/2 to the custom bridge.

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

## 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 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
```

## Configure the Layer 3 Datapath

### Step-by-Step Procedure

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

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

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

## Configure the VNF

### Step-by-Step Procedure

1. Launch the VNF:

```
user@host# set virtual-network-functions vnf-name image /var/public/centos-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. Create a VNF interface on the custom OVS bridge:

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

5. Attach a VNF interface to a physical interface by using the SR-IOV virtual function:

```
user@host# set virtual-network-functions vnf-name interfaces eth3 mapping interface heth-0-4
virtual-function vlan-id 300
```

6. Specify the memory allocation for the VNF:

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

## Verifying the Configuration

### IN THIS SECTION

- [Verify the Control Plane Configuration | 148](#)
- [Verify the Data Plane Configuration | 149](#)

### Verify the Control Plane Configuration

#### Purpose

Verify the control plane configuration:

#### Action

To verify the control plane configuration:

- 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	custom-br		
vmhost	vlan200	200	ge-1/0/2.0

- Verify that the VNF is operational. View the status of the VNF to ensure that the VNF is up and running.

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

ID	Name	State	Liveliness
1	vjunos0	Running	alive
2	centos1	Running	alive
3	centos2	Running	alive
11057	LTE	Running	alive

The **Liveliness** output field of the VNF indicates whether the IP address of the VNF is reachable or not reachable from JCP.

To view more details of the VNF:

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

## Verify the Data Plane Configuration

### Purpose

Verify the data plane configuration.

### Action

To verify the data plane configuration:

- Verify the status of the physical ports.

```
user@host> show interfaces heth-0-0 statistics
```

Physical interface: heth-0-0, 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:00:53:8d, Hardware address: 00:00:5e:00:53:8d
Input packets : 272469
Output packets: 674
MAC statistics:
  Input bytes: 17438016, Input packets: 272469, Output bytes: 48658, Output
packets: 674
VF statistics:
  VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
    Input bytes: 17433984, Input packets: 272406, Output bytes: 48658, Output
packets: 674, Multicast packets: 272406
  VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
  VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
  VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0

```

- 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 (1d 22:28 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 : 125892
Output packets: 22
Protocol eth-switch, MTU: 9192
  Flags: Is-Primary

```

## Example: Configuring Service Chaining for LAN-WAN Routing

### IN THIS SECTION

- [Requirements | 152](#)
- [Overview | 152](#)
- [Configuration | 153](#)
- [Verification | 154](#)

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

## Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

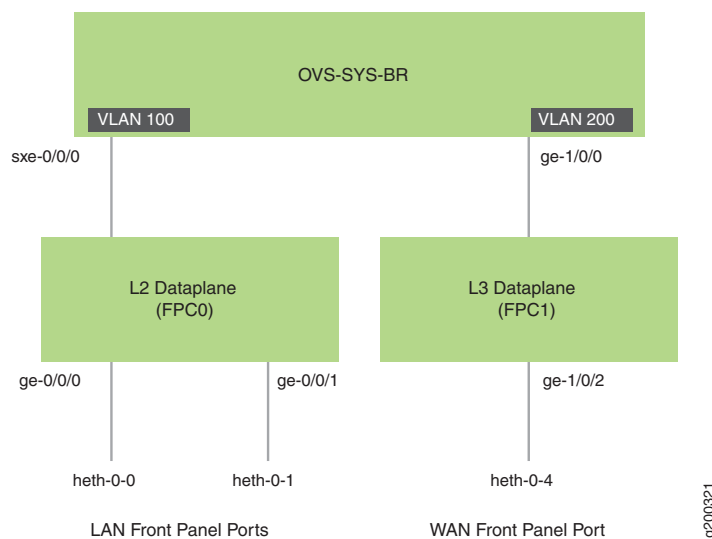
## Overview

This example explains how to configure the various layers of the device to enable traffic from the LAN network to enter the device, flow through the OVS, exit the device, and enter the WAN network.

### Topology

This example uses the topology shown in [Figure 13 on page 145](#).

**Figure 14: Service Chaining Using a Custom Bridge**





## Configuration

### IN THIS SECTION

- [Map the Interfaces | 153](#)
- [Configure the Layer 2 Datapath | 153](#)
- [Configure the Layer 3 Datapath | 154](#)

### Map the Interfaces

#### Step-by-Step Procedure

1. Map the heth-0-0 physical port to the FPC0 interface.

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

2. Map the heth-0-1 physical port to the FPC0 interface.

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

3. Map the FPC1 interface ge-1/0/2 to the physical port heth-0-4.

```
user@host# set vmhost virtualization-options interfaces ge-1/0/2 mapping interface heth-0-4
```

### Configure 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@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
```

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
```

## Configure the Layer 3 Datapath

### Step-by-Step Procedure

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

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

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

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

## Verification

### IN THIS SECTION

- [Verifying the Status of the Interfaces | 154](#)

### Verifying the Status of the Interfaces

#### Purpose

Verify the status of the Layer 2 and Layer 3 interfaces.

#### Action

To verify the status of the interfaces:

- Verify the status of the physical ports.

```
user@host> show interfaces heth-0-0 statistics
```

```
Physical interface: heth-0-0, 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:00:53:8d, Hardware address: 00:00:5e:00:53:8d
    Input packets : 272469
    Output packets: 674
  MAC statistics:
    Input bytes: 17438016, Input packets: 272469, Output bytes: 48658, Output
  packets: 674
  VF statistics:
    VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
      Input bytes: 17433984, Input packets: 272406, Output bytes: 48658, Output
packets: 674, Multicast packets: 272406
    VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
```

- 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 Cross Connect on NFX150 Devices

### IN THIS SECTION

- [Requirements | 157](#)
- [Overview | 157](#)
- [Configuration | 158](#)
- [Verifying the Configuration | 161](#)

This example shows how to configure cross-connect on NFX150 devices.

## Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

## Overview

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

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

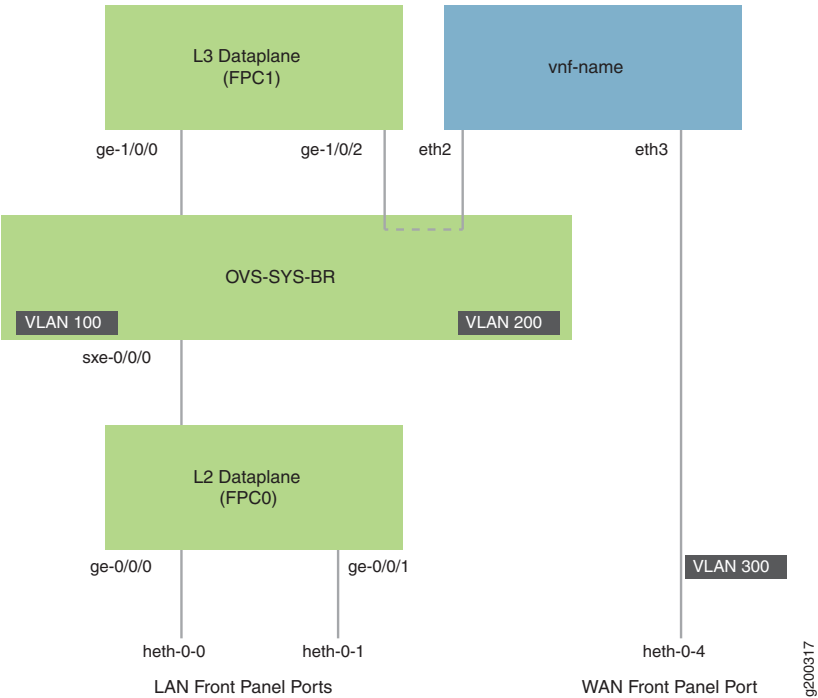
The **Cross-connect** feature supports the following:

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

## Topology

This example uses the topology shown in [Figure 13 on page 145](#).

Figure 15: Configuring Cross-Connect



## Configuration

### IN THIS SECTION

- Create VLANs | 158
- Map the Interfaces | 159
- Configure the Layer 2 Datapath | 159
- Configure the Layer 3 Datapath | 160
- Configure the VNF | 160
- Configure Cross-Connect | 161

### Create VLANs

#### 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 a VLAN for the WAN-side interface.

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

## Map the Interfaces

### Step-by-Step Procedure

1. Map the heth-0-0 physical port to the FPC0 interface.

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

2. Map the heth-0-1 physical port to the FPC0 interface.

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

3. Map the FPC1 interface ge-1/0/2 to the system bridge OVS.

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

## 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
```

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
```

## Configure the Layer 3 Datapath

### Step-by-Step Procedure

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

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

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

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

## Configure the VNF

### Step-by-Step Procedure

1. Launch the VNF:

```
user@host# set virtual-network-functions vnf-name image /var/public/centos-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. Create a host VLAN:

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

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
```

6. Attach a VNF interface to a physical interface by using the SR-IOV virtual function:



```
user@host# set virtual-network-functions vnf-name interfaces eth3 mapping interface heth-0-4
virtual-function vlan-id 300
```

7. Specify the memory allocation for the VNF:

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

## Configure Cross-Connect

### Step-by-Step Procedure

1. Configure cross-connect:

```
user@host# set vmhost cross-connect c1 virtual-interface ge-1/0/2
```

```
user@host# set vmhost cross-connect c1 virtual-network-function vnf-name interface eth2
```

## Verifying the Configuration

### IN THIS SECTION

- [Verify the Control Plane Configuration | 161](#)
- [Verify the Data Plane Configuration | 162](#)

### Verify the Control Plane Configuration

#### Purpose

Verify the control plane configuration:

#### Action

To verify the control plane configuration:

- 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	custom-br		ge-1/0/2.0
vmhost	vlan200	200	

- Verify that the VNF is operational. View the status of the VNF to ensure that the VNF is up and running.

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

ID	Name	State	Liveliness
2	VNF	Running	alive

The **Liveliness** output field of the VNF indicates whether the IP address of the VNF is reachable or not reachable from JCP.

To view more details of the VNF:

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

Virtual Network Function Information	
-----	
Id:	2
Name:	VNF
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

## Verify the Data Plane Configuration

### Purpose

Verify the data plane configuration.

### Action

To verify the data plane configuration:

- Verify the status of the physical ports.

```
user@host> show interfaces heth-0-0 statistics
```

```
Physical interface: heth-0-0, 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:00:53:8d, Hardware address: 00:00:5e:00:53:8d
    Input packets : 311143
    Output packets: 674
  MAC statistics:
    Input bytes: 19913152, Input packets: 311143, Output bytes: 48658, Output
  packets: 674
  VF statistics:
    VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
      Input bytes: 19909120, Input packets: 311080, Output bytes: 48658, Output
packets: 674, Multicast packets: 311080
    VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
```

- 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 (6d 00:28 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 : 311115
Output packets: 22
  Protocol eth-switch, MTU: 9192
  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: 158, SNMP ifIndex: 536
  Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Link-mode: Full-duplex,
  Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags      : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  CoS queues        : 8 supported, 8 maximum usable queues
  Current address: 00:00:5e:00:53:5d, Hardware address: 00:00:5e:00:53:5d
  Last flapped      : 2018-04-23 06:03:29 UTC (1d 00:04 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-1/0/2.0 (Index 342) (SNMP ifIndex 538)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.200 ] Encapsulation: ENET2
  Input packets : 0
  Output packets: 0
  Security: Zone: untrust
  Allowed host-inbound traffic : dns dhcp tftp https
  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
  Protocol multiservice, MTU: Unlimited

Logical interface ge-1/0/2.32767 (Index 343) (SNMP ifIndex 545)
  Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2
  Input packets : 0
  Output packets: 0
  Security: Zone: Null
  Protocol multiservice, MTU: Unlimited
  Flags: None

```

## RELATED DOCUMENTATION

[Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices](#) | 169

# Example: Configuring Service Chaining for LAN Routing

## IN THIS SECTION

- [Requirements | 166](#)
- [Overview | 166](#)
- [Configuration | 167](#)

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

## Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

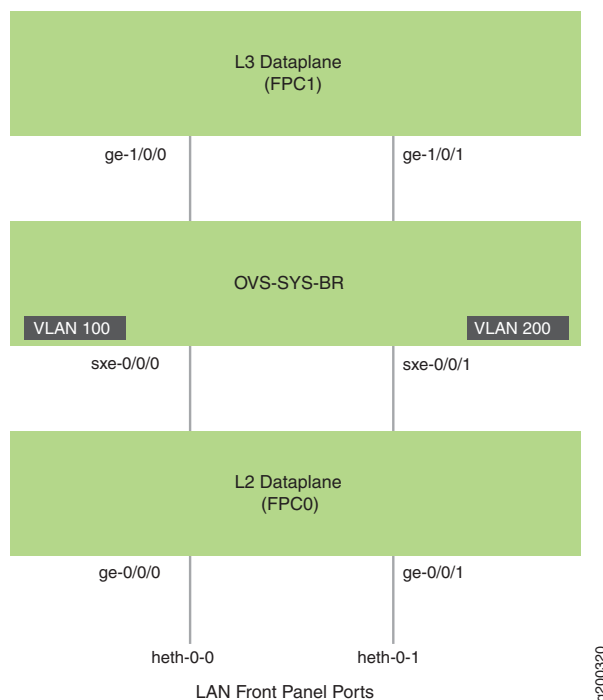
## Overview

This example explains how to configure the various layers of the device to enable traffic flow within a LAN network.

## Topology

This example uses the topology shown in [Figure 13 on page 145](#).

Figure 16: Service Chaining for LAN Routing



## Configuration

### IN THIS SECTION

- [Map the Interfaces | 167](#)
- [Configure the Layer 2 Datapath | 168](#)
- [Configure the Layer 3 Datapath | 168](#)

### Map the Interfaces

#### Step-by-Step Procedure

1. Map the heth-0-0 physical port to the FPC0 interface.

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

2. Map the heth-0-1 physical port to the FPC0 interface.

