

## How to Configure the NFX250 (NextGen)



Modified: 2019-03-25

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 NFX250 (NextGen)*

Copyright © 2019 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 . . . . .	xi
	Documentation and Release Notes . . . . .	xi
	Using the Examples in This Manual . . . . .	xi
	Merging a Full Example . . . . .	xii
	Merging a Snippet . . . . .	xii
	Documentation Conventions . . . . .	xiii
	Documentation Feedback . . . . .	xv
	Requesting Technical Support . . . . .	xv
	Self-Help Online Tools and Resources . . . . .	xvi
	Creating a Service Request with JTAC . . . . .	xvi
<b>Chapter 1</b>	<b>Overview . . . . .</b>	<b>17</b>
	NFX250 NextGen Overview . . . . .	17
	Software Architecture . . . . .	18
	NFX250 Models . . . . .	19
	Interfaces . . . . .	20
	Performance Modes . . . . .	21
	Benefits and Uses . . . . .	21
	Junos OS Releases Supported on NFX Series Hardware . . . . .	21
	NFX Product Compatibility . . . . .	22
	Hardware Compatibility . . . . .	22
	Hardware Compatibility Tool . . . . .	22
	Software Version Compatibility . . . . .	22
	NFX250 Software Version Compatibility . . . . .	23
<b>Chapter 2</b>	<b>Initial Configuration . . . . .</b>	<b>25</b>
	Initial Configuration on NFX250 (NG) Devices . . . . .	25
	Factory Default Settings . . . . .	25
	Enabling Basic Connectivity . . . . .	26
	Establishing the Connection . . . . .	27
	Zero Touch Provisioning on NFX Series Devices . . . . .	27
	Understanding Zero Touch Provisioning . . . . .	27
	Provisioning an NFX Series Device . . . . .	28
	Provisioning an NFX Series Device Using Sky Enterprise . . . . .	29

<b>Chapter 3</b>	<b>Configuring Interfaces . . . . .</b>	<b>31</b>
	Configuring the In-Band Management Interface . . . . .	31
	ADSL2 and ADSL2+ Interfaces on NFX250 (NextGen) Devices . . . . .	32
	ADSL Interface Overview . . . . .	32
	ADSL2 and ADSL2+ . . . . .	32
	Example: Configuring ADSL SFP Interface on NFX250 Devices . . . . .	33
	VDSL2 Interfaces on NFX250 Devices . . . . .	35
	VDSL Interface Overview . . . . .	36
	VDSL2 Vectoring Overview . . . . .	36
	VDSL2 Network Deployment Topology . . . . .	36
	VDSL2 Interface Support on NFX Series Devices . . . . .	38
	VDSL2 Interface Compatibility with ADSL Interfaces . . . . .	38
	VDSL2 Interfaces Supported Profiles . . . . .	38
	Example: Configuring VDSL SFP Interface on NFX250 Devices . . . . .	39
<b>Chapter 4</b>	<b>Configuring Security . . . . .</b>	<b>43</b>
	IP Security on NFX Devices . . . . .	43
	Overview . . . . .	43
	Configuring Security . . . . .	45
	Configuring Interfaces . . . . .	45
	Configuring Routing Options . . . . .	46
	Configuring Security IKE . . . . .	46
	Configuring Security IPsec . . . . .	48
	Configuring Security Policies . . . . .	50
	Configuring Security Zones . . . . .	51
	UTM on NFX Devices . . . . .	51
	Application Security on NFX Devices . . . . .	52
	Intrusion Detection and Prevention on NFX Devices . . . . .	53
	Integrated User Firewall Support on NFX Devices . . . . .	53
<b>Chapter 5</b>	<b>Configuring Virtual Network Functions . . . . .</b>	<b>55</b>
	Configuring VNFs on NFX250 (NG) Devices . . . . .	55
	VNF Configuration Overview . . . . .	55
	Loading a VNF Image . . . . .	56
	Preparing the Bootstrap Configuration . . . . .	56
	Allocating a CPU for a VNF . . . . .	56
	Allocating Memory for a VNF . . . . .	57
	Configuring Interfaces and VLANs for a VNF . . . . .	57
	Configuring Storage Devices for VNFs . . . . .	59
	Instantiating a VNF . . . . .	60
	Instantiating a VNF Using an XML Descriptor File . . . . .	61
	Verifying that a VNF is Instantiated Successfully . . . . .	61
	Managing VNFs on NFX Series Devices . . . . .	61
	Managing VNF States . . . . .	62
	Managing VNF MAC Addresses . . . . .	62
	Managing the MTU of a VNF Interface . . . . .	63
	Accessing a VNF from the JCP . . . . .	64
	Viewing the List of VNFs . . . . .	64
	Displaying the Details of a VNF . . . . .	64

	Deleting a VNF . . . . .	65
<b>Chapter 6</b>	<b>Configuring Service Chaining . . . . .</b>	<b>67</b>
	Example: Configuring Service Chaining Using VLANs on NFX250 (NG)	
	Devices . . . . .	67
	Example: Configuring Service Chaining Using SR-IOV on NFX250 (NG)	
	Devices . . . . .	72
	Example: Configuring Service Chaining Using a Custom Bridge on NFX250 (NG)	
	Devices . . . . .	77
	Example: Configuring Cross-Connect on NFX250 (NG) Devices . . . . .	83
	Example: Configuring Service Chaining for LAN Routing on NFX250 (NG)	
	Devices . . . . .	91
	Example: Configuring Service Chaining for LAN to WAN Routing on NFX250 (NG)	
	Devices . . . . .	93
<b>Chapter 7</b>	<b>Troubleshooting . . . . .</b>	<b>97</b>
	Recovering the Root Password for NFX150 and NFX250 (NG) Devices . . . . .	97
	Troubleshooting Interfaces on NFX Devices . . . . .	100
	Monitoring Interface Status and Traffic on NFX Series Devices . . . . .	100
<b>Chapter 8</b>	<b>Operational Commands . . . . .</b>	<b>103</b>
	request vmhost cleanup . . . . .	104
	request vmhost file-copy . . . . .	105
	request vmhost halt . . . . .	106
	request vmhost mode . . . . .	107
	request vmhost power-off . . . . .	108
	request vmhost reboot . . . . .	109
	request vmhost software add . . . . .	110
	show system visibility cpu . . . . .	113
	show system visibility host . . . . .	116
	show system visibility memory . . . . .	124
	show system visibility network . . . . .	126
	show system visibility vnf . . . . .	131
	show vmhost connections . . . . .	136
	show vmhost control-plane . . . . .	138
	show vmhost crash . . . . .	139
	show vmhost forwarding-options analyzer . . . . .	140
	show vmhost memory . . . . .	142
	show vmhost mode . . . . .	143
	show vmhost status . . . . .	146
	show vmhost storage . . . . .	148
	show vmhost uptime . . . . .	150
	show vmhost version . . . . .	151
	show vmhost vlans . . . . .	152



# List of Figures

<b>Chapter 1</b>	<b>Overview .....</b>	<b>17</b>
	Figure 1: NFX250 Device .....	17
	Figure 2: NFX250 (NG) Software Architecture .....	18
<b>Chapter 2</b>	<b>Initial Configuration .....</b>	<b>25</b>
	Figure 3: Workflow for Initial Provisioning of an NFX Series Device .....	28
<b>Chapter 3</b>	<b>Configuring Interfaces .....</b>	<b>31</b>
	Figure 4: Typical VDSL2 End-to-End Connectivity and Topology Diagram .....	37
	Figure 5: Backward-Compatible ADSL Topology (ATM DSLAM) .....	37
<b>Chapter 4</b>	<b>Configuring Security .....</b>	<b>43</b>
	Figure 6: Scenario for Integrated User Firewall .....	54
<b>Chapter 6</b>	<b>Configuring Service Chaining .....</b>	<b>67</b>
	Figure 7: Configuring Service Chaining Using VLANs .....	68
	Figure 8: Service Chaining Using SR-IOV .....	73
	Figure 9: Service Chaining Using a Custom Bridge .....	77
	Figure 10: Configuring Cross-Connect .....	85
	Figure 11: Service Chaining for LAN Routing .....	92
	Figure 12: Service Chaining for LAN to WAN Routing .....	94





# List of Tables

	<b>About the Documentation</b> . . . . .	<b>xi</b>
	Table 1: Notice Icons . . . . .	xiii
	Table 2: Text and Syntax Conventions . . . . .	xiv
<b>Chapter 1</b>	<b>Overview</b> . . . . .	<b>17</b>
	Table 3: NFX250 Models and Specifications . . . . .	19
	Table 4: Supported Junos OS Releases on NFX Series Devices . . . . .	21
	Table 5: Software Compatibility Details with vSRX and Cloud CPE Solution . . . . .	23
<b>Chapter 2</b>	<b>Initial Configuration</b> . . . . .	<b>25</b>
	Table 6: Security Policies . . . . .	25
	Table 7: Interfaces . . . . .	25
<b>Chapter 3</b>	<b>Configuring Interfaces</b> . . . . .	<b>31</b>
	Table 8: Standard Bandwidths of DSL Operating Modes . . . . .	32
	Table 9: VDSL2 Annex A and Annex B Features . . . . .	38
	Table 10: Supported Profiles on the VDSL2 Interfaces . . . . .	39
<b>Chapter 4</b>	<b>Configuring Security</b> . . . . .	<b>43</b>
	Table 11: IPsec Features Supported on NFX150 . . . . .	44
<b>Chapter 5</b>	<b>Configuring Virtual Network Functions</b> . . . . .	<b>55</b>
	Table 12: CPUs Available for VNF Usage (Junos OS 19.1R1 Release) . . . . .	56
	Table 13: Memory Availability for VNF Usage . . . . .	57
<b>Chapter 8</b>	<b>Operational Commands</b> . . . . .	<b>103</b>
	Table 14: show system visibility cpu Output Fields . . . . .	113
	Table 15: show system visibility host Output Fields . . . . .	116
	Table 16: show system visibility memory Output Fields . . . . .	124
	Table 17: show system visibility network Output Fields . . . . .	126
	Table 18: show system visibility vnf Output Fields . . . . .	131
	Table 19: show vmhost connections Output Fields . . . . .	136
	Table 20: show vmhost forwarding-options analyzer Output Fields . . . . .	140
	Table 21: show vmhost vlans Output Fields . . . . .	152



# About the Documentation

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

## 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.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

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

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

## Merging a Snippet

To merge a snippet, follow these steps:

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

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

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

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

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

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

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

For more information about the **load** command, see [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 xiv 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>
<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>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> <li>To configure a stub area, include the <b>stub</b> statement at the [edit protocols <b>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</b> <default-metric <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	<b>broadcast</b>   <b>multicast</b>  ( <i>string1</i>   <i>string2</i>   <i>string3</i> )
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<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</b> [ <b>community-ids</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>
<b>&gt;</b> (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/>.