```
user@jcp# set vmhost virtualization-options interfaces ge-0/0/2 mapping interface heth-0-1
```

## Configure the Layer 2 Datapath

### Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

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

```
user@jcp# set vlans vlan200 vlan-id 200
```

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

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

```
user@jcp# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan100
```

```
user@jcp# set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode trunk
```

```
user@jcp# set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100
```

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@jcp# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
```

```
user@jcp# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
```

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

```
user@jcp# set interfaces sxe-0/0/1 unit 0 family ethernet-switching vlan members vlan200
```

## Configure the Layer 3 Datapath

### Step-by-Step Procedure

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

```
user@jcp# set interfaces ge-1/0/1 vlan-tagging
```

```
user@jcp# set interfaces ge-1/0/1 unit 0 vlan-id 100
```

```
user@jcp# set interfaces ge-1/0/1 unit 0 family inet address 192.0.2.1/24
```

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

```
user@jcp# set interfaces ge-1/0/2 vlan-tagging
```

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

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



## RELATED DOCUMENTATION

| [Example: Configuring Service Chaining for LAN-WAN Routing | 151](#)

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

### IN THIS SECTION

- [Requirements | 169](#)
- [Overview | 169](#)
- [Configuration | 171](#)
- [Verifying the Configuration | 173](#)

This example shows how to configure cross-connect using a custom bridge on NFX150 devices.

### Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

### Overview

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

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

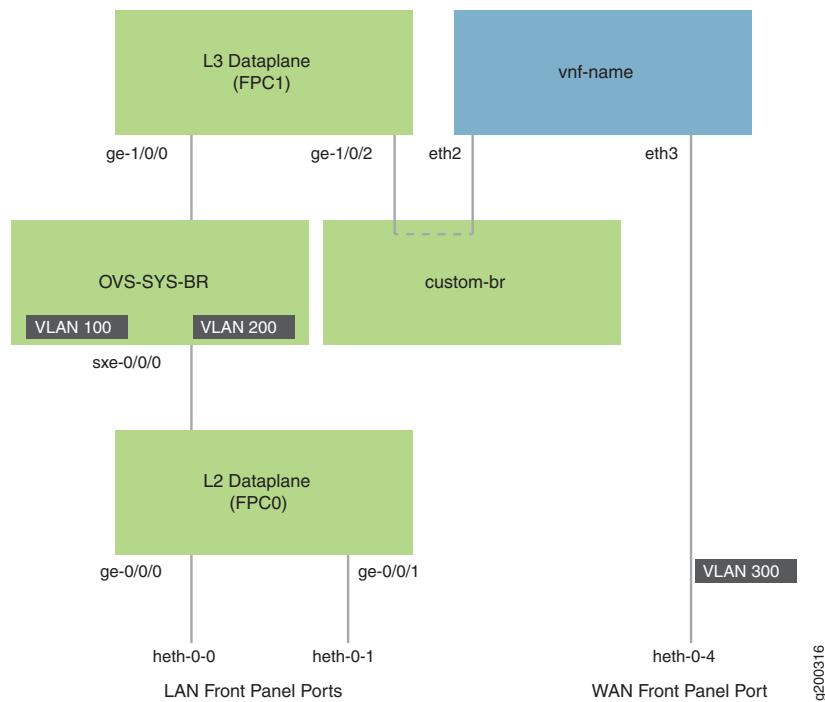
The **Cross-connect** feature supports the following:

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

## Topology

This example uses the topology shown in [Figure 13 on page 145](#).

**Figure 17: Configuring Cross-Connect**



## Configuration

### IN THIS SECTION

- [Create VLANs | 171](#)
- [Map the Interfaces | 171](#)
- [Configure the Layer 2 Datapath | 172](#)
- [Configure the Layer 3 Datapath | 172](#)
- [Configure the VNF | 172](#)
- [Configure Cross-Connect | 173](#)

### Create VLANs

#### Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

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

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

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

### Map the Interfaces

#### Step-by-Step Procedure

1. Map the heth-0-0 physical port to the FPC0 interface.

```
user@jcp# set vmhost virtualization-options interfaces ge-0/0/0 mapping interface heth-0-0
```

2. Map the heth-0-1 physical port to the FPC0 interface.

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

3. Map the FPC1 interface ge-1/0/2 to the custom bridge.

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

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

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

## Configure the Layer 3 Datapath

### Step-by-Step Procedure

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

```
user@jcp# set interfaces ge-1/0/0 vlan-tagging
user@jcp# set interfaces ge-1/0/0 unit 0 vlan-id 100
user@jcp# 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@jcp# set interfaces ge-1/0/2 vlan-tagging
user@jcp# set interfaces ge-1/0/2 unit 0 vlan-id 200
user@jcp# set interfaces ge-1/0/2 unit 0 family inet address 203.0.113.2/24
```

## Configure the VNF

### Step-by-Step Procedure

1. Launch the VNF:

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

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

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

3. Pin a virtual CPU to a physical CPU:

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

4. Create a host VLAN:

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

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

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

6. Attach a VNF interface to a physical interface by using the SR-IOV virtual function:

```
user@jcp# set virtual-network-functions vnf-name interfaces eth3 mapping interface heth-0-4
virtual-function vlan-id 300
```

7. Specify the memory allocation for the VNF:

```
user@jcp# set virtual-network-functions memory size 1048576
```

## Configure Cross-Connect

### Step-by-Step Procedure

1. Configure cross-connect:

```
user@jcp# set vmhost cross-connect c1 virtual-interface ge-1/0/2
```

```
user@jcp# set vmhost cross-connect c1 virtual-network-function vnf-name interface eth2
```

## Verifying the Configuration

### IN THIS SECTION

- [Verify the Control Plane Configuration | 174](#)
- [Verify the Data Plane Configuration | 175](#)

## Verify the Control Plane Configuration

### Purpose

Verify the control plane configuration:

### Action

To verify the control plane configuration:

- 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	custom-br		ge-1/0/2.0
vmhost	vlan200	200	

- Verify that the VNF is operational. View the status of the VNF to ensure that the VNF is up and running.

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

ID	Name	State	Liveliness
2	VNF	Running	alive

The **Liveliness** output field of the VNF indicates whether the IP address of the VNF is reachable or not reachable from JCP.

To view more details of the VNF:

```
user@host# show virtual-network-functions VNF detail
```

Virtual Network Function Information	
-----	
Id:	2
Name:	VNF
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
```

## Verify the Data Plane Configuration

### Purpose

Verify the data plane configuration.

### Action

To verify the data plane configuration:

- Verify the status of the physical ports.

```
user@host> show interfaces heth-0-0 statistics
```

```
Physical interface: heth-0-0, 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:00:53:8d, Hardware address: 00:00:5e:00:53:8d
    Input packets : 311143
    Output packets: 674
  MAC statistics:
    Input bytes: 19913152, Input packets: 311143, Output bytes: 48658, Output
  packets: 674
  VF statistics:
    VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
    Input bytes: 19909120, Input packets: 311080, Output bytes: 48658, Output
packets: 674, Multicast packets: 311080
    VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
      Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
```

- 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 (6d 00:28 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
    Bit errors          Seconds
    Errored blocks      0
    Ethernet FEC statistics
      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 : 311115
Output packets: 22
  Protocol eth-switch, MTU: 9192
  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: 158, SNMP ifIndex: 536
```



Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Link-mode: Full-duplex,  
Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,

Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,  
Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,  
Remote fault: Online

Device flags : Present Running

Interface flags: SNMP-Traps Internal: 0x4000

CoS queues : 8 supported, 8 maximum usable queues

Current address: 00:00:5e:00:53:5d, Hardware address: 00:00:5e:00:53:5d

Last flapped : 2018-04-23 06:03:29 UTC (1d 00:04 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-1/0/2.0 (Index 342) (SNMP ifIndex 538)

Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.200 ] Encapsulation: ENET2

Input packets : 0

Output packets: 0

Security: Zone: untrust

Allowed host-inbound traffic : dns dhcp tftp https

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: Sendbroadcast-pkt-to-re

Addresses, Flags: Is-Preferred Is-Primary

Destination: 203.0.113/24, Local: 203.0.113.2, Broadcast: 203.0.113.255

Protocol multiservice, MTU: Unlimited

Logical interface ge-1/0/2.32767 (Index 343) (SNMP ifIndex 545)

Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2

Input packets : 0

```
Output packets: 0
Security: Zone: Null
Protocol multiservice, MTU: Unlimited
Flags: None
```

## RELATED DOCUMENTATION

[Example: Configuring Cross Connect on NFX150 Devices](#) | 156

# 8

CHAPTER

## Troubleshooting

---

Recovering the Root Password for NFX150, NFX250 NextGen, and NFX350  
Devices | **180**

Troubleshooting Interfaces on NFX Devices | **183**

---

# 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 | 184](#)

## Monitoring Interface Status and Traffic on NFX Series Devices

### Purpose

View the interface status to monitor bandwidth utilization and traffic statistics of an interface.

### Action

To view the status of an interface:

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

For example:

- To view the status of an interface for an NFX350 device:

```
user@host> show interfaces ge-0/0/0 | no-more
```

```
Physical interface: ge-0/0/0, Enabled, Physical link is Down
Interface index: 150, SNMP ifIndex: 514
Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Unknown,
Speed: 1000mbps, Duplex: Full-Duplex, BPDU Error: None,
Loop Detect PDU Error: None, Ethernet-Switching Error: None,
MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online,
IEEE 802.3az Energy Efficient Ethernet: Disabled, Auto-MDIX: Enabled
Device flags      : Present Running Down
Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
Link flags       : None
CoS queues       : 12 supported, 12 maximum usable queues
Current address: d0:dd:49:e8:6e:7d, Hardware address: d0:dd:49:e8:6e:7d
Last flapped    : 2020-02-19 06:17:42 UTC (00:25:17 ago)
Input rate      : 0 bps (0 pps)
Output rate     : 0 bps (0 pps)
Active alarms   : LINK
Active defects  : LINK
PCS statistics                                     Seconds
  Bit errors                                         0
  Errored blocks                                    0
Ethernet FEC statistics                             Errors
  FEC Corrected Errors                             0
  FEC Uncorrected Errors                           0
  FEC Corrected Errors Rate                         0
  FEC Uncorrected Errors Rate                       0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled
```



```

Logical interface ge-0/0/0.0 (Index 74) (SNMP ifIndex 523)
  Flags: Device-Down SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
  Input packets : 0
  Output packets: 0
  Protocol eth-switch, MTU: 1514

```

user@host> **show interfaces xe-0/0/15 | no-more**

```

Physical interface: xe-0/0/15, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 557
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps,
  BPDU Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,

  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled
  Device flags      : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags       : None
  CoS queues       : 12 supported, 12 maximum usable queues
  Current address: d0:dd:49:e8:6e:8c, Hardware address: d0:dd:49:e8:6e:8c
  Last flapped     : 2020-02-19 06:17:43 UTC (00:25:32 ago)
  Input rate       : 0 bps (0 pps)
  Output rate      : 232 bps (0 pps)
  Active alarms    : None
  Active defects   : None

  PCS statistics
  Bit errors              Seconds
  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
    Bit errors           0
    Errored blocks       0
  Ethernet FEC statistics
    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: Sendbroadcast-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
```

# 9

CHAPTER

## Operational Commands

---

- `request chassis cluster failover node` | **190**
- `request chassis cluster failover redundancy-group` | **192**
- `request chassis cluster failover reset` | **194**
- `request chassis fpc` | **195**
- `request vmhost cleanup` | **197**
- `request vmhost file-copy` | **198**
- `request vmhost halt` | **200**
- `request vmhost mode` | **202**
- `request vmhost power-off` | **204**
- `request vmhost reboot` | **206**
- `request vmhost software add` | **208**
- `show chassis cluster control-plane statistics` | **212**
- `show chassis cluster data-plane interfaces` | **215**
- `show chassis cluster data-plane statistics` | **217**
- `show chassis cluster information` | **220**

[show chassis cluster interfaces | 226](#)

[show chassis cluster port-peering | 232](#)

[show chassis cluster statistics | 234](#)

[show chassis cluster status | 240](#)

[show interfaces | 244](#)

[show system visibility cpu | 247](#)

[show system visibility host | 251](#)

[show system visibility memory | 262](#)

[show system visibility network | 265](#)

[show system visibility vnf | 271](#)

[show vmhost connections | 278](#)

[show vmhost control-plane | 280](#)

[show vmhost crash | 281](#)

[show vmhost forwarding-options analyzer | 283](#)

[show vmhost memory | 285](#)

[show vmhost mode | 286](#)

[show vmhost status | 290](#)

[show vmhost storage | 292](#)

[show vmhost uptime | 294](#)

[show vmhost version | 296](#)

[show vmhost vlans | 298](#)

---

# request chassis cluster failover node

## Syntax

```
request chassis cluster failover node node-number
redundancy-group group-number
```

## Release Information

Command introduced in Junos OS Release 9.0.

## Description

For chassis cluster configurations, initiate manual failover in a redundancy group from one node to the other, which becomes the primary node, and automatically reset the priority of the group to 255. The failover stays in effect until the new primary node becomes unavailable, the threshold of the redundancy group reaches 0, or you use the **request chassis cluster failover reset** command.

After a manual failover, you must use the **request chassis cluster failover reset** command before initiating another failover.

## Options

- **node *node-number***—Number of the chassis cluster node to which the redundancy group fails over.
- **Range:** 0 through 1
- **redundancy-group *group-number***—Number of the redundancy group on which to initiate manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

**Range:** 0 through 255

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

*clear chassis cluster failover-count*

[request chassis cluster failover reset](#) | 194

[show chassis cluster status](#) | 240

## List of Sample Output

[request chassis cluster failover node on page 191](#)

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**request chassis cluster failover node**

**user@host> request chassis cluster failover node 0 redundancy-group 1**

```
Initiated manual failover for redundancy group 1
```

# request chassis cluster failover redundancy-group

## Syntax

```
request chassis cluster failover node node-number redundancy-group redundancy-group-number
```

## Release Information

Command introduced in Junos OS Release 9.0.

## Description

For chassis cluster configurations, initiate manual failover in a redundancy group from one node to the other, which becomes the primary node, and automatically reset the priority of the group to 255. The failover stays in effect until the new primary node becomes unavailable, the threshold of the redundancy group reaches 0, or you use the **request chassis cluster failover reset** command.

After a manual failover, you must use the **request chassis cluster failover reset** command before initiating another failover.

## Options

- **node** *node-number*—Number of the chassis cluster node to which the redundancy group fails over.
- **Range:** 0 or 1
- **redundancy-group** *group-number*—Number of the redundancy group on which to initiate manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

**Range:** 0 through 255

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

*Initiating a Chassis Cluster Manual Redundancy Group Failover*

*Verifying Chassis Cluster Failover Status*

## List of Sample Output

[request chassis cluster failover redundancy-group on page 193](#)

## Output Fields

When you enter this command, you are provided feedback on the status of your request.



## Sample Output

**request chassis cluster failover redundancy-group**

user@host> **request chassis cluster failover redundancy-group 0 node 1**

```
{primary:node0}
```

```
user@host> request chassis cluster failover redundancy-group 0 node 1
```

```
-----
```

```
Initiated manual failover for redundancy group 0
```

# request chassis cluster failover reset

## Syntax

```
request chassis cluster failover reset  
redundancy-group group-number
```

## Release Information

Command introduced in Junos OS Release 9.0.

## Description

In chassis cluster configurations, undo the previous manual failover and return the redundancy group to its original settings.

## Options

**redundancy-group *group-number***—Number of the redundancy group on which to reset manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

**Range:** 0 through 255

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

[clear chassis cluster failover-count](#)

[request chassis cluster failover node | 190](#)

[show chassis cluster status | 240](#)

## List of Sample Output

[request chassis cluster failover reset on page 194](#)

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**request chassis cluster failover reset**

```
user@host> request chassis cluster failover reset redundancy-group 0
```

# request chassis fpc

## Syntax

```
request chassis fpc (offline | online | restart) slot slot-number
```

## Release Information

Command modified in Junos OS Release 9.2.

Command introduced in Junos OS Release 17.2 for PTX10008 Routers.

## Description

Control the operation of the Flexible PIC Concentrator (FPC).

**NOTE:** The SRX5K-SPC-2-10-40 (SPC1) and SRX5K-SPC-4-15-320 (SPC2) does not support the **request chassis fpc** command. SRX5K-SPC3 card supports **request chassis fpc** command.

## Options

**offline**—Take the FPC offline.

**online**—Bring the FPC online.

**restart**—Restart the FPC.

**slot *slot-number***—Specify the FPC slot number.

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| [show chassis fpc \(View\)](#)

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

### **request chassis fpc (SRX Series)**

```
user@host> request chassis fpc online slot 0
```

```
FPC 0 already online
```

### **request chassis fpc (PTX10008 Router)**

```
user@host> request chassis fpc online slot 1
```

```
FPC 0 already online
```

# request vmhost cleanup

## Syntax

```
request vmhost cleanup
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Clean up temporary files, crash generated files, and log files located in the **/var/tmp**, **/var/crash**, and **/var/log** directories respectively on the host OS.

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| *vmhost*

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

# request vmhost file-copy

## Syntax

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

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Copy crash files or log files from the host OS to Junos OS. You can use these files for analysis and debugging purposes.

## Options

- **crash**—Files in `/var/crash` on the host.
- **from-jnode *filename***—Name of the host file to be copied.
- **log**—Files in `/var/log` on the host.
- **to-vjunos *filename***—Name of the Junos OS file to which the host file is copied.

## Additional Information

You can use the **show vmhost crash** and **show vmhost logs** commands to list or identify the files in the host OS to be copied to Junos OS.

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[request vmhost file-copy on page 198](#)

## Sample Output

**request vmhost file-copy**

```
user@host> request vmhost file-copy log from-jnode daemon.log to-vjunos /var/tmp
```

```
:/var/tmp # ls -lrt daemon.log  
-rw-r--r-- 1 root  wheel 1035126 Mar  4 20:33 daemon.log
```

# request vmhost halt

## Syntax

```
request vmhost halt
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

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

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[request vmhost halt on page 200](#)

## Sample Output

### request vmhost halt

```
user@host> request vmhost halt
```

```
Halt the vmhost ? [yes,no] (no) yes

Initiating vmhost halt... ok
Initiating Junos shutdown...  shutdown: [pid 8782]
Shutdown NOW!
ok
Junos shutdown is in progress...

*** FINAL System shutdown message from root@ ***

System going down IMMEDIATELY
```



...

...

Operating System halted

Please press any key to reboot

# request vmhost mode

## Syntax

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

## Release Information

Command introduced in Junos OS Release 19.1R1 for NFX150 and NFX250 (NG) devices.

## Description

Select the operational mode of the device.

### NOTE:

- Starting from Junos OS Release 19.3R1, if the same physical CPU is used for both VNFs and the Junos OS or device components, the request to change the mode fails and an error message is displayed. For example:

```
root> request vmhost mode throughput
```

```
error: Mode cannot be changed; Reason: Reserved CPUs conflict with VNF  
cpu pinnings: 3
```

- When you upgrade the software image that has a VNF CPU conflict to Junos OS Release 19.3R1 by using the CLI upgrade option, the upgrade succeeds and the VNF configuration is applied. The VNF CPU conflict is reported by JDM only if you issue a **commit** command. You must modify the VNF configurations accordingly.

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[request vmhost mode compute on page 203](#)

## Sample Output

**request vmhost mode compute**

user@host> **request vmhost mode compute**

```
warning: Device will be rebooted to change the mode from hybrid to compute
Do you want to continue? [yes,no] (no)
```

# request vmhost power-off

## Syntax

```
request vmhost power-off
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

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

## Description

Shut down the Junos OS software and the host OS.

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[request vmhost power-off on page 204](#)

## Sample Output

### request vmhost power-off

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

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

```
Initiating vmhost shutdown... ok
```

```
Initiating Junos shutdown... shutdown: [pid 3884]
```

```
Shutdown NOW!
```

```
ok
```

```
*** FINAL System shutdown message from root@host ***
```

```
System going down IMMEDIATELY
```

```
...
```

```
...
```

# request vmhost reboot

## Syntax

```
request vmhost reboot
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Reboot the Junos OS software and the host OS.

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[request vmhost reboot on page 206](#)

## Sample Output

### request vmhost reboot

```
user@host> request vmhost reboot
```

```
Reboot the vmhost ? [yes,no] (no) yes

warning: Rebooting re0
Initiating vmhost reboot... ok
Initiating Junos shutdown... shutdown: [pid 7273]
Shutdown NOW!
ok
Junos shutdown is in progress...

*** FINAL System shutdown message from root@ ***
```

System going down IMMEDIATELY

...

...

# request vmhost software add

## Syntax

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

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Install or upgrade the Junos OS and host software packages on the device.

## Options

- **in**—(Optional) Number of minutes to delay before the reboot operation.
- **no-validate**—(Optional) When loading a software package or bundle with a different release, suppress the default behavior of the **validate** option.
- **reboot**—(Optional) After adding the software package or bundle, reboot the system.
- **set**—(Optional) List of URLs or pathnames corresponding to the software packages.
- **unlink**—(Optional) Removes the software package after successful installation.
- **upgrade-to-model**—(Optional) *model number*—(Optional) Name of the model to upgrade to.

## Required Privilege Level

maintenance

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[request vmhost software add \(NFX150\) on page 209](#)

[request vmhost software add \(NFX250 \(NG\)\) on page 209](#)



## Sample Output

### request vmhost software add (NFX150)

```
user@host> request vmhost software add
/var/public/jinstall-host-nfx-3-x86-64-18.1R1.8-secure-signed.tgz no-validate reboot
```

```
Verified jinstall-host-nfx-3-x86-64-18.1R1.8-secure-signed signed by
PackageProductionEc_2018 method ECDSA256+SHA256
Pushing Junos image package to the host...
File already present in Host. Skipping pushing the image
Mounting primary partitions to stage upgrade operation
Installing
/mnt/.share/lshare/public/pkginst.7565/install-media-nfx-3-junos-18.1R1.8-secure.tgz
Extracting the package ...
..
..
```

### request vmhost software add (NFX250 (NG))

```
user@host> request vmhost software add
/var/public/jinstall-host-nfx-3-x86-64-18.4R1.8-secure-signed.tgz
```

```
Verified jinstall-host-nfx-3-x86-64-18.4R1.8-secure-signed signed by
PackageProductionEc_2018 method ECDSA256+SHA256
Pushing Junos image package to the host...
File already present in Host. Skipping pushing the image
Mounting alternate partitions to stage upgrade operation
Installing
/mnt/.share/lshare/public/pkginst.39634/install-media-nfx-3-junos-18.4R1.8-secure.tgz
Extracting the package ...
=====
Host OS upgrade is FORCED
Current Host kernel version : 4.1.27-rt30-WR8.0.0.25_ovp
Package Host kernel version : 4.1.27-rt30-WR8.0.0.25_ovp
Current Host version       : 3.0.3
Package Host version       : 3.0.3
Min host version required for applications: 3.0.2
=====
Validate linux image...
upgrade_platform: -----
upgrade_platform: Parameters passed:
upgrade_platform: silent=0
upgrade_platform:
```

```

package=/var/tmp/tmp.rV7S1sxWedjunos_cli_upg/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
upgrade_platform: clean install=0
upgrade_platform: on primary    =0
upgrade_platform: clean upgrade=0
upgrade_platform: Need reboot after staging=1
upgrade_platform: -----
upgrade_platform:
upgrade_platform: Checking input
/var/tmp/tmp.rV7S1sxWedjunos_cli_upg/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
...
upgrade_platform: Input package
/var/tmp/tmp.rV7S1sxWedjunos_cli_upg/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
is valid.
Secure Boot is enforced.
ALLOW:usr/secureboot/grub/BOOTX64.EFI
ALLOW:boot/bzImage-intel-x86-64.bin
ALLOW:boot/initramfs.cpio.gz
Setting up Junos host applications for installation ...
Current junos instance is 0
Installing Host OS ...
upgrade_platform: -----
upgrade_platform: Parameters passed:
upgrade_platform: silent=0
upgrade_platform: package=/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
upgrade_platform: clean install=0
upgrade_platform: on primary    =0
upgrade_platform: clean upgrade=0
upgrade_platform: Need reboot after staging=0
upgrade_platform: -----
upgrade_platform:
upgrade_platform: Checking input
/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz ...
upgrade_platform: Input package
/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz is valid.
Secure Boot is enforced.
ALLOW:usr/secureboot/grub/BOOTX64.EFI
ALLOW:boot/bzImage-intel-x86-64.bin
ALLOW:boot/initramfs.cpio.gz
upgrade_platform: Backing up boot assets..
upgrade_platform: Staging the upgrade package -
/var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz..
upgrade_platform: Checksum verified and OK...
upgrade_platform: Staging of /var/tmp/jinstall-nfx-3-junos-18.4R1.8-secure-linux.tgz
completed

```

```
upgrade_platform: System needs *REBOOT* to complete the upgrade  
Host OS upgrade staged. Reboot the system to complete installation!
```

# show chassis cluster control-plane statistics

Syntax

```
show chassis cluster control-plane statistics
```

Release Information

Command introduced in Junos OS Release 9.3. Output changed to support dual control ports in Junos OS Release 10.0.

Description

Display information about chassis cluster control plane statistics.

Required Privilege Level

view

RELATED DOCUMENTATION

| *clear chassis cluster control-plane statistics*

List of Sample Output

- [show chassis cluster control-plane statistics on page 213](#)
- [show chassis cluster control-plane statistics \(SRX5000 Line Devices\) on page 214](#)

Output Fields

[Table 23 on page 212](#) lists the output fields for the **show chassis cluster control-plane statistics** command. Output fields are listed in the approximate order in which they appear.

Table 23: show chassis cluster control-plane statistics Output Fields

Field Name	Field Description
Control link statistics	<p>Statistics of the control link used by chassis cluster traffic. Statistics for <b>Control link 1</b> are displayed when you use dual control links (SRX5600 and SRX5800 devices only).</p> <ul style="list-style-type: none"><li>• <b>Heartbeat packets sent</b>—Number of heartbeat messages sent on the control link.</li><li>• <b>Heartbeat packets received</b>—Number of heartbeat messages received on the control link.</li><li>• <b>Heartbeat packet errors</b>—Number of heartbeat packets received with errors on the control link.</li></ul>

Table 23: show chassis cluster control-plane statistics Output Fields (continued)

Field Name	Field Description
<b>Fabric link statistics</b>	<p>Statistics of the fabric link used by chassis cluster traffic. Statistics for <b>Child Link 1</b> are displayed when you use dual fabric links.</p> <ul style="list-style-type: none"> <li>• <b>Probes sent</b>—Number of probes sent on the fabric link.</li> <li>• <b>Probes received</b>—Number of probes received on the fabric link.</li> </ul>
<b>Switch fabric link statistics</b>	<p>Statistics of the switch fabric link used by chassis cluster traffic.</p> <ul style="list-style-type: none"> <li>• <b>Probe state</b>—State of the probe, <b>UP</b> or <b>DOWN</b>.</li> <li>• <b>Probes sent</b>—Number of probes sent.</li> <li>• <b>Probes received</b>—Number of probes received.</li> <li>• <b>Probe recv error</b> —Error in receiving probe.</li> <li>• <b>Probe send error</b>—Error in sending probe.</li> </ul>

## Sample Output

**show chassis cluster control-plane statistics**

user@host> **show chassis cluster control-plane statistics**

```
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 11646
    Heartbeat packets received: 8343
    Heartbeat packet errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 11644
    Probes received: 8266
Switch fabric link statistics:
  Probe state : DOWN
  Probes sent: 8145
  Probes received: 8013
  Probe recv errors: 0
  Probe send errors: 0
```

## Sample Output

**show chassis cluster control-plane statistics (SRX5000 Line Devices)**

user@host> **show chassis cluster control-plane statistics**

```
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 2061982
    Heartbeat packets received: 2060367
    Heartbeat packet errors: 0
  Control link 1:
    Heartbeat packets sent: 2061982
    Heartbeat packets received: 0
    Heartbeat packet errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 3844342
    Probes received: 3843841
  Child link 1
    Probes sent: 0
    Probes received: 0
```

# show chassis cluster data-plane interfaces

Syntax

```
show chassis cluster data-plane interfaces
```

Release Information

Command introduced in Junos OS Release 10.2.

Description

Display the status of the data plane interface (also known as a fabric interface) in a chassis cluster configuration.