## CHAPTER 1

# Overview

- [NFX250 NextGen Overview on page 17](#)
- [NFX Product Compatibility on page 22](#)

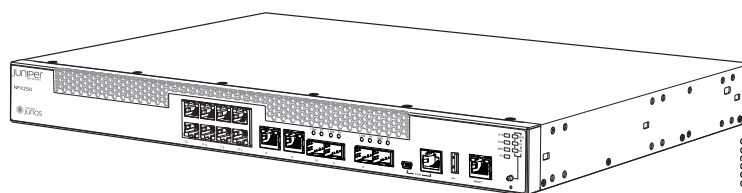
### NFX250 NextGen Overview

---

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

[Figure 1 on page 17](#) shows the NFX250 device.

*Figure 1: NFX250 Device*



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

NGFW includes security features such as

- VPN (see [VPN Feature Guide for Security Devices](#))
- NAT (see [NAT Feature Guide](#))
- ALG (see [Application Layer Gateways Feature Guide](#))
- Application Security (see [AppSecure Feature Guide](#))
- UTM features including Enhanced Web Filtering and Anti-Virus (see [UTM Feature Guide](#))

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

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



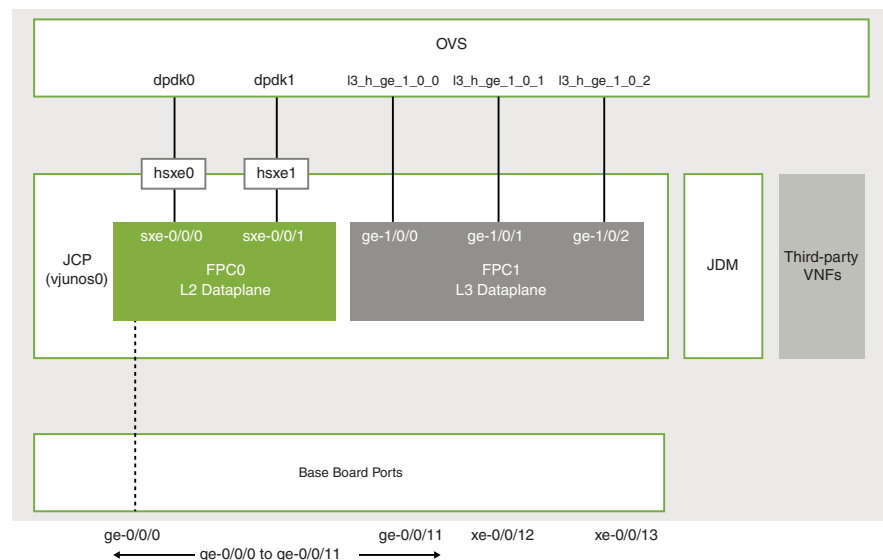
**NOTE:** For documentation purposes, NFX250 devices that use this architecture are referred to as NFX250 NextGen (NG) devices.

- [Software Architecture on page 18](#)
- [NFX250 Models on page 19](#)
- [Interfaces on page 20](#)
- [Performance Modes on page 21](#)
- [Benefits and Uses on page 21](#)
- [Junos OS Releases Supported on NFX Series Hardware on page 21](#)

## Software Architecture

Figure 2 on page 18 illustrates the software architecture of the NFX250 (NG). The architecture is designed to provide a unified control plane that functions as a single management point. Key components in the NFX250 (NG) software include the JCP, JDM, Layer 2 data plane, Layer 3 data plane, and VNFs.

**Figure 2: NFX250 (NG) Software Architecture**



Key components of the system software include:

- **Linux**—The host OS, which functions as the hypervisor.
- **VNF**—A VNF is a virtualized implementation of a network device and its functions. In the NFX250 (NG) architecture, Linux functions as the hypervisor, and it creates and

runs the VNFs. The VNFs include functions such as firewalls, routers, and WAN accelerators.

You can connect VNFs together as blocks in a chain to provide networking services.

- JCP—Junos virtual machine (VM) running on the host OS, Linux. The JCP functions as the single point of management for all the components.

The JCP supports:

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

In addition, the JCP enables VNF lifecycle management.

- JDM—An application container that manages VNFs and provides infrastructure services. The JDM functions in the background. Users cannot access the JDM directly.
- L2 data plane—Manages Layer 2 traffic. The Layer 2 dataplane forwards the LAN traffic to the Open vSwitch (OVS) bridge, which acts as the NFV backplane. The Layer 2 dataplane is mapped to the virtual FPC0 on the JCP.
- L3 data plane—Provides data path functions for the Layer 3 to Layer 7 services. The Layer 3 data plane is mapped to the virtual FPC1 on the JCP.
- Open vSwitch (OVS) bridge—The OVS bridge is a VLAN-aware system bridge that acts as the NFV backplane to which the VNFs, FPC1, and FPC0 connect. Additionally, you can create custom OVS bridges to isolate connectivity between different VNFs.

For the list of supported features, see [Feature Explorer](#).

## NFX250 Models

[Table 3 on page 19](#) lists the NFX250 device models and its specifications. For more information, see the *NFX250 Hardware Guide*.

**Table 3: NFX250 Models and Specifications**

Components	NFX250-S1	NFX250-S2	NFX250-S1E
CPU	2.0 GHz 6-core Intel CPU	2.0 GHz 6-core Intel CPU	2.0 GHz 6-core Intel CPU
RAM	16 GB	32 GB	16 GB
Storage	100 GB SSD	400 GB SSD	200 GB SSD
Form Factor	Desktop	Desktop	Desktop

Table 3: NFX250 Models and Specifications (continued)

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

## Interfaces

The NFX250 (NG) device includes the following network interfaces:

- Ten 1-Gigabit Ethernet RJ-45 ports and two 1-Gigabit Ethernet network ports that support small form-factor pluggable (SFP) transceivers. The ports follow the naming convention, ge-0/0/*n*, where *n* ranges from 0 to 11. These ports are used for LAN connectivity.
- Two 1-Gigabit or 10-Gigabit uplink ports that support small form-factor pluggable plus (SFP+) transceivers. The ports follow the naming convention xe-0/0/*n*, where the value of *n* is either 12 or 13. These ports are used as WAN uplink ports.
- A dedicated management port labeled **MGMT** (fxp0) functions as the out-of-band management interface. The fxp0 interface is assigned the IP address 192.168.1.1/24.
- Two static interfaces, sxe-0/0/0 and sxe-0/0/1, which connect the Layer 2 data plane (FPC0) to the OVS backplane.



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

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

## Performance Modes

Starting in Junos OS Release 19.1R1, NFX250 (NextGen) 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.

## Benefits and Uses

The NFX250 (NG) provides the following benefits:

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

## Junos OS Releases Supported on NFX Series Hardware

The [Table 4 on page 21](#) provides details of Junos OS software releases supported on the NFX Series devices.

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

NFX Series Platform	Supported Junos OS Release	Software Package	Software Downloads Page
NFX150	18.1R1 or later	nfx-3 (jinstall-host-nfx-3-x86-64-<release-number>-secure-signed.tgz)	<a href="#">NFX150 Software Download Page</a>

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

NFX Series Platform	Supported Junos OS Release	Software Package	Software Downloads Page
NFX250	15.1X53-D45, 15.1X53-D47, 15.1X53-D470, and 15.1X53-D471	nfx-2  (jinstall-host-nfx-2-flex-x86-64-<release-number>-secure-signed.tgz)	<a href="#">NFX250 Software Download Page</a>
	17.2R1 or later		
	18.4 R1 or later	nfx-3  (jinstall-host-nfx-3-flex-x86-64-<release-number>-secure-signed.tgz)  install-media-host-usb-nfx-3-x86-64-<release-number>-secure.img	<a href="#">NFX250 Software Download Page</a>

## NFX Product Compatibility

- [Hardware Compatibility on page 22](#)
- [Software Version Compatibility on page 22](#)

### 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, NFX150 and NFX250 devices support the same version of platform software and vSRX. For example, see [Table 5 on page 23](#).

### NFX250 Software Version Compatibility

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

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

NFX250 Junos OS Release	vSRX	Cloud CPE Solution
15.1X53-D40.3	15.1X49-D40.6	Cloud CPE Solution 2.0
15.1X53-D41.6	15.1X49-D40.6	Cloud CPE Solution 2.1
15.1X53-D102.2	15.1X49-D61	Cloud CPE Solution 3.0
15.1X53-D47.4	15.1X49-D100.6	Cloud CPE Solution 3.0.1
15.1X53-D490	15.1X49-D143	Cloud CPE Solution 4.0
15.1X53-D495	15.1X49-D160	Cloud CPE Solution 4.1
15.1X53-D496	15.1X49-D170	Cloud CPE Solution 4.1
15.1X53-D45.3	15.1X49-D61	Not applicable
17.2R1	15.1X49-D78.3	Not applicable
17.3R1	15.1X49-D78.3	Not applicable
17.4R1	15.1X49-D78.3	Not applicable
15.1X53-D471	15.1X49-D143	Not applicable
18.1R1	18.1R1	Not applicable
18.1R2	18.1R2	Not applicable
18.1R3	18.1R3	Not applicable
18.2R1	18.2R1	Not applicable
18.3R1	18.3R1	Not applicable
18.4R1	18.4R1	Not applicable





## CHAPTER 2

# Initial Configuration

- [Initial Configuration on NFX250 \(NG\) Devices on page 25](#)
- [Zero Touch Provisioning on NFX Series Devices on page 27](#)

### Initial Configuration on NFX250 (NG) Devices

---

- [Factory Default Settings on page 25](#)
- [Enabling Basic Connectivity on page 26](#)
- [Establishing the Connection on page 27](#)

### Factory Default Settings

The NFX250 (NG) is shipped with the following factory default settings:

*Table 6: Security Policies*

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

*Table 7: Interfaces*

Port Label	Interface	Security Zone	DHCP State	IP Address
0/1 to 0/11	ge-0/0/1 to ge-0/0/11	trust	server	192.168.2.1/24
0/12 to 0/13	xe-0/0/12 to xe-0/0/13	untrust	client	ISP assigned
MGMT	fxp0	N/A	N/A	192.168.1.1/24

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

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

## Enabling Basic Connectivity

1. Ensure that the device is powered on.
2. Connect to the console port:
  - a. Plug one end of the Ethernet cable into the console port on your device.
  - b. Connect the other end of the Ethernet cable to the RJ-45 to DB-9 serial port adapter shipped with your device.
  - c. Connect the RJ-45 to DB-9 serial port adapter to the serial port on the management device. Use the following values to configure the serial port:  
Bits per second—9600; Parity—None; Data bits—8; Stop bits—1; Flow control—None.