Required Privilege Level

view

RELATED DOCUMENTATION

| *cluster (Chassis)*

List of Sample Output

[show chassis cluster data-plane interfaces on page 215](#)

Output Fields

[Table 24 on page 215](#) lists the output fields for the **show chassis cluster data-plane interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 24: show chassis cluster data-plane interfaces Output Fields

Field Name	Field Description
fab0/fab1	<div>Name of the logical fabric interface.<ul style="list-style-type: none"><li>• <b>Name</b>—Name of the physical Ethernet interface.</li><li>• <b>Status</b>—State of the fabric interface: <b>up</b> or <b>down</b>.</li></ul></div>

## Sample Output

show chassis cluster data-plane interfaces

user@host> show chassis cluster data-plane interfaces

```
fab0:
    Name                Status
    ge-2/1/9            up
    ge-2/2/5            up

fab1:
    Name                Status
    ge-8/1/9            up
    ge-8/2/5            up
```



# show chassis cluster data-plane statistics

## Syntax

```
show chassis cluster data-plane statistics
```

## Release Information

Command introduced in Junos OS Release 9.3.

## Description

Display information about chassis cluster data plane statistics.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| *clear chassis cluster data-plane statistics*

## List of Sample Output

[show chassis cluster data-plane statistics on page 218](#)

## Output Fields

[Table 25 on page 218](#) lists the output fields for the **show chassis cluster data-plane statistics** command. Output fields are listed in the approximate order in which they appear.

Table 25: show chassis cluster data-plane statistics Output Fields

Field Name	Field Description
Services Synchronized	<ul style="list-style-type: none"> <li>• <b>Service name</b>—Name of the service.</li> <li>• <b>Rtos sent</b>—Number of runtime objects (RTOs) sent.</li> <li>• <b>Rtos received</b>—Number of RTOs received.</li> <li>• <b>Translation context</b>—Messages synchronizing Network Address Translation (NAT) translation context.</li> <li>• <b>Incoming NAT</b>—Messages synchronizing incoming Network Address Translation (NAT) service.</li> <li>• <b>Resource manager</b>—Messages synchronizing resource manager groups and resources.</li> <li>• <b>Session create</b>—Messages synchronizing session creation.</li> <li>• <b>Session close</b>—Messages synchronizing session close.</li> <li>• <b>Session change</b>—Messages synchronizing session change.</li> <li>• <b>Gate create</b>—Messages synchronizing creation of pinholes (temporary openings in the firewall).</li> <li>• <b>Session ageout refresh request</b>—Messages synchronizing request session after age-out.</li> <li>• <b>Session ageout refresh reply</b>—Messages synchronizing reply session after age-out.</li> <li>• <b>IPsec VPN</b>—Messages synchronizing VPN session.</li> <li>• <b>Firewall user authentication</b>—Messages synchronizing firewall user authentication session.</li> <li>• <b>MGCP ALG</b>—Messages synchronizing MGCP ALG sessions.</li> <li>• <b>H323 ALG</b>—Messages synchronizing H.323 ALG sessions.</li> <li>• <b>SIP ALG</b>—Messages synchronizing SIP ALG sessions.</li> <li>• <b>SCCP ALG</b>—Messages synchronizing SCCP ALG sessions.</li> <li>• <b>PPTP ALG</b>—Messages synchronizing PPTP ALG sessions.</li> <li>• <b>RTSP ALG</b>—Messages synchronizing RTSP ALG sessions.</li> </ul>

## Sample Output

```
show chassis cluster data-plane statistics
```

```
user@host> show chassis cluster data-plane statistics
```

## Services Synchronized:

Service name	RTOS sent	RTOS received
Translation context	0	0
Incoming NAT	0	0
Resource manager	0	0
Session create	0	0
Session close	0	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPsec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RTSP ALG	0	0

# show chassis cluster information

## Syntax

```
show chassis cluster information
```

## Release Information

Command introduced in Junos OS Release 12.1X47-D10.

## Description

Display chassis cluster messages. The messages indicate each node's health condition and details of the monitored failure.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [show chassis cluster status](#) | [240](#)

## List of Sample Output

- [show chassis cluster information on page 221](#)
- [show chassis cluster information \(Monitoring Abnormal Case\) on page 222](#)
- [show chassis cluster information \(Preempt Delay Timer\) on page 224](#)

## Output Fields

[Table 26 on page 220](#) lists the output fields for the **show chassis cluster information** command. Output fields are listed in the approximate order in which they appear.

Table 26: show chassis cluster information Output Fields

Field Name	Field Description
Node	Node (device) in the chassis cluster ( <b>node0</b> or <b>node1</b> ).

Table 26: show chassis cluster information Output Fields (*continued*)

Field Name	Field Description
Redundancy Group Information	<ul style="list-style-type: none"> <li>Redundancy Group—ID number (0 - 255) of a redundancy group in the cluster.</li> <li>Current State—State of the redundancy group: <b>primary</b>, <b>secondary</b>, <b>hold</b>, or <b>secondary-hold</b>.</li> <li>Weight—Relative importance of the redundancy group.</li> <li>Time—Time when the redundancy group changed the state.</li> <li>From—State of the redundancy group before the change.</li> <li>To—State of the redundancy group after the change.</li> <li>Reason—Reason for the change of state of the redundancy group.</li> </ul>
Chassis cluster LED information	<ul style="list-style-type: none"> <li>Current LED color—Current color state of the LED.</li> <li>Last LED change reason—Reason for change of state of the LED.</li> </ul>

## Sample Output

show chassis cluster information

user@host> show chassis cluster information

```
node0:
-----
Redundancy Group Information:

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

Time           From           To           Reason
Mar 27 17:44:19 hold           secondary    Hold timer expired
Mar 27 17:44:27 secondary    primary      Better priority (200/200)

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

Time           From           To           Reason
Mar 27 17:44:19 hold           secondary    Hold timer expired
Mar 27 17:44:27 secondary    primary      Remote yield (0/0)

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

Time           From           To           Reason
```

```

Mar 27 17:44:19 hold          secondary    Hold timer expired
Mar 27 17:44:27 secondary    primary     Remote yield (0/0)
Mar 27 17:50:24 primary      secondary-hold Preempt/yield(100/200)
Mar 27 17:50:25 secondary-hold secondary    Ready to become secondary

```

Chassis cluster LED information:

Current LED color: Green

Last LED change reason: No failures

node1:

-----  
Redundancy Group Information:

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

Time	From	To	Reason
Mar 27 17:44:27	hold	secondary	Hold timer expired

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

Time	From	To	Reason
Mar 27 17:44:27	hold	secondary	Hold timer expired
Mar 27 17:50:23	secondary	primary	Remote yield (100/0)
Mar 27 17:50:24	primary	secondary-hold	Preempt/yield(100/200)
Mar 27 17:50:25	secondary-hold	secondary	Ready to become secondary

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

Time	From	To	Reason
Mar 27 17:44:27	hold	secondary	Hold timer expired
Mar 27 17:50:23	secondary	primary	Remote yield (200/0)

Chassis cluster LED information:

Current LED color: Green

Last LED change reason: No failures

## Sample Output

**show chassis cluster information (Monitoring Abnormal Case)**

```
user@host> show chassis cluster information
```

The following output is specific to monitoring abnormal (unhealthy) case.

node0:

-----  
Redundancy Group Information:

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

Time	From	To	Reason
Apr 1 11:07:38	hold	secondary	Hold timer expired
Apr 1 11:07:41	secondary	primary	Only node present
Apr 1 11:29:20	primary	secondary-hold	Manual failover
Apr 1 11:34:20	secondary-hold	secondary	Ready to become secondary

Redundancy Group 1 , Current State: primary, Weight: 0

Time	From	To	Reason
Apr 1 11:07:38	hold	secondary	Hold timer expired
Apr 1 11:07:41	secondary	primary	Only node present

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

Time	From	To	Reason
Apr 1 11:07:38	hold	secondary	Hold timer expired
Apr 1 11:07:41	secondary	primary	Only node present

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

Failure Information:

IP Monitoring Failure Information:

Redundancy Group 1, Monitoring Status: Failed

IP Address	Status	Reason
1.1.1.1	Unreachable	redundancy-group state unknown

node1:

-----  
Redundancy Group Information:

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

Time	From	To	Reason
Apr 1 11:08:40	hold	secondary	Hold timer expired

```
Apr  1 11:29:20 secondary      primary      Remote is in secondary hold
```

```
Redundancy Group 1 , Current State: secondary, Weight: 0
```

Time	From	To	Reason
Apr  1 11:08:40	hold	secondary	Hold timer expired

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

Time	From	To	Reason
Apr  1 11:08:40	hold	secondary	Hold timer expired

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

Failure Information:

IP Monitoring Failure Information:

Redundancy Group 1, Monitoring Status: Failed

IP Address	Status	Reason
1.1.1.1	Unreachable	redundancy-group state unknown

## Sample Output

**show chassis cluster information (Preempt Delay Timer)**

```
user@host> show chassis cluster information
```

```
node0:
```

```
-----  
Redundancy Group Information:
```

```
Redundancy Group 0 , Current State: secondary, Weight: 255
```

Time	From	To	Reason
Aug  4 12:30:02	hold	secondary	Hold timer expired
Aug  4 12:30:05	secondary	primary	Only node present
Aug  4 14:19:58	primary	secondary-hold	Manual failover
Aug  4 14:24:58	secondary-hold	secondary	Ready to become secondary



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

Time	From	To	Reason
Aug 4 14:07:57	secondary	primary	Remote is in secondary hold
Aug 4 14:20:23	primary	secondary-hold	Monitor failed: IF
Aug 4 14:20:24	secondary-hold	secondary	Ready to become secondary
Aug 4 14:20:54	secondary	primary	Remote is in secondary hold
Aug 4 14:21:30	primary	secondary-hold	Monitor failed: IF
Aug 4 14:21:31	secondary-hold	secondary	Ready to become secondary

Chassis cluster LED information:

Current LED color: Green  
Last LED change reason: No failures

node1:  
-----  
Redundancy Group Information:

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

Time	From	To	Reason
Aug 4 12:33:47	hold	secondary	Hold timer expired
Aug 4 14:19:57	secondary	primary	Remote is in secondary hold

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

Time	From	To	Reason
Aug 4 14:07:56	secondary-hold	secondary	Ready to become secondary
Aug 4 14:20:22	secondary	primary	Remote is in secondary hold
Aug 4 14:20:37	primary	primary-preempt-hold	Preempt (99/101)
Aug 4 14:20:52	primary-preempt-hold	secondary-hold	Primary preempt hold timer e
Aug 4 14:20:53	secondary-hold	secondary	Ready to become secondary
Aug 4 14:21:28	secondary	primary	Remote yield (99/0)

Chassis cluster LED information:

Current LED color: Green  
Last LED change reason: No failures

# show chassis cluster interfaces

## Syntax

```
show chassis cluster interfaces
```

## Release Information

Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0. Output changed to support control interfaces in Junos OS Release 11.2. Output changed to support redundant pseudo interfaces in Junos OS Release 12.1X44-D10. For SRX5000 line devices, output changed to support the internal security association (SA) option in Junos OS Release 12.1X45-D10. Output changed to support MACsec status on control and fabric interfaces in Junos OS Release 15.1X49-D60. For vSRX, output changed to support the internal security association (SA) option in Junos OS Release 19.4R1.

## Description

Display the status of the control interface in a chassis cluster configuration.

## Required Privilege Level

view

## RELATED DOCUMENTATION

*cluster (Chassis)*

## List of Sample Output

[show chassis cluster interfaces \(SRX5000 line devices with RE3\) on page 228](#)

[show chassis cluster interfaces \(SRX5000 line devices with RE2\) on page 228](#)

[show chassis cluster interfaces on page 229](#)

[show chassis cluster interfaces \(SRX5400, SRX5600, and SRX5800 Devices with SRX5000 line SRX5K-SCB3 \[SCB3\] with Enhanced Midplanes and SRX5K-MPC3-100G10G \[IOC3\] or SRX5K-MPC3-40G10G \[IOC3\]\) on page 230](#)

[show chassis cluster interfaces \(vSRX\) on page 231](#)

## Output Fields

[Table 27 on page 227](#) lists the output fields for the **show chassis cluster interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 27: show chassis cluster interfaces Output Fields

Field Name	Field Description
Control link status	State of the chassis cluster control interface: <b>up</b> or <b>down</b> .
Control interfaces	<ul style="list-style-type: none"> <li>• <b>Index</b>—Index number of the chassis cluster control interface.</li> <li>• <b>Interface</b>—Name of the chassis cluster control interface. The control interface names differ based on the routing engine. For RE2, the control interfaces are displayed as em0 and em1 and for RE3, the control interfaces are displayed as ixlv0 and igb0.</li> <li>• <b>Monitored-Status</b>—Monitored state of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Internal SA</b>—State of the internal SA option on the chassis cluster control link: <b>enabled</b> or <b>disabled</b>.  NOTE: This field is available only on SRX5000 line devices.</li> <li>• <b>Security</b>—State of MACsec on chassis cluster control interfaces.</li> </ul>
Fabric link status	State of the fabric interface: <b>up</b> or <b>down</b> .
Fabric interfaces	<ul style="list-style-type: none"> <li>• <b>Name</b>—Name of the fabric interface.</li> <li>• <b>Child-interface</b>—Name of the child fabric interface.</li> <li>• <b>Status</b>—State of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Security</b>—State of MACsec on chassis cluster fabric interfaces.</li> </ul>
Redundant-ethernet Information	<ul style="list-style-type: none"> <li>• <b>Name</b>—Name of the redundant Ethernet interface.</li> <li>• <b>Status</b>—State of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Redundancy-group</b>—Identification number (1–255) of the redundancy group associated with the redundant Ethernet interface.</li> </ul>
Redundant-pseudo-interface Information	<ul style="list-style-type: none"> <li>• <b>Name</b>—Name of the redundant pseudointerface.</li> <li>• <b>Status</b>—State of the redundant pseudointerface: <b>up</b> or <b>down</b>.</li> <li>• <b>Redundancy-group</b>—Identification number (1–255) of the redundancy group associated with the redundant pseudointerface.</li> </ul>
Interface Monitoring	<ul style="list-style-type: none"> <li>• <b>Interface</b>—Name of the interface to be monitored.</li> <li>• <b>Weight</b>—Relative importance of the interface to redundancy group operation.</li> <li>• <b>Status</b>—State of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Redundancy-group</b>—Identification number of the redundancy group associated with the interface.</li> </ul>

## Sample Output

**show chassis cluster interfaces (SRX5000 line devices with RE3)**

user@host> **show chassis cluster interfaces**

Control link status: Down

Control interfaces:

Index	Interface	Monitored-Status	Internal-SA	Security
0	ixlv0	Down	Enabled	Disabled
1	igb0	Down	Enabled	Disabled

Fabric link status: Down

Fabric interfaces:

Name	Child-interface	Status (Physical/Monitored)	Security
fab0			
fab0			

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Down	Not configured
reth1	Down	Not configured
reth2	Down	Not configured
reth3	Down	Not configured
reth4	Down	Not configured

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

## Sample Output

**show chassis cluster interfaces (SRX5000 line devices with RE2)**

user@host> **show chassis cluster interfaces**

Control link status: Up

## Control interfaces:

Index	Interface	Monitored-Status	Internal-SA	Security
0	em0	Up	Disabled	Disabled
1	em1	Down	Disabled	Disabled

Fabric link status: Up

## Fabric interfaces:

Name	Child-interface	Status (Physical/Monitored)	Security
fab0	xe-1/0/3	Up / Down	Disabled
fab1	xe-7/0/3	Up / Down	Disabled

## Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	2
reth2	Down	Not configured
reth3	Down	Not configured
reth4	Down	Not configured
reth5	Down	Not configured
reth6	Down	Not configured
reth7	Down	Not configured
reth8	Down	Not configured
reth9	Down	Not configured
reth10	Down	Not configured
reth11	Down	Not configured

## Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	1

## Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-0/1/9	100	Up	0
ge-0/1/9	100	Up	

## Sample Output

**show chassis cluster interfaces**

user@host> **show chassis cluster interfaces**

The following output is specific to fabric monitoring failure:

```
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      em0      Up                Disabled     Disabled

Fabric link status: Down

Fabric interfaces:
  Name    Child-interface  Status              Security
              (Physical/Monitored)
  fab0    ge-0/0/2        Down / Down        Disabled
  fab0
  fab1    ge-9/0/2        Up   / Up          Disabled
  fab1

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

## Sample Output

**show chassis cluster interfaces**  
 (SRX5400, SRX5600, and SRX5800 Devices with SRX5000 line SRX5K-SCB3 [SCB3] with Enhanced Midplanes and SRX5K-MPC3-100G10G [IOC3] or SRX5K-MPC3-40G10G [IOC3])

```
user@host> show chassis cluster interfaces
```

The following output is specific to SRX5400, SRX5600, and SRX5800 devices in a chassis cluster cluster, when the PICs containing fabric links on the SRX5K-MPC3-40G10G (IOC3) are powered off to turn on alternate PICs. If no alternate fabric links are configured on the PICs that are turned on, RTO synchronous communication between the two nodes stops and the chassis cluster session state will not back up, because the fabric link is missing.

```
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
```

0	em0	Up	Disabled	Disabled
1	em1	Down	Disabled	Disabled

Fabric link status: Down

Fabric interfaces:

Name	Child-interface	Status	Security
(Physical/Monitored)			
fab0	<<< fab child missing once PIC off lined		Disabled
fab0			
fab1	xe-10/2/7	Up / Down	Disabled
fab1			

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	Not configured
reth1	Down	1

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

## Sample Output

**show chassis cluster interfaces (vSRX)**

user@host> **show chassis cluster interfaces**

The following output is specific to view control link status with internal SAs.

Control link status: Up

Control interfaces:

Index	Interface	Status	Internal SA
305			
0	em0	Up	enabled
306			
1	em1	Down	enabled

# show chassis cluster port-peering

## Syntax

```
show chassis cluster port-peering
```

## Release Information

Command introduced in Junos OS Release 19.1.

## Description

Display the status of the L3 back-end interface and the corresponding L2 mapped interface.

Before running this command, you must deploy the chassis-cluster and map the L2 and L3 interfaces:

```
user@host# set groups $node chassis cluster redundant-interface L3 interface
mapping-interface L2 interface
```

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [cluster \(Chassis\)](#)

## List of Sample Output

[show chassis cluster port-peering on page 233](#)

## Output Fields

[Table 28 on page 232](#) lists the output fields for the **show chassis cluster port-peering** command. Output fields are listed in the approximate order in which they appear.

Table 28: show chassis cluster port-peering Output Fields

Field Name	Field Description
Backend L3 Interface	L3 interface of the peer map.
Backend L3 Interface status	Status of the L3 interface.
Mapped L2 Interface	L2 interface of the peer map.
Mapped L2 Interface status	Status of the L2 interface.



## Sample Output

**show chassis cluster port-peering**

user@host> **show chassis cluster port-peering**

```
{primary:node0}
user@host> show chassis cluster port-peering
node0:
-----

Port peering interfaces:
      Backend L3           Mapping L2
      Interface  Status       Interface  Status
      ge-1/0/1   Up           ge-0/0/1   Up

node1:
-----

Port peering interfaces:
      Backend L3           Mapping L2
      Interface  Status       Interface  Status
      ge-8/0/1   Up           ge-7/0/1   Up
```

# show chassis cluster statistics

## Syntax

```
show chassis cluster statistics
```

## Release Information

Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0.

## Description

Display information about chassis cluster services and interfaces.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [clear chassis cluster statistics](#)

## List of Sample Output

[show chassis cluster statistics on page 236](#)

[show chassis cluster statistics \(SRX5000 Line Devices\) on page 237](#)

[show chassis cluster statistics \(SRX5000 Line Devices\) on page 238](#)

## Output Fields

[Table 29 on page 235](#) lists the output fields for the **show chassis cluster statistics** command. Output fields are listed in the approximate order in which they appear.

Table 29: show chassis cluster statistics Output Fields

Field Name	Field Description
<b>Control link statistics</b>	<p>Statistics of the control link used by chassis cluster traffic. Statistics for <b>Control link 1</b> are displayed when you use dual control links (SRX5000 lines only). Note that the output for the SRX5000 lines will always show <b>Control link 0</b> and <b>Control link 1</b> statistics, even though only one control link is active or working.</p> <ul style="list-style-type: none"> <li>• <b>Heartbeat packets sent</b>—Number of heartbeat messages sent on the control link.</li> <li>• <b>Heartbeat packets received</b>—Number of heartbeat messages received on the control link.</li> <li>• <b>Heartbeat packet errors</b>—Number of heartbeat packets received with errors on the control link.</li> </ul>
<b>Fabric link statistics</b>	<p>Statistics of the fabric link used by chassis cluster traffic. Statistics for <b>Child Link 1</b> are displayed when you use dual fabric links.</p> <ul style="list-style-type: none"> <li>• <b>Probes sent</b>—Number of probes sent on the fabric link.</li> <li>• <b>Probes received</b>—Number of probes received on the fabric link.</li> </ul>

Table 29: show chassis cluster statistics Output Fields (*continued*)

Field Name	Field Description
Services Synchronized	<ul style="list-style-type: none"> <li>• <b>Service name</b>—Name of the service.</li> <li>• <b>Rtos sent</b>—Number of runtime objects (RTOs) sent.</li> <li>• <b>Rtos received</b>—Number of RTOs received.</li> <li>• <b>Translation context</b>—Messages synchronizing Network Address Translation (NAT) translation context.</li> <li>• <b>Incoming NAT</b>—Messages synchronizing incoming Network Address Translation (NAT) service.</li> <li>• <b>Resource manager</b>—Messages synchronizing resource manager groups and resources.</li> <li>• <b>Session create</b>—Messages synchronizing session creation.</li> <li>• <b>Session close</b>—Messages synchronizing session close.</li> <li>• <b>Session change</b>—Messages synchronizing session change.</li> <li>• <b>Gate create</b>—Messages synchronizing creation of pinholes (temporary openings in the firewall).</li> <li>• <b>Session ageout refresh request</b>—Messages synchronizing request session after age-out.</li> <li>• <b>Session ageout refresh reply</b>—Messages synchronizing reply session after age-out.</li> <li>• <b>IPsec VPN</b>—Messages synchronizing VPN session.</li> <li>• <b>Firewall user authentication</b>—Messages synchronizing firewall user authentication session.</li> <li>• <b>MGCP ALG</b>—Messages synchronizing MGCP ALG sessions.</li> <li>• <b>H323 ALG</b>—Messages synchronizing H.323 ALG sessions.</li> <li>• <b>SIP ALG</b>—Messages synchronizing SIP ALG sessions.</li> <li>• <b>SCCP ALG</b>—Messages synchronizing SCCP ALG sessions.</li> <li>• <b>PPTP ALG</b>—Messages synchronizing PPTP ALG sessions.</li> <li>• <b>RTSP ALG</b>—Messages synchronizing RTSP ALG sessions.</li> <li>• <b>MAC address learning</b>—Messages synchronizing MAC address learning.</li> </ul>

## Sample Output

```
show chassis cluster statistics
```

```
user@host> show chassis cluster statistics
```

```

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 798
    Heartbeat packets received: 784
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 793
    Probes received: 0
Services Synchronized:
  Service name                                RTOs sent    RTOs received
  Translation context                          0             0
  Incoming NAT                                0             0
  Resource manager                            0             0
  Session create                              0             0
  Session close                               0             0
  Session change                              0             0
  Gate create                                 0             0
  Session ageout refresh requests             0             0
  Session ageout refresh replies             0             0
  IPsec VPN                                   0             0
  Firewall user authentication               0             0
  MGCP ALG                                    0             0
  H323 ALG                                    0             0
  SIP ALG                                     0             0
  SCCP ALG                                    0             0
  PPTP ALG                                    0             0
  RTSP ALG                                    0             0
  MAC address learning                       0             0

```

## Sample Output

**show chassis cluster statistics (SRX5000 Line Devices)**

user@host> **show chassis cluster statistics**

```

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
  Control link 1:

```

```

    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
  Child link 1
    Probes sent: 258501
    Probes received: 258501
Services Synchronized:
  Service name                                RTOs sent    RTOs received
  Translation context                          0             0
  Incoming NAT                                0             0
  Resource manager                            0             0
  Session create                              1             0
  Session close                               1             0
  Session change                              0             0
  Gate create                                 0             0
  Session ageout refresh requests              0             0
  Session ageout refresh replies              0             0
  IPSec VPN                                   0             0
  Firewall user authentication                 0             0
  MGCP ALG                                    0             0
  H323 ALG                                    0             0
  SIP ALG                                     0             0
  SCCP ALG                                    0             0
  PPTP ALG                                    0             0
  RPC ALG                                     0             0
  RTSP ALG                                    0             0
  RAS ALG                                     0             0
  MAC address learning                         0             0
  GPRS GTP                                    0             0

```

## Sample Output

**show chassis cluster statistics (SRX5000 Line Devices)**

user@host> **show chassis cluster statistics**

```

Control link statistics:
  Control link 0:

```

```

    Heartbeat packets sent: 82371
    Heartbeat packets received: 82321
    Heartbeat packets errors: 0
Control link 1:
    Heartbeat packets sent: 0
    Heartbeat packets received: 0
    Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258681
        Probes received: 258681
    Child link 1
        Probes sent: 258501
        Probes received: 258501
Services Synchronized:
    Service name                                RTOs sent    RTOs received
    Translation context                          0             0
    Incoming NAT                                0             0
    Resource manager                            0             0
    Session create                              1             0
    Session close                              1             0
    Session change                             0             0
    Gate create                                0             0
    Session ageout refresh requests             0             0
    Session ageout refresh replies             0             0
    IPSec VPN                                  0             0
    Firewall user authentication                0             0
    MGCP ALG                                   0             0
    H323 ALG                                   0             0
    SIP ALG                                    0             0
    SCCP ALG                                   0             0
    PPTP ALG                                   0             0
    RPC ALG                                    0             0
    RTSP ALG                                   0             0
    RAS ALG                                    0             0
    MAC address learning                       0             0
    GPRS GTP                                  0             0

```

# show chassis cluster status

## Syntax

```
show chassis cluster status
<redundancy-group group-number >
```

## Release Information

Support for monitoring failures added in Junos OS Release 12.1X47-D10.

## Description

Display the current status of the Chassis Cluster. You can use this command to check the status of chassis cluster nodes, redundancy groups, and failover status.

## Options

- none—Display the status of all redundancy groups in the chassis cluster.
- **redundancy-group group-number**—(Optional) Display the status of the specified redundancy group.

## Required Privilege Level

view

## RELATED DOCUMENTATION

<i>redundancy-group (Chassis Cluster)</i>
<i>clear chassis cluster failover-count</i>
<a href="#">request chassis cluster failover node   190</a>
<a href="#">request chassis cluster failover reset   194</a>

## List of Sample Output

- [show chassis cluster status on page 241](#)
- [show chassis cluster status with preemptive delay on page 242](#)
- [show chassis cluster status redundancy-group 1 on page 243](#)

## Output Fields

[Table 30 on page 241](#) lists the output fields for the **show chassis cluster status** command. Output fields are listed in the approximate order in which they appear.



Table 30: show chassis cluster status Output Fields

Field Name	Field Description
Cluster ID	ID number (1-15) of a cluster is applicable for releases upto Junos OS Release 12.1X45-D10. ID number (1-255) is applicable for Releases 12.1X45-D10 and later. Setting a cluster ID to 0 is equivalent to disabling a cluster.
Redundancy-Group	You can create up to 128 redundancy groups in the chassis cluster.
Node name	Node (device) in the chassis cluster ( <b>node0</b> or <b>node1</b> ).
Priority	Assigned priority for the redundancy group on that node.
Status	<p>State of the redundancy group (<b>Primary</b>, <b>Secondary</b>, <b>Lost</b>, or <b>Unavailable</b>).</p> <ul style="list-style-type: none"> <li>• <b>Primary</b>—Redundancy group is active and passing traffic.</li> <li>• <b>Secondary</b>—Redundancy group is passive and not passing traffic.</li> <li>• <b>Lost</b>—Node loses contact with the other node through the control link. Most likely to occur when both nodes are in a cluster and there is a control link failure, one node cannot exchange heartbeats, or when the other node is rebooted.</li> <li>• <b>Unavailable</b>—Node has not received a single heartbeat over the control link from the other node since the other node booted up. Most likely to occur when one node boots up before the other node, or if only one node is present in the cluster.</li> </ul>
Preempt	<ul style="list-style-type: none"> <li>• <b>Yes</b>: Primary state can be preempted based on priority.</li> <li>• <b>No</b>: Change in priority will not preempt the primary state.</li> </ul>
Manual failover	<ul style="list-style-type: none"> <li>• <b>Yes</b>: Primary state is set manually through the CLI with the <b>request chassis cluster failover node</b> or <b>request chassis cluster failover redundancy-group</b> command. This overrides <b>Priority</b> and <b>Preempt</b>.</li> <li>• <b>No</b>: Primary state is not set manually through the CLI.</li> </ul>
Monitor-failures	<ul style="list-style-type: none"> <li>• <b>None</b>: Cluster working properly.</li> <li>• <b>Monitor Failure code</b>: Cluster is not working properly and the respective failure code is displayed.</li> </ul>

## Sample Output