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

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

3. Use any terminal emulation program such as HyperTerminal to connect to the device console. The CLI displays a login prompt.
4. Log in as **root**. If the software completes booting before you connect to the console, you might need to press the Enter key for the prompt to appear.

```
login: root
```

5. Start the CLI.

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

6. Enter configuration mode.

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

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

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

Retype new password: *password*

8. Enable SSH service for the root user.

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

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

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

10. Commit the configuration.

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

## Establishing the Connection

1. Connect the device to the ISP by connecting one of the WAN ports (0/12 and 0/13) to the ISP. The device is assigned an IP address by the ISP through DHCP.



**NOTE:** For information about NFX250 (NG) interfaces, see [Table 7 on page 25](#).

2. Connect the laptop to one of the front panel LAN ports (0/0 to 0/11). The laptop is assigned an IP address by the DHCP server running on the device.
3. Open a browser window on your laptop, navigate to [www.juniper.net](http://www.juniper.net), and verify your connectivity.

## Zero Touch Provisioning on NFX Series Devices

- [Understanding Zero Touch Provisioning on page 27](#)
- [Provisioning an NFX Series Device on page 28](#)
- [Provisioning an NFX Series Device Using Sky Enterprise on page 29](#)

## 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.

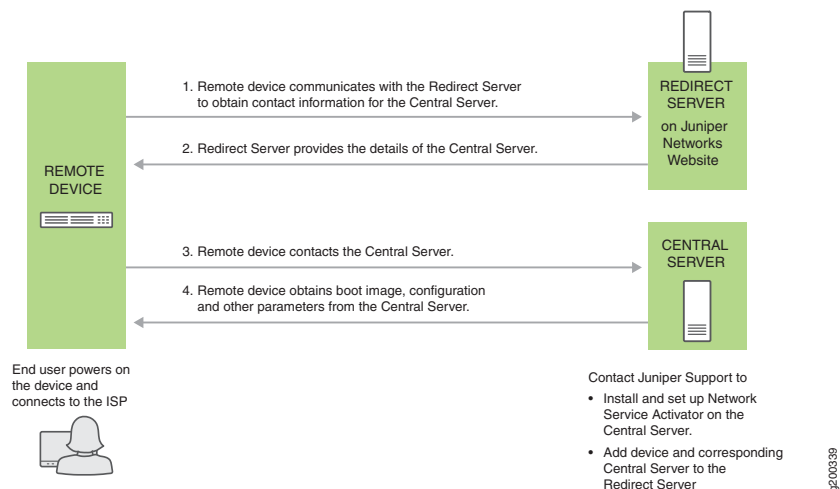
The NFX Series device is shipped with a factory default configuration. The factory default configuration includes the URL of the redirect server, that is used to connect to the central servers by using a secure encrypted connection.

- 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.

## Provisioning an NFX Series Device

Figure 3 on page 28 illustrates the workflow of the initial provisioning of NFX Series devices.

*Figure 3: 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 3 on page 28](#) illustrates the workflow of the initial provisioning of NFX Series devices using Sky Enterprise.

The provisioning workflow proceeds as follows:

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



## CHAPTER 3

# Configuring Interfaces

- [Configuring the In-Band Management Interface on page 31](#)
- [ADSL2 and ADSL2+ Interfaces on NFX250 \(NextGen\) Devices on page 32](#)
- [VDSL2 Interfaces on NFX250 Devices on page 35](#)

### 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. 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 NFX250 (NextGen) Devices

- [ADSL Interface Overview on page 32](#)
- [Example: Configuring ADSL SFP Interface on NFX250 Devices on page 33](#)

### ADSL Interface Overview

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

ADSL transmission is asymmetric because the downstream bandwidth is typically greater than the upstream bandwidth. The typical bandwidths of ADSL2 and ADSL2+ circuits are defined in [Table 8 on page 32](#).

*Table 8: Standard Bandwidths of DSL Operating Modes*

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

ADSL2 and ADSL2+ support the following standards:

- LLC SNAP bridged 802.1q
- VC MUX bridged

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

### ADSL2 and ADSL2+

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



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

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

### Example: Configuring ADSL SFP Interface on NFX250 Devices

- [Requirements on page 33](#)
- [Overview on page 33](#)
- [Configuration on page 33](#)
- [Results on page 35](#)

#### Requirements

This example uses the following hardware and software components:

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

#### Overview

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

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



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

#### Configuration

##### Step-by-Step Procedure

To configure ADSL SFP interfaces on NFX250 (NextGen) devices:

1. Connect to the host.

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

2. Allocate hugepages:

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

3. Configure virtual interfaces:

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

4. Create VLANs using VLAN IDs:

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

5. Configure interfaces:

```
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan50
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan101
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan200
user@host# set interfaces ge-0/0/11 native-vlan-id 50
user@host# set interfaces ge-0/0/11 dsl-sfp-options adsl-options vpi 3
user@host# set interfaces ge-0/0/11 dsl-sfp-options adsl-options vci 32
user@host# set interfaces ge-0/0/11 dsl-sfp-options adsl-options encaps llcsnap-bridged-802dot1q
user@host# set interfaces ge-0/0/11 dsl-sfp-options vdsl-options profile 17a
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan50
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan101
user@host# set interfaces ge-1/0/3 vlan-tagging
user@host# set interfaces ge-1/0/3 unit 0 vlan-id 50
user@host# set interfaces ge-1/0/3 unit 0 family inet address 130.1.1.1/24
user@host# set interfaces ge-1/0/3 unit 0 family inet6 address 2001::1/64
```

6. Commit the configuration.

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

## Results

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

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

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

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

**Related  
Documentation**

## VDSL2 Interfaces on NFX250 Devices

- [VDSL Interface Overview on page 36](#)
- [VDSL2 Network Deployment Topology on page 36](#)

- [VDSL2 Interface Support on NFX Series Devices on page 38](#)
- [Example: Configuring VDSL SFP Interface on NFX250 Devices on page 39](#)

## VDSL Interface Overview

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 on page 36](#)

---

### 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 4 on page 37](#).
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 5 on page 37](#).

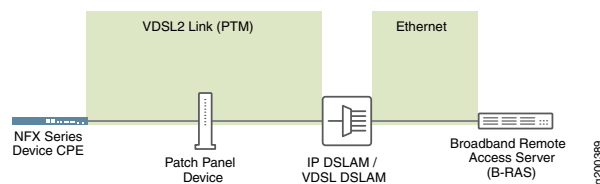


**NOTE:** The VDSL2 technology is backward compatible with ADSL2 and ADSL2+. VDSL2 provides an ADSL2 and ADSL2+ interface in an ATM DSLAM topology and provides a VDSL2 interface in an IP or VDSL DSLAM topology.

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

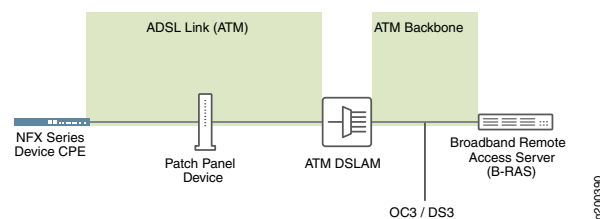
[Figure 4 on page 37](#) shows a typical VDSL2 network topology.

**Figure 4: Typical VDSL2 End-to-End Connectivity and Topology Diagram**



[Figure 5 on page 37](#) shows a backward-compatible ADSL topology using ATM DSLAM.

**Figure 5: 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 9 on page 38](#). (Platform support depends on the Junos OS release in your installation.)

**Table 9: VDSL2 Annex A and Annex B Features**

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



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

### VDSL2 Interface Compatibility with ADSL Interfaces

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



**NOTE:**

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

### VDSL2 Interfaces Supported Profiles

A profile is a table that contains a list of pre-configured VDSL2 settings.

[Table 10 on page 39](#) lists the different profiles supported on the VDSL2 interfaces and their properties.

Table 10: Supported Profiles on the VDSL2 Interfaces

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

### Example: Configuring VDSL SFP Interface on NFX250 Devices

- [Requirements on page 39](#)
- [Overview on page 39](#)
- [Configuration on page 40](#)
- [Results on page 42](#)

#### Requirements

This example uses the following hardware and software components:

- NFX250 device running Junos OS Release 15.1X53-D495.