```
show chassis cluster status
```

```
user@host> show chassis cluster status
```

Monitor Failure codes:

CS	Cold Sync monitoring	FL	Fabric Connection monitoring
GR	GRES monitoring	HW	Hardware monitoring
IF	Interface monitoring	IP	IP monitoring
LB	Loopback monitoring	MB	Mbuf monitoring
NH	Nexthop monitoring	NP	NPC monitoring
SP	SPU monitoring	SM	Schedule monitoring
CF	Config Sync monitoring		

Cluster ID: 1

Node	Priority	Status	Preempt	Manual	Monitor-failures
------	----------	--------	---------	--------	------------------

Redundancy group: 0 , Failover count: 1

node0	200	primary	no	no	None
node1	1	secondary	no	no	None

Redundancy group: 1 , Failover count: 1

node0	101	primary	no	no	None
node1	1	secondary	no	no	None

## Sample Output

**show chassis cluster status with preemptive delay**

user@host> **show chassis cluster status**

Cluster ID: 1

Node	Priority	Status	Preempt	Manual	Monitor-failures
------	----------	--------	---------	--------	------------------

Redundancy group: 0, Failover count: 1

node0	200	primary	no	no	None
node1	100	secondary	no	no	None

Redundancy group: 1, Failover count: 3

<b>node0</b>	<b>200</b>	<b>primary-preempt-hold</b>	<b>yes</b>	<b>no</b>	<b>None</b>	<b>node1</b>	<b>100</b>	<b>secondary</b>
		yes	no	None				

## Sample Output

**show chassis cluster status redundancy-group 1**

user@host> **show chassis cluster status redundancy-group 1**

```
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring

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

Redundancy group: 1 , Failover count: 1
node0  101      primary    no      no      None
node1  1        secondary  no      no      None
```

# show interfaces

## Syntax

```
show interfaces extensive interface-name
show interfaces interface-name
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

## Description

Display details of the interfaces configured on NFX150.

## Required Privilege Level

view

## Sample Output

### show interfaces extensive

```
user@host> show interfaces extensive heth-0-1
```

```
Physical interface: heth-0-1, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Copper, MTU: 9192, Speed: 1Gbps, Duplex:
Full-duplex, Auto-negotiation: Enabled
  Device flags      : Present Running
  Current address: e8:b6:c2:cb:b8:89, Hardware address: e8:b6:c2:cb:b8:89
  Traffic statistics:
    Input  bytes   :                0                N/A bps
    Output bytes   :                0                N/A bps
    Input  packets :                0                N/A pps
    Output packets :                0                N/A pps
  MAC statistics:
    Receive      Transmit
    Total octets      0                0
    Total packets     0                0
    Unicast packets   0                0
    Broadcast packets 0                0
    Multicast packets 0                0
    CRC/Align errors  0                N/A
    FIFO errors       0                0
    DMA errors        0                0
    MAC control frames 0                0
```

```

MAC pause frames          0          0
Oversized frames          0
VF statistics:
  VF Number: 0, PCI Address: 0000:06:10:0, Mapped to: ge-1/0/1
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
  VF Number: 1, PCI Address: 0000:06:10:4, Mapped to: ge-1/0/2
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
  VF Number: 2, PCI Address: 0000:06:11:0, Mapped to: None
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
  VF Number: 3, PCI Address: 0000:06:11:4, Mapped to: None
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0

```

### show interfaces <interface1>

user@host> show interfaces ge-1/0/1

```

Physical interface: ge-1/0/1, Enabled, Physical link is Up
  Interface index: 184, SNMP ifIndex: 551
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online
Device flags      : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
Link flags       : None
CoS queues       : 8 supported, 8 maximum usable queues
Current address: e8:b6:c2:cb:b8:5c, Hardware address: e8:b6:c2:cb:b8:5c
Last flapped    : 2019-09-12 15:43:49 UTC (00:00:03 ago)
Input rate      : 0 bps (0 pps)
Output rate     : 0 bps (0 pps)
Active alarms   : None
Active defects   : None
PCS statistics           Seconds
  Bit errors             0
  Errored blocks         0
Ethernet FEC statistics   Errors
  FEC Corrected Errors   0
  FEC Uncorrected Errors 0
  FEC Corrected Errors Rate 0

```

```

    FEC Uncorrected Errors Rate          0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled

```

### show interfaces <interface1>

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

```

Physical interface: ge-1/0/2, Enabled, Physical link is Up
  Interface index: 185, SNMP ifIndex: 552
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
  Speed: 1000Mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online
  Device flags      : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags        : None
  CoS queues        : 8 supported, 8 maximum usable queues
  Current address: e8:b6:c2:cb:b8:5d, Hardware address: e8:b6:c2:cb:b8:5d
  Last flapped      : 2019-09-12 15:43:49 UTC (00:00:07 ago)
  Input rate        : 0 bps (0 pps)
  Output rate       : 0 bps (0 pps)
  Active alarms     : None
  Active defects    : None
  PCS statistics
    Bit errors              Seconds
    Errored blocks          0
  Ethernet FEC statistics
    FEC Corrected Errors    0
    FEC Uncorrected Errors  0
    FEC Corrected Errors Rate 0
    FEC Uncorrected Errors Rate 0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

```

# show system visibility cpu

## Syntax

```
show system visibility cpu
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
 Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

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

## Required Privilege Level

view

## RELATED DOCUMENTATION

<a href="#">show system visibility host   251</a>
<a href="#">show system visibility memory   262</a>
<a href="#">show system visibility network   265</a>
<a href="#">show system visibility vnf   271</a>

## List of Sample Output

- [show system visibility cpu \(NFX150\) on page 248](#)
- [show system visibility cpu \(NFX250 \(NG\)\) on page 249](#)

## Output Fields

[Table 31 on page 247](#) lists the output fields for the **show system visibility cpu** command. Output fields are listed in the approximate order in which they appear.

Table 31: show system visibility cpu Output Fields

Field Name	Field Description
<b>Fields for CPU Statistics</b>	
CPU ID	The CPU ID
User Time	The amount of user time, in seconds.
System Time	The amount of system time, in seconds.

Table 31: show system visibility cpu Output Fields (*continued*)

Field Name	Field Description
Idle Time	The amount of time spent in idle mode, in seconds.
Nice Time	The amount of spent nice time, in seconds.
I/O Wait Time	The amount of time spent waiting for input/output (I/O) operations, in seconds.
Interrupt Service Time	The amount of interrupt service time, in seconds.
Service Time	The amount of service time, in seconds.
<b>Fields for CPU Usages</b>	
CPU ID	The CPU ID
CPU Usage	The percentage of CPU used.
<b>Fields for CPU Pinning Information</b>	
Virtual Machine	The name of the virtual machine.
vCPU	The ID of virtual CPUs used by the virtual machine.
CPU	The ID of CPUs used by the virtual machine.
System Component	The name of the system component.
CPUs	The ID of CPUs used by the system component.

## Sample Output

**show system visibility cpu (NFX150)**

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

```

CPU Statistics (Time in sec)
-----
CPU Id User Time System Time Idle Time Nice Time IOWait Time Intr. Service Time
-----
0      26583    40107      105816    0         102         0

```



1	53183	64078	56959	0	0	0
2	72	67	171189	0	1	0
3	0	96	171241	0	0	0

#### CPU Usages

-----

CPU Id CPU Usage

-----

0	36.399999999999999
1	66.700000000000003
2	0.0
3	0.0

#### CPU Pinning Information

-----

Virtual Machine	vCPU	CPU
-----------------	------	-----

-----

vjunos0	0	0
---------	---	---

System Component	CPUs
------------------	------

-----

ovs-vswitchd	1
--------------	---

### show system visibility cpu (NFX250 (NG))

user@host> show system visibility cpu

#### CPU Statistics (Time in sec)

-----

CPU Id	User Time	System Time	Idle Time	Nice Time	IOWait Time	Intr. Service Time
--------	-----------	-------------	-----------	-----------	-------------	--------------------

-----

0	28568	4549	236916	0	205	0
1	272502	0	48	0	0	0
2	165	45	272268	0	11	0
3	40	9	272470	0	0	0
4	0	0	272494	0	0	0
5	0	0	272550	0	0	0
6	0	0	272552	0	0	0
7	272507	0	47	0	0	0
8	0	0	272552	0	0	0
9	0	0	272553	0	0	0
10	0	0	272553	0	0	0
11	0	0	272547	0	0	0

CPU Usages	
-----	
CPU Id	CPU Usage
-----	-----
0	11.9
1	100.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	100.0
8	0.0
9	0.0
10	0.0
11	0.0

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

System Component	CPUs
-----	-----
ovs-vswitchd	0, 1, 7

# show system visibility host

## Syntax

```
show system visibility host
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Displays details such as the host uptime, number of tasks, CPU statistics, list of disk partitions, disk usage, disk I/O statistics, list of network interfaces, and per port statistics for a Junos OS platform.

## Required Privilege Level

view

## RELATED DOCUMENTATION

- [show system visibility cpu | 247](#)
- [show system visibility memory | 262](#)
- [show system visibility network | 265](#)
- [show system visibility vnf | 271](#)

## List of Sample Output

- [show system visibility host \(NFX150\) on page 254](#)
- [show system visibility host \(NFX250 \(NG\)\) on page 258](#)

## Output Fields

[Table 32 on page 251](#) lists the output fields for the **show system visibility host** command. Output fields are listed in the approximate order in which they appear.

Table 32: show system visibility host Output Fields

Field Name	Field Description
Field for Host Uptime	
Uptime	The time the host has been operational.
Fields for Host Tasks	

Table 32: show system visibility host Output Fields (*continued*)

Field Name	Field Description
<b>Total</b>	The total number of tasks.
<b>Running</b>	The total number of tasks running.
<b>Sleeping</b>	The total number of tasks in sleeping state.
<b>Stopped</b>	The total number of tasks that are stopped.
<b>Zombie</b>	The total number of zombie processes.
<b>Fields for Host CPU Information</b>	
<b>User Time</b>	The amount of user time, in seconds.
<b>System Time</b>	The amount of system time, in seconds.
<b>Idle Time</b>	The amount of time spent in idle mode, in seconds.
<b>Nice Time</b>	The amount of spent nice time, in seconds.
<b>I/O Wait Time</b>	The amount of time spent waiting for input/output (I/O) operations, in seconds.
<b>Interrupt Service Time</b>	The amount of interrupt service time, in seconds.
<b>Fields for Host Disk Partitions</b>	
<b>Device</b>	The device path.
<b>Mount Point</b>	The mount point of the device path.
<b>File System</b>	The file system type.
<b>Options</b>	Options available for the device path.
<b>Fields for Host Disk Usage Information</b>	
<b>Total</b>	The total amount of disk usage space, in mebibytes (MiB).
<b>Used</b>	The amount of used disk usage space, in mebibytes (MiB).
<b>Free</b>	The amount of free disk usage space, in mebibytes (MiB).

Table 32: show system visibility host Output Fields (*continued*)

Field Name	Field Description
Percentage Used	The percentage of used disk space.
<b>Fields for Host Disk I/O Information</b>	
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.
<b>Fields for List of Host Interfaces</b>	
Interfaces	The name of the interface.
State	The state of the Host Interface.
MAC	The MAC address of the interface.
<b>Fields for List of Host Port Statistics</b>	
Interface	The name of the interface.
Bytes Sent	The number of bytes sent.
Bytes Received	The number of bytes received.
Packets Sent	The number of packets sent.
Packets Received	The number of packets received.
Errors In	The number of errors in.
Errors Out	The number of errors out.
Drops In	The number of drops in.

Table 32: show system visibility host Output Fields (continued)

Field Name	Field Description
Drops Out	The number of drops out.