#### Overview

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

- Physical interface - **ge-0/0/11**
- Virtual network function (VNF) - **nfx250-a-vsrx1**
- Memory size - **4194304**
- VDSL SFP options - **profile auto and carrier auto**

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



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

## Configuration

### Step-by-Step Procedure

To configure VDSL SFP interfaces on NFX250 devices:

1. Connect to the host.

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

2. Allocate hugepages:

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

3. Create VLANs using VLAN IDs:

```
user@host# set host-os vlans xdsl-test vlan-id 50
user@host# set host-os vlans vlan130 vlan-id 130
user@host# set host-os vlans vlan131 vlan-id 131
user@host# set host-os vlans vlan132 vlan-id 132
```

4. Allocate resources for a VNF:

```
user@host# set virtual-network-functions nfx250-a-vsrx1 image
/var/public/media-vsrx-vm disk-15.1-2018-04-24.0_DEV_X_151_X49.qcow2
user@host# set virtual-network-functions nfx250-a-vsrx1 virtual-cpu 0 physical-cpu 2
user@host# set virtual-network-functions nfx250-a-vsrx1 virtual-cpu 1 physical-cpu 6
user@host# set virtual-network-functions nfx250-a-vsrx1 virtual-cpu count 2
user@host# set virtual-network-functions nfx250-a-vsrx1 virtual-cpu features
hardware-virtualization
user@host# set virtual-network-functions nfx250-a-vsrx1 no-default-interfaces
user@host# set virtual-network-functions nfx250-a-vsrx1 memory size 4194304
user@host# set virtual-network-functions nfx250-a-vsrx1 memory features hugepages
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth0 management
out-of-band
```

5. Map VNF interfaces to NFV backplane:

```
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth1 mapping vlan
mode trunk
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth1 mapping vlan
members vlan130
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth2 mapping vlan
mode trunk
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth2 mapping vlan
members vlan130
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth2 mapping vlan
members vlan131
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth2 mapping vlan
members vlan132
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth2 mapping vlan
members xdsl-test
```



```

user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth2 mapping vlan
native-vlan-id 50
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth3 mapping vlan
mode trunk
user@host# set virtual-network-functions nfx250-a-vsrx1 interfaces eth3 mapping vlan
members xdsl-test

```

6. Configure the Junos Control Plane (JCP):

```

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
xdsl-test
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members
vlan130
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members
vlan131
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members
vlan132
user@host# set interfaces ge-0/0/11 native-vlan-id 50
user@host# set interfaces ge-0/0/11 dsl-sfp-options vdsl-options profile auto
user@host# set interfaces ge-0/0/11 dsl-sfp-options vdsl-options carrier auto
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members
xdsl-test
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan130
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan131
user@host# set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members vlan132
user@host# set vlans vlan130 vlan-id 130
user@host# set vlans vlan131 vlan-id 131
user@host# set vlans vlan132 vlan-id 132
user@host# set vlans xdsl-test vlan-id 50

```

7. Commit the configuration.

```

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

```

## Results

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

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

Physical interface: ge-0/0/11, Enabled, Physical link is Up
  Interface index: 258, SNMP ifIndex: 533
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, VDSL2 mode, Speed: 1000mbps,
  BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled,
  Auto-negotiation: Enabled, Remote fault: Online, IEEE 802.3az Energy Efficient
  Ethernet: Disabled, Auto-MDIX: Enabled
  VDSL status:
    Modem status : Showtime (Profile-12a)
    VDSL profile  :      Auto      Annex B
    Device flags  : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 08:b2:58:1f:0d:0d, Hardware address: 08:b2:58:1f:0d:0d
  Last flapped   : 2018-11-02 08:43:20 UTC (6d 00:29 ago)
  Input rate     : 888 bps (1 pps)
  Output rate    : 888 bps (1 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics
    Bit errors           Seconds 0
    Errored blocks       Seconds 0
  Ethernet FEC statistics
    FEC Corrected Errors      Errors 0
    FEC Uncorrected Errors    Errors 0
    FEC Corrected Errors Rate Errors 0
    FEC Uncorrected Errors Rate Errors 0
    PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

Logical interface ge-0/0/11.0 (Index 336) (SNMP ifIndex 535)
  Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
  Input packets : 0
  Output packets: 0
  Protocol eth-switch, MTU: 1514
  Flags: Trunk-Mode
```

[Related  
Documentation](#)

## CHAPTER 4

# Configuring Security

- [IP Security on NFX Devices on page 43](#)
- [UTM on NFX Devices on page 51](#)
- [Application Security on NFX Devices on page 52](#)
- [Intrusion Detection and Prevention on NFX Devices on page 53](#)
- [Integrated User Firewall Support on NFX Devices on page 53](#)

## IP Security on NFX Devices

---

- [Overview on page 43](#)
- [Configuring Security on page 45](#)

### Overview

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

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

The key concepts of IPsec include:

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

For more information about IPsec security protocols, see [IPsec Security Protocols](#).

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

For more information about IPsec tunnel negotiation, see [IPsec Tunnel Negotiation](#).

[Table 11 on page 44](#) lists the IPsec features supported on NFX150 devices.

**Table 11: IPsec Features Supported on NFX150**

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>

Table 11: IPsec Features Supported on NFX150 (continued)

Features	Reference
NAT-T	<a href="#">Understanding NAT-T</a>
Dead Peer Detection	<a href="#">Understanding VPN Monitoring</a>

## Configuring Security

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

- [Configuring Interfaces on page 45](#)
- [Configuring Routing Options on page 46](#)
- [Configuring Security IKE on page 46](#)
- [Configuring Security IPsec on page 48](#)
- [Configuring Security Policies on page 50](#)
- [Configuring Security Zones on page 51](#)

### 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
```

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**

- [Intrusion Detection and Prevention on NFX Devices on page 53](#)
- [Integrated User Firewall Support on NFX Devices on page 53](#)

---

## Application Security on NFX Devices

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

The AppSecure feature comprises of the following services:

- Application identification (AppID)—Recognizes traffic at different network layers using characteristics other than port number. Once the application is determined, AppSecure service modules can be configured to monitor and control traffic for tracking, prioritization, access control, detection, and prevention based on the application ID of the traffic. For more information, see *Application Identification*.
- Application Tracking (AppTrack)—Tracks and reports applications passing through the device. For more information, see *Application Tracking on NFX Devices*.
- Application Firewall (AppFW)—Implements an application firewall using application-based rules. For more information, see *Application Firewall on NFX Devices*.
- Application Quality of Service (AppQoS)—Provides quality-of-service prioritization based on application awareness. For more information, see *Application QoS*.
- Advanced policy-based routing (APBR)—Classifies session based on applications and applies the configured rules to reroute the traffic. For more information, see *Advanced Policy-Based Routing on NFX Devices*.

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

**Related Documentation** • [Integrated User Firewall Support on NFX Devices on page 53](#)

---

## 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 Feature Guide*.

**Related Documentation** • [UTM on NFX Devices on page 51](#)

---

## Integrated User Firewall Support on NFX Devices

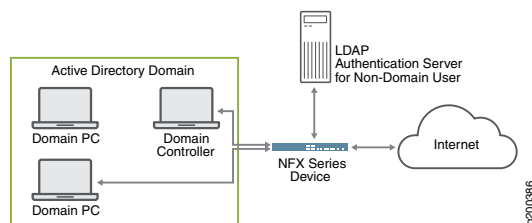
---

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 6 on page 54 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 6: 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 *Integrated User Firewall Feature Guide for NFX Devices*.

**Related Documentation**

- [UTM on NFX Devices on page 51](#)

## CHAPTER 5

# Configuring Virtual Network Functions

- [Configuring VNFs on NFX250 \(NG\) Devices on page 55](#)
- [Managing VNFs on NFX Series Devices on page 61](#)

### Configuring VNFs on NFX250 (NG) Devices

---

NFX250 (NG) 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.

The number of VNFs that can be instantiated on the device is dependent on the system resources such as number of cores in the CPU and system memory.

- [VNF Configuration Overview on page 55](#)
- [Loading a VNF Image on page 56](#)
- [Preparing the Bootstrap Configuration on page 56](#)
- [Allocating a CPU for a VNF on page 56](#)
- [Allocating Memory for a VNF on page 57](#)
- [Configuring Interfaces and VLANs for a VNF on page 57](#)
- [Configuring Storage Devices for VNFs on page 59](#)
- [Instantiating a VNF on page 60](#)
- [Instantiating a VNF Using an XML Descriptor File on page 61](#)
- [Verifying that a VNF is Instantiated Successfully on page 61](#)

### VNF Configuration Overview

Configuring a VNF includes the following tasks:

1. Log in to the JCP.
2. Load the VNF image.
3. Allocate a CPU for the VNF.
4. Allocate memory for the VNF.
5. Configure interfaces and VLANs for the VNF.
6. Configure storage devices for the VNF.

7. Instantiate the VNF.
8. Verify that the VNF is instantiated successfully.

## Loading a VNF Image

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



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

## Preparing the Bootstrap Configuration

You can bootstrap a VNF by using either a CD or a USB storage device that contains a bootstrap configuration ISO file.

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

## Allocating a CPU for a VNF

Table 12 on page 56 lists the CPUs available for VNF usage for the NFX250 models.

**Table 12: CPUs Available for VNF Usage (Junos OS 19.1R1 Release)**

Model	CPUs Available for VNF Usage		
NFX250-S1	0	4	8
NFX250-S2	0	4	8
NFX250-S1E	0	4	8



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

To check the CPU availability and its status:

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

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

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

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

2. Connect a virtual CPU to a physical CPU:



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

3. Commit the configuration:

```
user@host# commit
```

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



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

To enable hardware virtualization or hardware acceleration for VNF CPUs:

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

## Allocating Memory for a VNF

By default, a certain amount of memory is allocated for VNFs. [Table 13 on page 57](#) lists the possible memory availability for VNF usage for the NFX250 models.

*Table 13: Memory Availability for VNF Usage*

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

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

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



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

## Configuring Interfaces and VLANs for a VNF

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

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

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

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

To attach a VNF interface to a VLAN:

- Create a VLAN:

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

- Attach a VNF interface to a VLAN:

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



NOTE:

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

To specify the target PCI address for a VNF interface:

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

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

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



NOTE:

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

To delete a VNF interface:

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

**NOTE:**

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

## Configuring Storage Devices for VNFs

An NFX250 (NG) device supports the following storage options for VNFs:

- CD-ROM
- Disk
- USB

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

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

You can specify a valid device name in the format hdx, sdx, or vdx—for example, hdb, sdc, vdb, and so on.

To add a virtual USB storage device:

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

To attach an additional hard disk:

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

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

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

**NOTE:**

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

## Instantiating a VNF

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

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



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

To instantiate a VNF by using an image:

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



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

(Optional) To specify a UUID for the VNF:

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

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



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

## Instantiating a VNF Using an XML Descriptor File

You can instantiate a VNF by using an XML descriptor file. You must save the XML descriptor 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 the VNF and create a VNF again.

## Verifying that a VNF is Instantiated Successfully

To verify that the VNF is instantiated successfully:

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

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

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

## Managing VNFs on NFX Series Devices

- [Managing VNF States on page 62](#)
- [Managing VNF MAC Addresses on page 62](#)
- [Managing the MTU of a VNF Interface on page 63](#)
- [Accessing a VNF from the JCP on page 64](#)
- [Viewing the List of VNFs on page 64](#)
- [Displaying the Details of a VNF on page 64](#)
- [Deleting a VNF on page 65](#)

## 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 can be configured in the system is limited to 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.



## CHAPTER 6

# Configuring Service Chaining

- [Example: Configuring Service Chaining Using VLANs on NFX250 \(NG\) Devices on page 67](#)
- [Example: Configuring Service Chaining Using SR-IOV on NFX250 \(NG\) Devices on page 72](#)
- [Example: Configuring Service Chaining Using a Custom Bridge on NFX250 \(NG\) Devices on page 77](#)
- [Example: Configuring Cross-Connect on NFX250 \(NG\) Devices on page 83](#)
- [Example: Configuring Service Chaining for LAN Routing on NFX250 \(NG\) Devices on page 91](#)
- [Example: Configuring Service Chaining for LAN to WAN Routing on NFX250 \(NG\) Devices on page 93](#)

### Example: Configuring Service Chaining Using VLANs on NFX250 (NG) Devices

---

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

- [Requirements on page 67](#)
- [Overview on page 67](#)
- [Configuration on page 68](#)

#### Requirements

This example uses an NFX250 (NG) device running Junos OS Release 19.1R1.

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

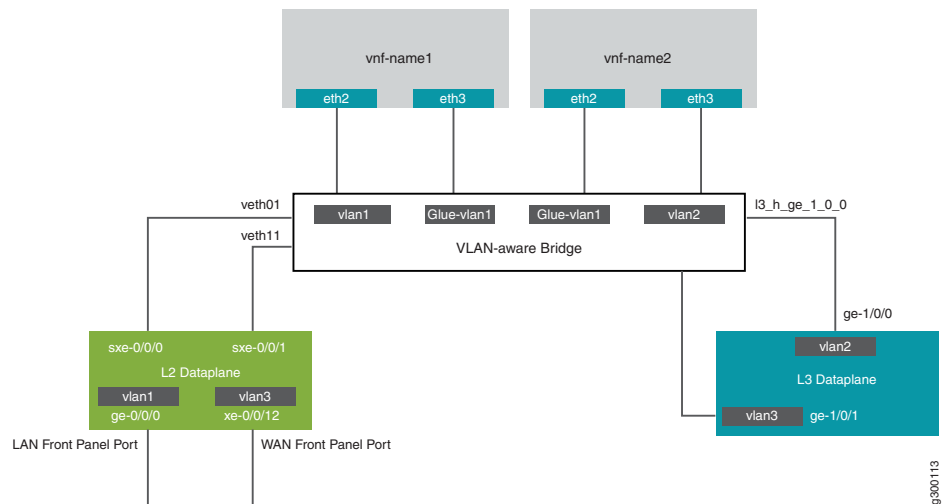
#### Overview

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

## Topology

This example uses a single NFX250 (NG) device running Junos OS, as shown in [Figure 7 on page 68](#).

**Figure 7: Configuring Service Chaining Using VLANs**



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

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

## Configuration

- [Configuring the JCP Interfaces on page 69](#)
- [Configuring the VNF Interfaces and Creating the Service Chain on page 72](#)

## Configuring the JCP Interfaces

### Step-by-Step Procedure

To configure the interfaces:

1. Connect to the JCP.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11. Commit the configuration.

```
user@host# commit
```

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

```
[edit]
user@host# show interfaces ge-0/0/0

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

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

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

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

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

```
[edit]
user@host# show interfaces sxe-0/0/0

mtu 9192;
unit 0 {
  family ethernet-switching {
    interface-mode trunk;
    vlan {
```

```
        members [ default vlan1 ];
    }
}
```

```
[edit]
user@host# show interfaces sxe-0/0/1
```

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

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

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

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

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

## Configuring the VNF Interfaces and Creating the Service Chain

### Step-by-Step Procedure

Configure the VNF interfaces.

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

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

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

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

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

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

4. Configure the IP addresses and static routes for each interface of the VNFs as shown in [Figure 7 on page 68](#).

## Example: Configuring Service Chaining Using SR-IOV on NFX250 (NG) Devices

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

- [Requirements on page 72](#)
- [Overview on page 72](#)
- [Configuration on page 74](#)

### Requirements

This example uses an NFX250 (NG) device running Junos OS Release 19.1R1.

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

### Overview

This example uses the front panel ports ge-0/0/0 and xe-0/0/13 associated with the PFE, and its internal-facing ports, sxe-0/0/0 and sxe-0/0/1. The internal NIC ports, sxe0 and sxe1, are not configured directly; instead, they are abstracted at the host OS layer



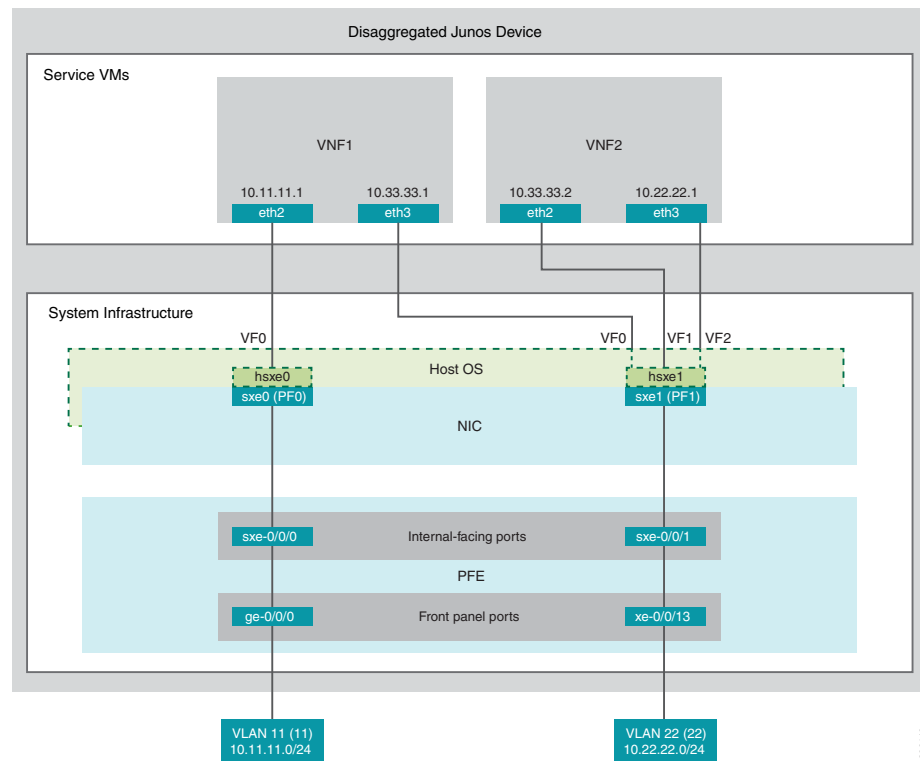
and configured as interfaces `hsxe0` and `hsxe1`. The VNFs use two interfaces, `eth2` and `eth3`. These elements are generally separated into a LAN side and a WAN side.

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

### Topology

Figure 8 on page 73 shows the topology for this example.

**Figure 8: Service Chaining Using SR-IOV**



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

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



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

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

## Configuration

This example describes:

- [Configuring the Packet Forwarding Engine Interfaces on page 74](#)
- [Configuring the VNF Interfaces and Creating the Service Chain on page 76](#)

### Configuring the Packet Forwarding Engine Interfaces

#### CLI Quick Configuration

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

```
[edit]
user@host#

set vlans Vlan11 vlan-id 11
set interfaces ge-0/0/0.0 family ethernet-switching vlan member Vlan11
set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
set interfaces sxe-0/0/0.0 family ethernet-switching vlan member Vlan11
set vlans Vlan22 vlan-id 22
set interfaces xe-0/0/13.0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/13.0 family ethernet-switching vlan member Vlan22
set interfaces sxe-0/0/1.0 family ethernet-switching interface-mode trunk
set interfaces sxe-0/0/1.0 family ethernet-switching vlan member Vlan22
```

#### Step-by-Step Procedure

To configure the Packet Forwarding Engine interfaces:

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

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

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

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

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

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

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

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

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

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

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

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

```
user@host# set interfaces xe-0/0/13.0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-0/0/13.0 family ethernet-switching vlan members Vlan22
```

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

```
user@host# set interfaces sxe-0/0/1.0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/1.0 family ethernet-switching vlan members Vlan22
```

7. Commit the configuration.