# Sample Output

show system visibility host (NFX150)

user@host> show system visibility host

```

Host Uptime
-----
Uptime: 1 day 23:19:41.21000

Host Tasks
-----
Total:      187
Running:    3
Sleeping:   179
Stopped:    0
Zombie:     5

Host CPU Information (Time in sec)
-----
User Time:      79359
System Time:    0
Idle Time:      502215
I/O Wait Time:  103
Nice Time:      103724
Interrupt Service Time: 0

Host Disk Partitions
-----
Device                                Mount Point      File System  Options
-----
/dev/sda2                             /                 ext4
rw,relatime,i_version,data=ordered
/dev/sda1                             /boot/efi        vfat
rw,noatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro
/dev/sda7                             /config          ext4

```

```
rw,noatime,data=ordered
/dev/sda8                /var/log                ext4
rw,noatime,data=ordered
/dev/sda9                /mnt/.share             ext4
rw,noatime,discard,data=ordered
/dev/sda5                /junos                  ext4
rw,noatime,discard,data=ordered
/dev/loop0               /var/tmp                ext4
rw,relatime,data=ordered
/dev/loop1               /mnt/.share/lshare/jnpr/jlog ext4
rw,relatime,data=ordered
/dev/loop0               /mnt/.share/lshare/jnpr/jtmp ext4
rw,relatime,data=ordered
```

Host Disk Usage Information

```
-----
Total (MiB):      1469
Used  (MiB):      948
Free  (MiB):      429
Percentage Used:  64.5
```

Host Disk I/O Information

```
-----
Read Count: 187083
Write Count: 256206
Read Bytes: 2290787328
Write Bytes: 3331667456
Read Time: 33977
Write Time: 258864
```

Host Interfaces

Interface	State	MAC
-----	-----	-----
heth-0-1	active	00:00:5e:00:53:8e
heth-0-0	active	00:00:5e:00:53:8d
heth-0-3	active	00:00:5e:00:53:90
heth-0-2	active	00:00:5e:00:53:8f
heth-0-5	inactive	00:00:5e:00:53:92
heth-0-4	inactive	00:00:5e:00:53:91
ctrlbr0	active	00:00:5e:00:53:10
docker0	inactive	00:00:5e:00:53:8c
eth0br	active	00:00:5e:00:53:00
eth1br	inactive	00:00:5e:00:53:67

```

13_h_ge_1_0_0      active  00:00:5e:00:53:6d
13_h_ltectrl        active  00:00:5e:00:53:f1
13_h_ltedata        active  00:00:5e:00:53:91
lo                  inactive 00:00:00:00:00:00
lte_crt10           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
-----
-----
13_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_crt10	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					

-----

-----

13_h_ge_1_0_0	0	648	0	8	0	0
0	0					
ovs-sys-br	0	0	0	0	0	0
0	0					
vnet0	2573491477	117345734	2448205	1790887	0	0
0	0					
vnet1	670930985	585788796	7585078	8400542	0	0
0	0					
vnet2	454043208	224389433	2873376	416585	0	0
0	0					
vnet3	7129616	9814	137213	231	0	0
0	0					
eth0	224389433	464747548	416585	2889060	0	0
9829	0					
lo	61305	61305	920	920	0	0
0	0					
virbr1	2475291351	90762062	1008399	1774468	0	0
0	0					
irb	0	0	0	0	0	0
0	0					
hsxe1	0	0	0	0	0	0
0	0					
hsxe0	0	0	0	0	0	0
0	0					
docker0	0	0	0	0	0	0
0	0					
dcapi-tap	0	0	0	0	0	0
0	0					
sit0	0	0	0	0	0	0
0	0					
flowd_h_mgmt	387545386	426690199	5662328	5294853	0	0
0	0					
virbr0-nic	0	0	0	0	0	0
0	0					
virbr0	3021873	1067179	4573	6153	0	0
0	0					
jdm-hbme1	1785562	33378	34145	404	0	0
0	0					
jdm-hbme2	41904	72344	321	323	0	0
0	0					
eth0br	0	401858893	0	2755416	0	0
226	0					
ctrlbr0	243770080	159923150	2283092	2738720	0	0
0	0					

eth1br	0	0	0	0	0	0
0	0					
ovs-netdev	0	0	0	0	0	0
0	0					

# show system visibility memory

Syntax

```
show system visibility memory
```

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

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

Required Privilege Level

view

RELATED DOCUMENTATION

- [show system visibility cpu | 247](#)
- [show system visibility host | 251](#)
- [show system visibility network | 265](#)
- [show system visibility vnf | 271](#)

List of Sample Output

- [show system visibility memory \(NFX150\) on page 263](#)
- [show system visibility memory \(NFX250 \(NG\)\) on page 263](#)

Output Fields

Table 33 on page 262 lists the output fields for the **show system visibility memory** command. Output fields are listed in the approximate order in which they appear.

Table 33: show system visibility memory Output Fields

Field Name	Field Description
Fields for Memory Information—Virtual Memory	
Total	The total amount of available virtual memory, in kibibytes (KiBs).
Used	The total amount of used virtual memory, in kibibytes (KiBs).
Available	The total amount of available virtual memory, in kibibytes (KiBs).

Table 33: show system visibility memory Output Fields (*continued*)

Field Name	Field Description
<b>Free</b>	The total amount of free virtual memory, in kibibytes (KiBs).
<b>Percent Used</b>	The percentage of buffer virtual memory used.
<b>Fields for Memory Information—Swap Memory</b>	
<b>Total</b>	The total amount of available swap memory, in kibibytes (KiBs).
<b>Used</b>	The total amount of used swap memory, in kibibytes (KiBs).
<b>Free</b>	The total amount of free swap memory, in kibibytes (KiBs).
<b>Percent Used</b>	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 400	other process	1
ovs-vswitchd 0	other process	2



# show system visibility network

## Syntax

```
show system visibility network
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.  
Command introduced in Junos OS Release 19.4R1 for NFX350 devices.

## Description

Displays details such as the list of MAC addresses assigned to VNF interfaces, the list of internal IP addresses for VNFs, the list of virtual functions used by VNFs, and the list of VNF interfaces for a Junos OS platform.

## Required Privilege Level

view

## RELATED DOCUMENTATION

- [show system visibility cpu | 247](#)
- [show system visibility host | 251](#)
- [show system visibility memory | 262](#)
- [show system visibility vnf | 271](#)

## List of Sample Output

- [show system visibility network \(NFX150\) on page 266](#)
- [show system visibility network \(NFX250 \(NG\)\) on page 269](#)

## Output Fields

Table 34 on page 265 lists the output fields for the **show system visibility network** command. Output fields are listed in the approximate order in which they appear.

Table 34: show system visibility network Output Fields

Field Name	Field Description
Fields for List of VNF MAC Addresses	
VNF	The name of the VNF.
MAC	The MAC address of the VNF.

Table 34: show system visibility network Output Fields (*continued*)

Field Name	Field Description
<b>Fields for List of VNF Internal IP Addresses</b>	
VNF	The name of the VNF.
IP	The IP address of the VNF.
<b>Fields for List of VNF Virtual Functions</b>	
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.
<b>Fields for List of Free Virtual Functions</b>	
PF	The names of the Physical Functions available.
VF	The names of the Virtual Functions available for each Physical Function.
<b>Fields for List of VNF Interfaces</b>	
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
13_ge_1_0_4_vfdef0	heth-0-1	0000:04:10:0
12_ge_0_0_0_vfdef0	heth-0-0	0000:04:10:1
12_ge_0_0_0_vfdef1	heth-0-0	0000:04:10:5
12_ge_0_0_0_vfdef2	heth-0-0	0000:04:11:1
12_ge_0_0_0_vfdef3	heth-0-0	0000:04:11:5
13_ge_1_0_2_vfdef0	heth-0-5	0000:07:10:0
12_ge_0_0_2_vfdef0	heth-0-2	0000:04:10:3
12_ge_0_0_2_vfdef1	heth-0-2	0000:04:10:7
12_ge_0_0_2_vfdef2	heth-0-2	0000:04:11:3
12_ge_0_0_2_vfdef3	heth-0-2	0000:04:11:7
13_ge_1_0_1_vfdef0	heth-0-4	0000:07:10:1
12_ge_0_0_3_vfdef0	heth-0-3	0000:04:10:2
12_ge_0_0_3_vfdef1	heth-0-3	0000:04:10:6
12_ge_0_0_3_vfdef2	heth-0-3	0000:04:11:2
12_ge_0_0_3_vfdef3	heth-0-3	0000:04:11:6

## Free Virtual Functions

PF	VF
----	----

```

heth-0-1 0000:04:10:4
heth-0-1 0000:04:11:0
heth-0-1 0000:04:11:4
heth-0-5 0000:07:10:2
heth-0-5 0000:07:10:4
heth-0-5 0000:07:10:6
heth-0-4 0000:07:10:3
heth-0-4 0000:07:10:5
heth-0-4 0000:07:10:7

```

#### VNF Interfaces

VNF VLAN-ID	Interface Type	Source	Model	MAC
centos2	centos2_vnet6	network default	virtio	00:00:5e:00:53:a2
--				
centos2	centos2_vnet7	bridge eth0br	virtio	00:00:5e:00:53:a3
--				
centos2	centos2_eth2	bridge ovs-sys-br	virtio	00:00:5e:00:53:a4
199				
centos2	centos2_eth3	bridge custom1	virtio	00:00:5e:00:53:a5
--				
centos1	centos1_vnet4	network default	virtio	00:00:5e:00:53:9e
--				
centos1	centos1_vnet5	bridge eth0br	virtio	00:00:5e:00:53:9f
--				
centos1	centos1_eth2	bridge ovs-sys-br	virtio	00:00:5e:00:53:a0
100				
centos1	centos1_eth3	bridge custom1	virtio	00:00:5e:00:53:a1
--				

#### OVS Interfaces

NAME	MTU
custom1	1500
centos2_eth3	1500
centos1_eth3	1500
veth11	9200
l3_h_ge_1_0_0	9200
veth01	9200
ovs-sys-br	1500

```
centos1_eth2      1500
centos2_eth2      1500
```

**show system visibility network (NFX250 (NG))**

user@host> **show system visibility network**

```
VNF Virtual Functions
-----
VNF                                PF      VF
-----
System_vfdef0                     hsxe0   0000:03:13:6
System_vfdef0                     hsxe1   0000:03:13:7

Free Virtual Functions
-----
PF      VF
-----
hsxe0   0000:03:10:0
hsxe0   0000:03:10:2
hsxe0   0000:03:10:4
hsxe0   0000:03:10:6
hsxe0   0000:03:11:0
hsxe0   0000:03:11:2
hsxe0   0000:03:11:4
hsxe0   0000:03:11:6
hsxe0   0000:03:12:0
hsxe0   0000:03:12:2
hsxe0   0000:03:12:4
hsxe0   0000:03:12:6
hsxe0   0000:03:13:0
hsxe0   0000:03:13:2
hsxe0   0000:03:13:4
hsxe1   0000:03:10:1
hsxe1   0000:03:10:3
hsxe1   0000:03:10:5
hsxe1   0000:03:10:7
hsxe1   0000:03:11:1
hsxe1   0000:03:11:3
hsxe1   0000:03:11:5
hsxe1   0000:03:11:7
hsxe1   0000:03:12:1
hsxe1   0000:03:12:3
hsxe1   0000:03:12:5
```

hsxe1	0000:03:12:7
hsxe1	0000:03:13:1
hsxe1	0000:03:13:3
hsxe1	0000:03:13:5
OVS Interfaces	
-----	
NAME	MTU
-----	-----
dpdk1	1500
ovs-sys-br	1500
13_h_ge_1_0_0	1500
dpdk0	1500

# show system visibility vnf

## Syntax

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

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

If a VNF name is not specified, this command displays the details of all VNFs present in the system. Details include VNF memory usage, CPU statistics, the list of network interfaces, the list of disk files, per disk usage, per port I/O statistics, and media information, which includes details about CD-ROM and USB storage devices.

If a VNF name is specified, this command displays the details of that particular VNF. Details include VNF memory usage, CPU statistics, the list of network interfaces, the list of disk files, per disk usage, per port I/O statistics, and media information, which includes details about CD-ROM and USB storage devices.

## Required Privilege Level

view

## RELATED DOCUMENTATION

<a href="#">show system visibility cpu</a>	<a href="#">  247</a>
<a href="#">show system visibility host</a>	<a href="#">  251</a>
<a href="#">show system visibility memory</a>	<a href="#">  262</a>
<a href="#">show system visibility network</a>	<a href="#">  265</a>

## List of Sample Output

[show system visibility vnf on page 274](#)

## Output Fields

[Table 35 on page 272](#) lists the output fields for the **show system visibility vnf** command. Output fields are listed in the approximate order in which they appear.

Table 35: show system visibility vnf Output Fields

Field Name	Field Description
<b>Fields for List of VNFs</b>	
<b>ID</b>	ID of the VNF.
<b>Name</b>	Name of the VNF.
<b>State</b>	State of the VNF.
<b>Fields for VNF Memory Usage</b>	
<b>Name</b>	Name of the VNF.
<b>Maximum Memory</b>	The maximum amount of memory, in kibibytes (KiBs).
<b>Used Memory</b>	The total amount of used memory, in kibibytes (KiBs).
<b>Used 1G Hugepages</b>	The total number of 1G hugepages used.
<b>Used 2M Hugepages</b>	The total number of 2M hugepages used.
<b>Fields for VNF CPU Stats</b>	
<b>Name</b>	Name of the VNF.
<b>CPU Time</b>	The total CPU time, in seconds.
<b>System Time</b>	The amount of system CPU time, in seconds.
<b>User Time</b>	The amount of user CPU time, in seconds.
<b>Fields for List of VNF MAC Addresses</b>	
<b>VNF</b>	Names of the VNFs.
<b>MAC</b>	MAC addresses of the VNFs.
<b>Fields for List of VNF Internal IP Addresses</b>	
<b>VNF</b>	Names of the VNFs.
<b>IP</b>	Internal IP addresses of the VNFs.



Table 35: show system visibility vnf Output Fields (*continued*)

Field Name	Field Description
<b>Fields for List of Virtual Functions per VNF</b>	
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.
<b>Fields for the VNF Interfaces</b>	
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.
<b>Fields for List of VNF Disk Information</b>	
VNF	The name of the VNF.
Disk	The name of the disk.
File	The path to the disk.
<b>Fields for List of VNF Disk Usage</b>	
VNF	The name of the VNF.
Disk	The name of the disk.
Read Requests	The number of times a read operation has happened on the disk.
Bytes Read	The number of read bytes on the disk.
Write Requests	The number of times a write operation has happened on the disk.

Table 35: show system visibility vnf Output Fields (*continued*)

Field Name	Field Description
Bytes Written	The number of bytes written on the disk.
<b>Fields for List of VNF Port Statistics</b>	
VNF	The name of the VNF.
Port	The name of the port.
Rcvd Bytes	The number of bytes received.
Rcvd Packets	The number of packets received.
Rcvd Error	The number of errors received.
Rcvd Drop	The number of drops received.
Trxd Bytes	The number of bytes transferred.
Trxd Packets	The number of packets transferred.
Trxd Error	The number of errors transferred.
Trxd Drop	The number of drops transferred.