```
user@host# commit
```

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

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

```
user@host> show interfaces xe-0/0/13
unit 0 {
  family ethernet-switching {
    interface-mode trunk;
    vlan {
      members Vlan22;
    }
  }
}
```

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

```
user@host> show interfaces sxe-0/0/1
```

```
unit 0 {
  family ethernet-switching {
    interface-mode trunk;
    vlan {
      members Vlan22;
    }
  }
}
```

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

```
Vlan11 {
  vlan-id 11;
}
Vlan22 {
  vlan-id 22;
}
```

### Configuring the VNF Interfaces and Creating the Service Chain

#### Step-by-Step Procedure

To configure the VNF interfaces and create the service chain:

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

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

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

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

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

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

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

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

- Related Documentation**
- [Understanding Service Chaining on Disaggregated Junos OS Platforms](#)
  - [Disaggregated Junos OS VMs](#)
  - [Understanding SR-IOV Usage](#)

## Example: Configuring Service Chaining Using a Custom Bridge on NFX250 (NG) Devices

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

- [Requirements on page 77](#)
- [Overview on page 77](#)
- [Configuration on page 78](#)
- [Verifying the Configuration on page 79](#)

### Requirements

This example uses an NFX250 (NG) device running Junos OS Release 19.1R1.

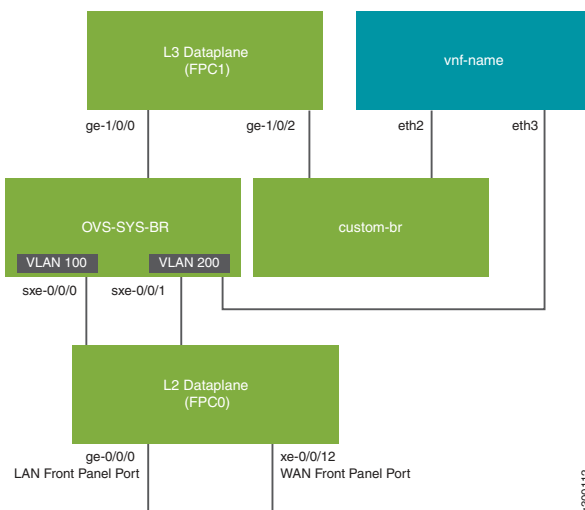
### Overview

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

### Topology

This example uses the topology shown in [Figure 9 on page 77](#).

**Figure 9: Service Chaining Using a Custom Bridge**



## Configuration

- [Configuring VLANs and Creating the Custom Bridge on page 78](#)
- [Configuring the Layer 2 Datapath on page 78](#)
- [Configuring the Layer 3 Datapath on page 78](#)
- [Configuring the VNF on page 79](#)

### Configuring VLANs and Creating the Custom Bridge

#### Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces:  
  

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan200 vlan-id 200
```
2. Create a custom bridge:  
  

```
user@host# set vmhost vlans custom-br vlan-id none
```
3. Map the Layer 3 interface to the custom bridge:  
  

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

### Configuring the Layer 2 Datapath

#### Step-by-Step Procedure

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

```
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members vlan200
```
2. Configure the internal-facing interfaces as trunk ports and add them to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.  
  

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

### Configuring the Layer 3 Datapath

#### Step-by-Step Procedure

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

```
user@host# set interfaces ge-1/0/0 vlan-tagging
user@host# set interfaces ge-1/0/0 unit 0 vlan-id 100
```

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

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

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

## Configuring the VNF

### Step-by-Step Procedure



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

1. Launch the VNF:

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

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

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

3. Pin a virtual CPU to a physical CPU:

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

4. Configure the vmhost instance:

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

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

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

6. Create a VNF interface on the OVS bridge:

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

7. Specify the memory allocation for the VNF:

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

## Verifying the Configuration

- [Verify the Control Plane Configuration on page 80](#)
- [Verifying the Data Plane Configuration on page 81](#)

## Verify the Control Plane Configuration

**Purpose** Verify the control plane configuration:

**Action** • Verify that the VLANs are configured:

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

Routing instance	VLAN name	Tag	Interfaces
default-switch	default	1	
default-switch	vlan100	100	ge-0/0/0.0* sxe-0/0/0.0*
default-switch	vlan200	200	sxe-0/0/1.0* xe-0/0/12.0*

• Verify the vmhost VLANs:

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

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

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

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

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

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

To view more details of the VNF:

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

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

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



## Verifying the Data Plane Configuration

**Purpose** Verify the data plane configuration.

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

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

For example:

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

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 149, SNMP ifIndex: 517
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Link-mode: Full-duplex,
  Speed: 1000mbps, Duplex: Full-Duplex, BPDU Error: None,
  Loop Detect PDU Error: None, Ethernet-Switching Error: None, MAC-REWRITE
  Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online, IEEE
  802.3az Energy Efficient Ethernet: Disabled, Auto-MDIX: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 30:7c:5e:4c:78:03, Hardware address: 30:7c:5e:4c:78:03
  Last flapped   : 2018-11-26 11:03:32 UTC (04:25:39 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics
    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 330) (SNMP ifIndex 519)
  Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
  Input packets : 0
  Output packets: 0
  Protocol eth-switch, MTU: 1514
  Flags: Trunk-Mode
```

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

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

```
Network Name : custom-br

Interface : custom-br
Type : internal, Link type : Full-Duplex, MAC : 2e:8e:a3:e3:e5:40
MTU : [], Link State :down, Admin State : down
```

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

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

Network Name : ovs-sys-br

```
Interface : ovs-sys-br
Type : internal, Link type : Full-Duplex, MAC : 66:9c:3f:25:04:40
MTU : [], Link State :down, Admin State : down
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0
```

```
Interface : dpdk0
Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:1a:c6:ee
MTU : [], Link State :up, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0
```

```
Interface : dpdk1
Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:7b:6c:47
MTU : [], Link State :up, Admin State : up
IPV4 : None, Netmask : None
IPV6 : None, IPV6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0
```

```

Interface : l3_h_ge_1_0_0
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0

Interface : l3_h_ge_1_0_1
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0

Interface : l3_h_ge_1_0_2
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0

Interface : vnf-name_eth3
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : 1500, Link State :down, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0

```

## Example: Configuring Cross-Connect on NFX250 (NG) Devices

This example shows how to configure the cross-connect feature on NFX250 (NG) devices.

- [Requirements on page 84](#)
- [Overview on page 84](#)

- [Configuration on page 85](#)
- [Verifying the Configuration on page 87](#)

## Requirements

This example uses an NFX250 (NG) device running Junos OS Release 19.1R1.

## 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:

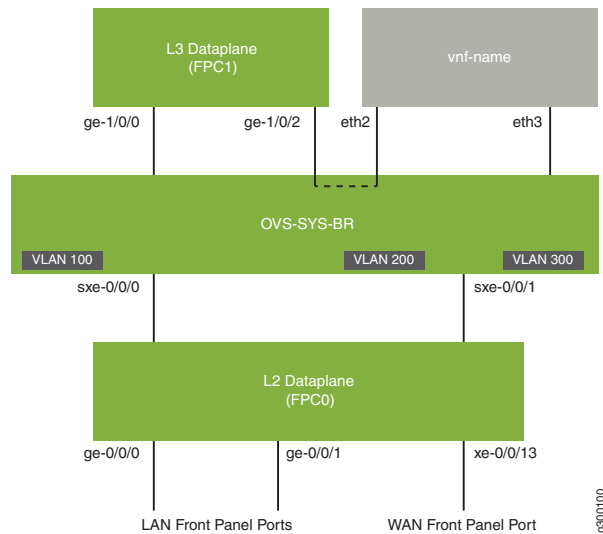
- Port cross-connect between two VNF interfaces for all network traffic.
- VLAN-based traffic forwarding between VNF interfaces that support the following functions:
  - Provides an option to switch traffic based on a VLAN ID.
  - Supports VLAN PUSH, POP, and SWAP operations.
  - Supports network traffic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

## Topology

---

This example uses the topology shown in [Figure 10 on page 85](#).

Figure 10: Configuring Cross-Connect



## Configuration

- [Configuring VLANs on page 85](#)
- [Configure the Layer 2 Datapath on page 85](#)
- [Configuring the Layer 3 Datapath on page 86](#)
- [Configuring the VNF on page 86](#)
- [Configuring Cross-Connect on page 87](#)

### Configuring VLANs

#### Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.  

```
user@host# set vlans vlan100 vlan-id 100
```
2. Configure a VLAN for the WAN-side interface.  

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

### Configure the Layer 2 Datapath

#### Step-by-Step Procedure

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

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

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

```

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

```

### Configuring the Layer 3 Datapath

#### Step-by-Step Procedure

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

```

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

```

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

```

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

```

### Configuring the VNF

#### Step-by-Step Procedure

1. Launch the VNF:

```

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

```

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

```

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

```

3. Pin a virtual CPU to a physical CPU:

```

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

```

4. Create host VLANs:

```

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

```

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

```

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

```

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

- Specify the memory allocation for the VNF:

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

### Configuring Cross-Connect

#### Step-by-Step Procedure

- 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

- Verifying the Control Plane Configuration on page 87
- Verifying the Data Plane Configuration on page 88

#### Verifying the Control Plane Configuration

**Purpose** Verify the control plane configuration:

- Action**
- Verify the VLANs configured.

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

Routing instance	VLAN name	Tag	Interfaces
default-switch	default	1	
default-switch	vlan100	100	ge-0/0/0.0* ge-0/0/1.0* sxe-0/0/0.0*
default-switch	vlan200	200	sxe-0/0/1.0* xe-0/0/12.0*
default-switch	vlan300	300	sxe-0/0/1.0* xe-0/0/13.0*

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

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

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

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

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

ID	Name	State	Liveliness
3	vnf-name	Running	alive

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

To view more details of the VNF:

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

#### Virtual Network Function Information

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

## Verifying the Data Plane Configuration

**Purpose** Verify the data plane configuration.

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

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

For example:

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

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



```

Active defects : None
PCS statistics                                     Seconds
  Bit errors                                     0
  Errored blocks                               0
Ethernet FEC statistics                           Errors
  FEC Corrected Errors                         0
  FEC Uncorrected Errors                       0
  FEC Corrected Errors Rate                     0
  FEC Uncorrected Errors Rate                   0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled

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

```

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

```

Physical interface: ge-1/0/2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 547
  Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Link-mode: Half-duplex,
Speed: 1000Mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

Remote fault: Online
Device flags   : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
CoS queues    : 8 supported, 8 maximum usable queues
Current address: 30:7c:5e:4c:78:1d, Hardware address: 30:7c:5e:4c:78:1d
Last flapped  : 2018-11-26 11:03:45 UTC (04:19:57 ago)
Input rate    : 0 bps (0 pps)
Output rate   : 0 bps (0 pps)
Active alarms : None
Active defects: None
PCS statistics                                     Seconds
  Bit errors                                     0
  Errored blocks                               0
Ethernet FEC statistics                           Errors
  FEC Corrected Errors                         0
  FEC Uncorrected Errors                       0
  FEC Corrected Errors Rate                     0
  FEC Uncorrected Errors Rate                   0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled

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

```

```

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

  Input packets : 0
  Output packets: 0
  Security: Zone: Null

```

- Verify the status of the OVS interfaces.

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

```
Network Name : ovs-sys-br
```

```

Interface : ovs-sys-br
Type : internal, Link type : Full-Duplex, MAC : 52:86:3c:df:9c:44
MTU : [], Link State :down, Admin State : down
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      1
  Tx-drops   :      1
  Tx-errors  :      0

```

```

Interface : dpdk0
Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:e2:b9:08
MTU : [], Link State :up, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      1
  Tx-drops   :      0
  Tx-errors  :      0

```

```

Interface : dpdk1
Type : dpdk, Link type : Full-Duplex, MAC : 02:09:c0:83:39:72
MTU : [], Link State :up, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0
  Tx-drops   :      0
  Tx-errors  :      0

```

```

Interface : l3_h_ge_1_0_0
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :up, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
  Rx-packets :      0
  Rx-drops   :      0
  Rx-errors  :      0
  Tx-packets :      0

```

```

Tx-drops    :      0
Tx-errors   :      0

Interface : l3_h_ge_1_0_2
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : [], Link State :down, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
Rx-packets  :      0
Rx-drops    :      0
Rx-errors   :      0
Tx-packets  :      0
Tx-drops    :      0
Tx-errors   :      0

Interface : vnf-name_eth2
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : 1500, Link State :down, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
Rx-packets  :      0
Rx-drops    :      0
Rx-errors   :      0
Tx-packets  :      0
Tx-drops    :      0
Tx-errors   :      0

Interface : vnf-name_eth3
Type : dpdkvhostuser, Link type : Full-Duplex, MAC : 00:00:00:00:00:00
MTU : 1500, Link State :down, Admin State : up
IPv4 : None, Netmask : None
IPv6 : None, IPv6 netmask : None
Rx-packets  :      0
Rx-drops    :      0
Rx-errors   :      0
Tx-packets  :      0
Tx-drops    :      0
Tx-errors   :      0

```

**Related Documentation** • *Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices*

## Example: Configuring Service Chaining for LAN Routing on NFX250 (NG) Devices

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

- [Requirements on page 91](#)
- [Overview on page 92](#)
- [Configuration on page 92](#)

### Requirements

This example uses an NFX250 (NG) device running Junos OS Release 19.1R1.

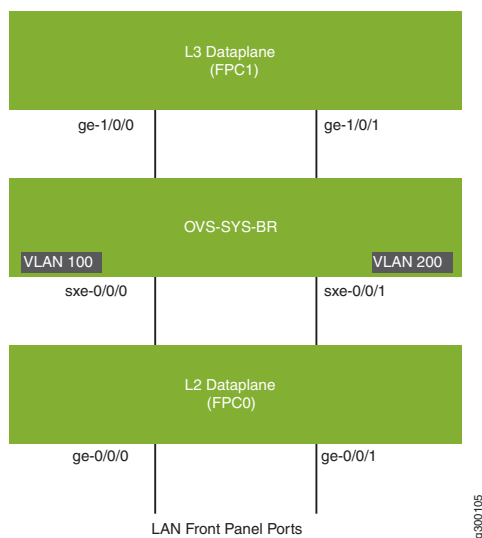
## Overview

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 11 on page 92](#).

*Figure 11: Service Chaining for LAN Routing*



## Configuration

- [Configuring the Layer 2 Datapath on page 92](#)
- [Configuring the Layer 3 Datapath on page 93](#)

### Configuring the Layer 2 Datapath

#### Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.  

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan200 vlan-id 200
```
2. Configure the LAN-side front panel ports and add them to the LAN-side VLAN.  

```
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@jcp# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan200
```
3. Configure the internal-facing interfaces as trunk ports and add them to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.

```

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

```

### Configuring the Layer 3 Datapath

#### Step-by-Step Procedure

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

```

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

```

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

```

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

```

#### Related Documentation

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

## Example: Configuring Service Chaining for LAN to WAN Routing on NFX250 (NG) Devices

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

- [Requirements on page 93](#)
- [Overview on page 93](#)
- [Configuration on page 94](#)
- [Verification on page 95](#)

### Requirements

This example uses an NFX250 (NG) device running Junos OS Release 19.1R1.

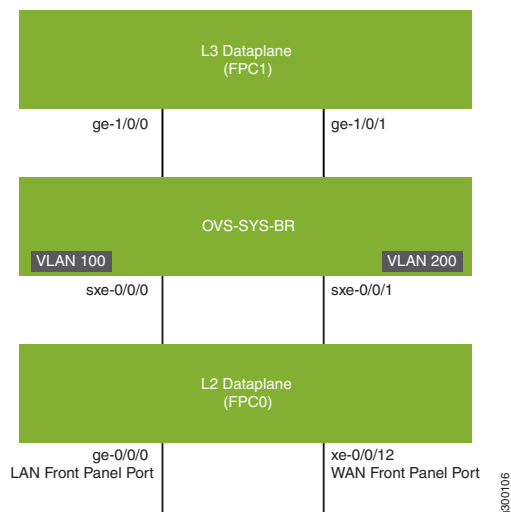
### Overview

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 12 on page 94](#).

Figure 12: Service Chaining for LAN to WAN Routing



## Configuration

- [Configuring the Layer 2 Datapath on page 94](#)
- [Configuring the Layer 3 Datapath on page 95](#)

### Configuring the Layer 2 Datapath

#### Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.  

```
user@host# set vlans vlan100 vlan-id 100
user@host# set vlans vlan200 vlan-id 200
```
2. Configure the LAN-side front panel ports and add them to the LAN-side and WAN-side VLANs.  

```
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members vlan200
```
3. Configure the internal-facing interface, sxe-0/0/0, as a trunk port and add it to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports as they must support traffic from multiple front panel ports and VLANs.  

```
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces sxe-0/0/0 unit 0 family ethernet-switching vlan members vlan100
```
4. Configure the internal-facing interface, sxe-0/0/1, as a trunk port and add it to the WAN-side VLAN.

```

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

```

### Configuring the Layer 3 Datapath

#### Step-by-Step Procedure

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

```

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

```

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

```

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

```

### Verification

- [Verifying the Status of the Interfaces on page 95](#)

### Verifying the Status of the Interfaces

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

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

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

For example:

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

```

Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 144, SNMP ifIndex: 518
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 1000mbps,
  BPDU Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues    : 8 supported, 8 maximum usable queues
  Current address: 00:00:5e:00:53:43, Hardware address: 00:00:5e:00:53:43
  Last flapped  : 2018-04-18 05:38:22 UTC (2d 10:07 ago)
  Statistics last cleared: Never
  Input rate    : 0 bps (0 pps)
  Output rate   : 0 bps (0 pps)
  Input errors: 0, Output errors: 0
  Active alarms : None
  Active defects: None
  PCS statistics

```

```
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
```



## CHAPTER 7

# Troubleshooting

- [Recovering the Root Password for NFX150 and NFX250 \(NG\) Devices on page 97](#)
- [Troubleshooting Interfaces on NFX Devices on page 100](#)

### Recovering the Root Password for NFX150 and NFX250 (NG) 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

- [Monitoring Interface Status and Traffic on NFX Series Devices on page 100](#)

### 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 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
```



## CHAPTER 8

# Operational Commands

- request vmhost cleanup
- request vmhost file-copy
- request vmhost halt
- request vmhost mode
- request vmhost power-off
- request vmhost reboot
- request vmhost software add
- show system visibility cpu
- show system visibility host
- show system visibility memory
- show system visibility network
- show system visibility vnf
- show vmhost connections
- show vmhost control-plane
- show vmhost crash
- show vmhost forwarding-options analyzer
- show vmhost memory
- show vmhost mode
- show vmhost status
- show vmhost storage
- show vmhost uptime
- show vmhost version
- show vmhost vlans

## request vmhost cleanup

---

<b>Syntax</b>	<code>request vmhost cleanup</code>
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Clean up temporary files, crash generated files, and log files located in the <code>/var/tmp</code> , <code>/var/crash</code> , and <code>/var/log</code> directories respectively on the host OS.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>vmhost</i></li></ul>
<b>Output Fields</b>	When you enter this command, you are provided feedback on the status of your request.



## request vmhost file-copy

<b>Syntax</b>	<code>request vmhost file-copy (crash log) from-jnode <i>host file-name</i> to-vjunos <i>host file-name</i></code>
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Copy crash files or log files from the host OS to Junos OS. You can use these files for analysis and debugging purposes.
<b>Options</b>	<ul style="list-style-type: none"> <li><code>crash</code>—Files in <code>/var/crash</code> on the host.</li> <li><code>from-jnode <i>filename</i></code>—Name of the host file to be copied.</li> <li><code>log</code>—Files in <code>/var/log</code> on the host.</li> <li><code>to-vjunos <i>filename</i></code>—Name of the Junos OS file to which the host file is copied.</li> </ul>
<b>Additional Information</b>	You can use the <code>show vmhost crash</code> and <code>show vmhost logs</code> commands to list or identify the files in the host OS to be copied to Junos OS.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><code>vmhost</code></li> </ul>
<b>List of Sample Output</b>	<a href="#">request vmhost file-copy on page 105</a>

## 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

<b>Syntax</b>	<code>request vmhost halt</code>
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Stop the host OS and Junos OS running on the device.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">request vmhost halt on page 106</a>

### 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


<b>Syntax</b>	<code>request vmhost mode [compute   hybrid   throughput]</code>
<b>Release Information</b>	Command introduced in Junos OS Release 19.1R1 for NFX150 and NFX250 (NG) devices.
<b>Description</b>	Select the operational mode of the device.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>vmhost</i></li></ul>
<b>List of Sample Output</b>	<a href="#">request vmhost mode compute on page 107</a>

## 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

<b>Syntax</b>	<code>request vmhost power-off</code>
<b>Release Information</b>	<p>Command introduced in Junos OS Release 18.1R1 for NFX150 devices.</p> <p>Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.</p>
	<p> <b>NOTE:</b> <code>request vmhost power-on</code> is not supported on NFX150 and NFX250 (NG) devices.</p>
<b>Description</b>	Shut down the Junos OS software and the host OS.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><code>vmhost</code></li> </ul>
<b>List of Sample Output</b>	<a href="#">request vmhost power-off on page 108</a>

## 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

<b>Syntax</b>	<code>request vmhost reboot</code>
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Reboot the Junos OS software and the host OS.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">request vmhost reboot on page 109</a>

### 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

<b>Syntax</b>	<code>request vmhost software add <i>package-name</i> &lt;in&gt;  &lt;no-validate&gt;  &lt;reboot&gt;  &lt;set&gt;  &lt;unlink&gt;  &lt;upgrade-to-model <i>model-number</i>&gt;</code>
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Install or upgrade the Junos OS and host software packages on the device.
<b>Options</b>	<ul style="list-style-type: none"> <li><code>in</code>—(Optional) Number of minutes to delay before the reboot operation.</li> <li><code>no-validate</code>—(Optional) When loading a software package or bundle with a different release, suppress the default behavior of the <b>validate</b> option.</li> <li><code>reboot</code>—(Optional) After adding the software package or bundle, reboot the system.</li> <li><code>set</code>—(Optional) List of URLs or pathnames corresponding to the software packages.</li> <li><code>unlink</code>—(Optional) Removes the software package after successful installation.</li> <li><code>upgrade-to-model</code>—(Optional) <i>model-number</i>—(Optional) Name of the model to upgrade to.</li> </ul>
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">request vmhost software add (NFX150) on page 110</a> <a href="#">request vmhost software add (NFX250 (NG)) on page 111</a>

## 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/1share/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 system visibility cpu

<b>Syntax</b>	show system visibility cpu
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Display details such as per CPU statistics, per CPU usage, and CPU pinning for a Junos OS platform.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show system visibility host on page 116</a></li> <li>• <a href="#">show system visibility memory on page 124</a></li> <li>• <a href="#">show system visibility network on page 126</a></li> <li>• <a href="#">show system visibility vnf on page 131</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show system visibility cpu (NFX150) on page 114</a> <a href="#">show system visibility cpu (NFX250 (NG)) on page 115</a>
<b>Output Fields</b>	Table 14 on page 113 lists the output fields for the <b>show system visibility cpu</b> command. Output fields are listed in the approximate order in which they appear.

Table 14: show system visibility cpu Output Fields

Field Name	Field Description
Fields for CPU Statistics	
CPU ID	The CPU ID
User Time	The amount of user time, in seconds.
System Time	The amount of system time, in seconds.
Idle Time	The amount of time spent in idle mode, in seconds.
Nice Time	The amount of spent nice time, in seconds.
I/O Wait Time	The amount of time spent waiting for input/output (I/O) operations, in seconds.
Interrupt Service Time	The amount of interrupt service time, in seconds.
Service Time	The amount of service time, in seconds.

Table 14: show system visibility cpu Output Fields (continued)

Field Name	Field Description
Fields for CPU Usages	
CPU ID	The CPU ID
CPU Usage	The percentage of CPU used.
Fields for CPU Pinning Information	
Virtual Machine	The name of the virtual machine.
vCPU	The ID of virtual CPUs used by the virtual machine.
CPU	The ID of CPUs used by the virtual machine.
System Component	The name of the system component.
CPUs	The ID of CPUs used by the system component.

## Sample Output

### show system visibility cpu (NFX150)