## Sample Output

**show system visibility vnf**

user@host> **show system visibility vnf**

List of VNFs

```

-----
ID   Name                               State
-----
5    centos                             Running

```

VNF Memory Usage

```

-----
Name                               Maximum Memory (KiB)  Used Memory (KiB)

```

Used 1G Hugepages    Used 2M Hugepages

```
-----
centos                                2097152                260741                0
                                0
```

VNF CPU Statistics (Time in ms)

```
-----
Name                                CPU Time                System Time   User Time
-----
centos                                14029                   3650          1540
```

VNF MAC Addresses

```
-----
VNF                                MAC
-----
centos_ethdef0                     E8:B6:C2:CC:66:9B
centos_ethdef1                     E8:B6:C2:CC:66:9C
```

VNF Internal IP Addresses

```
-----
VNF                                IP
-----
centos                             192.0.2.100
```

VNF Virtual Functions

```
-----
VNF                                PF                VF
-----
12_ge_0_0_0_vfdef0                 heth-0-0          0000:02:10:1
12_ge_0_0_0_vfdef1                 heth-0-0          0000:02:10:5
12_ge_0_0_0_vfdef2                 heth-0-0          0000:02:11:1
12_ge_0_0_0_vfdef3                 heth-0-0          0000:02:11:5
12_ge_0_0_2_vfdef0                 heth-0-2          0000:02:10:3
12_ge_0_0_2_vfdef1                 heth-0-2          0000:02:10:7
12_ge_0_0_2_vfdef2                 heth-0-2          0000:02:11:3
12_ge_0_0_2_vfdef3                 heth-0-2          0000:02:11:7
13_ge_1_0_2_vfdef0                 heth-0-5          0000:05:10:0
12_ge_0_0_1_vfdef0                 heth-0-1          0000:02:10:0
12_ge_0_0_1_vfdef1                 heth-0-1          0000:02:10:4
12_ge_0_0_1_vfdef2                 heth-0-1          0000:02:11:0
12_ge_0_0_1_vfdef3                 heth-0-1          0000:02:11:4
12_ge_0_0_3_vfdef0                 heth-0-4          0000:05:10:1
12_ge_0_0_3_vfdef1                 heth-0-4          0000:05:10:3
12_ge_0_0_3_vfdef2                 heth-0-4          0000:05:10:5
```

```
12_ge_0_0_3_vfdef3          heth-0-4  0000:05:10:7
```

```
13_ge_1_0_1_vfdef0          heth-0-3  0000:02:10:2
```

#### VNF Interfaces

VNF	Interface	Type	Source	Model	MAC
-----					
-----					
-----					
centos	centos_vnet4	network	default	virtio	e8:b6:c2:cc:66:9b
--					
centos	centos_vnet5	bridge	eth0br	virtio	
e8:b6:c2:cc:66:9c	--				

#### VNF Disk Information

VNF	Disk	File
-----		
-----		
centos	vda	/var/public/centos-linux-1.img
centos	hda	/var/public/vnf_config_data_vnf0

#### VNF Disk Usage

VNF	Disk	Read Req	Read Bytes	Write Req	Write Bytes
-----					
centos	vda	5382	84654592	2068	4372480
centos	hda	15	37068	0	0

#### VNF Port Statistics

VNF	Port	Rcvd Bytes	Rcvd Packets	Rcvd Error	Rcvd Drop	Trxd
Bytes	Trxd Packets	Trxd Error	Trxd Drop			
-----						
centos	centos_vnet4	572	11	0	0	850
7	0	0				
centos	centos_vnet5	21729	258	0	395	0
0	0	0				

#### VNF Media Information

VNF	Media	Disk	File
-----			
-----			

vnf0	CDROM hda	/var/public/vnf_config_data_vnf0
------	-----------	----------------------------------

# show vmhost connections

## Syntax

```
show vmhost connections
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Display the details for the cross-connect connections. The NFX150 and NFX250 (NG) supports VLAN PUSH, POP, and SWAP operations.

## Options

- name**—Display the details of a specific connection.
- down**—Display the details of connections that are not operational.
- up**—Display the details of connections that are operational.
- up-down**—Display the details of both operational and non-operational connections.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost connections on page 279](#)

## Output Fields

[Table 36 on page 278](#) lists the output fields for the **show vmhost connections** command. Output fields are listed in the approximate order in which they appear.

Table 36: 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.

Table 36: show vmhost connections Output Fields (*continued*)

Field Name	Field Description
Interface	Specifies an interface on which the connection is established.
Status	Displays the status of the connection.

## Sample Output

**show vmhost connections**

user@host> **show vmhost connections**

Connection	Function	Interface	Vlan	Status
-----				
phy_cc	system	sxe0	200	up
	centos1	eth2	500	
push_pop_cc	centos1	eth2	none	down
	centos2	eth3	none	
swap_cc	centos1	eth2	300	up
	centos2	eth2	400	
vlan_cc	centos1	eth2	100	up
	centos2	eth2	100	

# show vmhost control-plane

## Syntax

```
show vmhost control-plane
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
 Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

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

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost control-plane on page 280](#)

## Sample Output

**show vmhost control-plane**

user@host> **show vmhost control-plane**

Vmhost Control Plane Information		
Name	State	Status
Junos Control Plane	RUNNING	OK
Juniper Device Manager	RUNNING	OK
Layer 2 Infrastructure	RUNNING	OK
Layer 3 Infrastructure	RUNNING	OK
LTE	RUNNING	OK



# show vmhost crash

## Syntax

```
show vmhost crash
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Display host OS crash information.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost crash on page 281](#)

## Sample Output

**show vmhost crash**

user@host> **show vmhost crash**

```
-rw-r--r-- 1 root root 306773 Mar 22 10:41
local-node.srxpfe.7439.1521715280.core.tgz
-rw-r--r-- 1 root root 307058 Mar 22 10:42
local-node.srxpfe.8184.1521715324.core.tgz
-rw-r--r-- 1 root root 306999 Mar 22 10:42
local-node.srxpfe.8918.1521715357.core.tgz
-rw-r--r-- 1 root root 315121 Apr 18 05:35
localhost.dummy_flowdapp.3037.1524029709.core.tgz
-rw-r--r-- 1 root root 315033 Apr 18 05:17
localhost.dummy_flowdapp.3432.1524028674.core.tgz
```

```
-rw-r--r-- 1 root root 315088 Apr 13 18:11  
localhost.dummy_flowdapp.3435.1523643106.core.tgz
```

# show vmhost forwarding-options analyzer

Syntax

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

Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

Description

Displays information about the VNF analyzers that are configured for port mirroring on a Junos OS platform.

Options

*analyzer-name*—Displays the details of a specific analyzer on the device.

Required Privilege Level

view

RELATED DOCUMENTATION

| [vmhost](#)

List of Sample Output

[show vmhost forwarding-options analyzer on page 284](#)

Output Fields

[Table 37 on page 283](#) 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 37: show vmhost forwarding-options analyzer Output Fields

Field Name	Field Description
Analyzer name	Displays the name of the analyzer instance.
Egress monitored interfaces	Displays interfaces for which the traffic leaving the interfaces is mirrored.
Output interface	Specifies an interface to which mirrored packets are sent.
Ingress monitored interfaces	Displays interfaces for which the traffic entering the interfaces is mirrored.

## Sample Output

**show vmhost forwarding-options analyzer**

user@host> **show vmhost forwarding-options analyzer**

```
Analyzer name           : mon1
Egress monitored interfaces : vnf-name1:eth2
Output interface        : analyzer1:eth2

Analyzer name           : mon2
Ingress monitored interfaces : vnf-name2:eth2
Output interface        : analyzer1:eth3
```

# show vmhost memory

## Syntax

```
show vmhost memory
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Display the memory information for the host OS.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost memory on page 285](#)

## Output Fields

## Sample Output

```
show vmhost memory
```

```
user@host> show vmhost memory
```

```
Memory Controller Information
-----

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

# show vmhost mode

## Syntax

```
show vmhost mode
```

## Release Information

Command introduced in Junos OS Release 19.1R1 for NFX150 and NFX250 (NG) devices.

## Description

Display the CPU and memory allocations for various components.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

- [show vmhost mode \(Throughput mode\) on page 286](#)
- [show vmhost mode \(Hybrid mode\) on page 287](#)
- [show vmhost mode \(Compute mode\) on page 289](#)

## Sample Output

### show vmhost mode (Throughput mode)

user@host> show vmhost mode

```
Mode:
-----
Current Mode: throughput

CPU Allocations:
Name                               Configured          Used
-----
Junos Control Plane                0                    0
Juniper Device Manager              0                    0
```

LTE	0	-
NFV Backplane Control Path	0	0
NFV Backplane Data Path	1,2	1,2
Layer 2 Control Path	0	0
Layer 2 Data Path	3,4	3,4
Layer 3 Control Path	0	0
Layer 3 Data Path	5,6,7	5,6,7
Memory Allocations:		
Name	Configured	Used
-----		
Junos Control Plane (mB)	2048	1548
NFV Backplane 1G hugepages	1	1
NFV Backplane 2M hugepages	-	0
Layer 2 1G hugepages	1	1
Layer 2 2M hugepages	-	0
Layer 3 1G hugepages	1	1
Layer 3 2M hugepages	651	650

## Sample Output

**show vmhost mode (Hybrid mode)**

user@host> **show vmhost mode**

Mode:
-----

Current Mode: hybrid

CPU Allocations:

Name	Configured	Used
Junos Control Plane	0	0
Juniper Device Manager	0	0
LTE	0	-
NFV Backplane Control Path	0	0
NFV Backplane Data Path	1,2	1,2
Layer 2 Control Path	0	0
Layer 2 Data Path	3	3
Layer 3 Control Path	0	0
Layer 3 Data Path	4,5	4,5

Memory Allocations:

Name	Configured	Used
Junos Control Plane (mB)	2048	1548
NFV Backplane 1G hugepages	1	1
NFV Backplane 2M hugepages	-	0
Layer 2 1G hugepages	1	1
Layer 2 2M hugepages	-	0
Layer 3 1G hugepages	1	1
Layer 3 2M hugepages	651	650



## Sample Output

show vmhost mode (Compute mode)

user@host> show vmhost mode

```

Mode:
-----
Current Mode: compute

CPU Allocations:

```

Name	Configured	Used
Junos Control Plane	16	16,6
Juniper Device Manager	16	16
LTE	16	-
NFV Backplane Control Path	16	16
NFV Backplane Data Path	1,2,3	1,2,3
Layer 2 Control Path	-	-
Layer 2 Data Path	-	-
Layer 3 Control Path	0	0
Layer 3 Data Path	4,5	4,5

```

CPUs available for VNFs
6,7,8,9,10,11,12,13,14,15,22,23,24,25,26,27,28,29,30,31 -

CPUs turned off
17,18,19,20,21

Memory Allocations:

```

Name	Configured	Used
Junos Control Plane (mB)	2048	2012
NFV Backplane 1G hugepages	12	18
NFV Backplane 2M hugepages	-	0
Layer 2 1G hugepages	-	-
Layer 2 2M hugepages	-	-
Layer 3 1G hugepages	6	6
Layer 3 2M hugepages	20481	20481

CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice
%idle									
Load Avg : 4.04 90.90	0.00		4.74	0.01	0.00	0.01	0.00	0.30	0.00
cpu0 : 8.26 73.23	0.00		15.91	0.06	0.00	0.06	0.00	2.47	0.00

cpu1	:	24.73	0.00	22.95	0.00	0.00	0.00	0.00	0.00	0.00
52.32										
cpu2	:	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00
99.97										
cpu3	:	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100.00										
cpu4	:	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
99.98										
cpu5	:	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100.00										
cpu6	:	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100.00										
cpu7	:	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100.00										
Device:	tps	kB_read/s		kB_wrtn/s		kB_read		kB_wrtn		
-----										
sda	2.15	7.60		30.04		4057951		16046703		

# show vmhost storage

## Syntax

```
show vmhost storage
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
 Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Display the vmhost storage information.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost storage on page 292](#)

## Sample Output

**show vmhost storage**

user@host> **show vmhost storage**

Vmhost Storage Information		
-----		
Storage Name :sda		
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	6562
12	Power_Cycle_Count	72

160	Uncorrectable_Sector_Count	0
161	Spare_Blocks	555
163	Number_of_Initial_Invalid_Blocks	31
164	Total_Erase_Count	72780
165	Maximum_Erase_Count	56
166	Minimum_Erase_Count	0
167	Average_Erase_Count	35
168	Maximum_Specified_Erase_Count	3000
169	Power-On_UECC_Count	56
192	Power-Off_Retract_Count	555
193	Dynamic_Remaps	0
194	Temperature_Celsius	37
195	Hardware_ECC_Recovered	646747
196	Reallocated_Event_Count	0
198	Offline_Uncorrectable	0
199	UDMA_CRC_Error_Count	0
215	TRIM_Count	80433
235	Total_Flash_LBAs_Written	103297788
237	Total_Flash_LBAs_Written_Expanded	0
241	Total_LBAs_Written	4262373185
242	Total_LBAs_Read	2322062690
243	Total_Host_LBAs_Written_Expanded	0
244	Total_Host_LBAs_Read_Expanded	0
248	SSD_Remaining_Life	99
249	Spare_Blocks_Remaining_Life	100

# show vmhost uptime

## Syntax

```
show vmhost uptime
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Display the current time and information such as how long the host OS has been running, number of users, average load, and the last reboot reason.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost uptime on page 294](#)

## 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 294](#).

# show vmhost version

## Syntax

```
show vmhost version
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

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

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost connections \(NFX150\) on page 296](#)

[show vmhost connections \(NFX250 \(NG\)\) on page 297](#)

## Sample Output

### show vmhost connections (NFX150)

```
user@host> show vmhost version
```

```
Partition set      : primary
Software version   : 18.2-20180402_18.2T_x_tvp.0
                   Host kernel release  : 4.1.27-rt30-WR8.0.0.23_ovp
                   Host kernel version  : #1 SMP Sat Mar 24 02:04:51 PDT 2018
```



## Sample Output

**show vmhost connections (NFX250 (NG))**

user@host> **show vmhost version**

```
Partition set : primary
Software version : 18.4R1.6
Host kernel release : 4.1.27-rt30-WR8.0.0.25_ovp
Host kernel version : #1 SMP Mon Nov 19 20:24:06 PST 2018
```

# show vmhost vlans

## Syntax

```
show vmhost vlans
```

## Release Information

Command introduced in Junos OS Release 18.1R1 for NFX150 Network Services Platform.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

## Description

Display details about the vmhost VLANs.

## Options

- vlan-name**—Display information for a specified VLAN.
- brief | detail | extensive** —Display the specified level of output.
- instance**—Display information for a specified instance.
- interface**—Name of interface for which the table is displayed.
- logical-system**—Name of logical system.

## Required Privilege Level

view

## RELATED DOCUMENTATION

| [vmhost](#)

## List of Sample Output

[show vmhost vlans on page 299](#)

## Output Fields

[Table 38 on page 298](#) 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 38: show vmhost vlans Output Fields

Field Name	Field Description
vlan-name	Display information for a specified VLAN

Table 38: show vmhost vlans Output Fields (*continued*)

Field Name	Field Description
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
-----
```