```

user@host> show system visibility cpu
CPU Statistics (Time in sec)
-----
CPU Id User Time System Time Idle Time Nice Time IOWait Time Intr. Service Time
-----
0      26583    40107    105816    0        102        0
1      53183    64078    56959    0         0         0
2       72      67     171189    0         1         0
3       0       96     171241    0         0         0

CPU Usages
-----
CPU Id CPU Usage
-----
0      36.39999999999999
1      66.70000000000003
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&gt; 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

<b>Syntax</b>	show system visibility host
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	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.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show system visibility cpu on page 113</a></li> <li>• <a href="#">show system visibility memory on page 124</a></li> <li>• <a href="#">show system visibility network on page 126</a></li> <li>• <a href="#">show system visibility vnf on page 131</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show system visibility host (NFX150) on page 118</a> <a href="#">show system visibility host (NFX250 (NG)) on page 121</a>
<b>Output Fields</b>	Table 15 on page 116 lists the output fields for the <b>show system visibility host</b> command. Output fields are listed in the approximate order in which they appear.

*Table 15: show system visibility host Output Fields*

Field Name	Field Description
Field for Host Uptime	
<b>Uptime</b>	The time the host has been operational.
Fields for Host Tasks	
<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.
Fields for Host CPU Information	

Table 15: show system visibility host Output Fields (continued)

Field Name	Field Description
User Time	The amount of user time, in seconds.
System Time	The amount of system time, in seconds.
Idle Time	The amount of time spent in idle mode, in seconds.
Nice Time	The amount of spent nice time, in seconds.
I/O Wait Time	The amount of time spent waiting for input/output (I/O) operations, in seconds.
Interrupt Service Time	The amount of interrupt service time, in seconds.
Fields for Host Disk Partitions	
Device	The device path.
Mount Point	The mount point of the device path.
File System	The file system type.
Options	Options available for the device path.
Fields for Host Disk Usage Information	
Total	The total amount of disk usage space, in mebibytes (MiB).
Used	The amount of used disk usage space, in mebibytes (MiB).
Free	The amount of free disk usage space, in mebibytes (MiB).
Percentage Used	The percentage of used disk space.
Fields for Host Disk I/O Information	
Read Count	The number of times the disk has been read.
Write Count	The number of times a write operation has happened on the disk.
Read Bytes	The number of bytes used in read operations on the disk.
Write Bytes	The number of bytes used in write operations on the disk.
Read Time	The amount of time the disk has been read, in milliseconds.
Write Time	The amount of time write operations have been performed on the disk, in milliseconds.
Fields for List of Host Interfaces	
Interfaces	The name of the interface.

Table 15: show system visibility host Output Fields (continued)

Field Name	Field Description
State	The state of the Host Interface.
MAC	The MAC address of the interface.
Fields for List of Host Port Statistics	
Interface	The name of the interface.
Bytes Sent	The number of bytes sent.
Bytes Received	The number of bytes received.
Packets Sent	The number of packets sent.
Packets Received	The number of packets received.
Errors In	The number of errors in.
Errors Out	The number of errors out.
Drops In	The number of drops in.
Drops Out	The number of drops out.

## Sample Output

### show system visibility host (NFX150)

```

user@host> show system visibility host

Host Uptime
-----
Uptime: 1 day 23:19:41.21000

Host Tasks
-----
Total:    187
Running:  3
Sleeping: 179
Stopped:  0
Zombie:   5

Host CPU Information (Time in sec)
-----
User Time:      79359
System Time:    0
Idle Time:      502215
I/O Wait Time:  103
Nice Time:      103724
Interrupt Service Time: 0

```

## Host Disk Partitions

Device	Mount Point	File System	Options
/dev/sda2	/	ext4	rw,relatime,i_version,data=ordered
/dev/sda1	/boot/efi	vfat	rw,noatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro
/dev/sda7	/config	ext4	rw,noatime,data=ordered
/dev/sda8	/var/log	ext4	rw,noatime,data=ordered
/dev/sda9	/mnt/.share	ext4	rw,noatime,discard,data=ordered
/dev/sda5	/junos	ext4	rw,noatime,discard,data=ordered
/dev/loop0	/var/tmp	ext4	rw,relatime,data=ordered
/dev/loop1	/mnt/.share/lshare/jnpr/jlog	ext4	rw,relatime,data=ordered
/dev/loop0	/mnt/.share/lshare/jnpr/jtmp	ext4	rw,relatime,data=ordered

## Host Disk Usage Information

```

Total (MiB): 1469
Used (MiB): 948
Free (MiB): 429
Percentage Used: 64.5

```

## Host Disk I/O Information

```

Read Count: 187083
Write Count: 256206
Read Bytes: 2290787328
Write Bytes: 3331667456
Read Time: 33977
Write Time: 258864

```

## Host Interfaces

Interface	State	MAC
heth-0-1	active	00:00:5e:00:53:8e
heth-0-0	active	00:00:5e:00:53:8d
heth-0-3	active	00:00:5e:00:53:90
heth-0-2	active	00:00:5e:00:53:8f
heth-0-5	inactive	00:00:5e:00:53:92
heth-0-4	inactive	00:00:5e:00:53:91
ctrlbr0	active	00:00:5e:00:53:10
docker0	inactive	00:00:5e:00:53:8c
eth0br	active	00:00:5e:00:53:00
eth1br	inactive	00:00:5e:00:53:67
l3_h_ge_1_0_0	active	00:00:5e:00:53:6d
l3_h_ltectrl	active	00:00:5e:00:53:f1
l3_h_ltedata	active	00:00:5e:00:53:91
lo	inactive	00:00:00:00:00:00
lte_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 00:00:00:00:00:00
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 Out Drops	Bytes Rcvd In Drops	Packets Sent	Packets Rcvd	Errors In	Errors 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					
13_h_ltedata	1648	648	14	8	0	0
0	0					
sit0	0	0	0	0	0	0
0	0					
flowd_h_mgmt	391536979	448871585	5975703	5507199	0	0
0	0					



virbr1	29553905	8096581	137792	128808	0	0
0	0					
virbr0	46365	48232	467	540	0	0
0	0					
l3_h_ltectrl	12570778	818395	22762	22718	0	0
0	0					
jdm-hbme1	4474379	55866	85622	537	0	0
0	0					
jdm-hbme2	813479	1526643	7992	15288	0	0
0	0					
eth0br	0	595875398	0	4835907	0	0
222	0					
ctrlbr0	408483097	256713674	3800585	4571275	0	0
0	0					
heth-0-1	0	5368334	0	89330	0	0
0	0					
heth-0-0	0	5366462	0	89349	0	0
0	0					
heth-0-3	0	5367002	0	89358	0	0
0	0					
heth-0-2	0	5365262	0	89329	0	0
0	0					
heth-0-5	0	0	0	0	0	0
0	0					
heth-0-4	0	0	0	0	0	0
0	0					

### show system visibility host (NFX250 (NG))

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

```
Host Uptime
```

```
-----
```

```
Uptime: 3 days 3:47:05.09000
```

```
Host Tasks
```

```
-----
```

```
Total: 198
```

```
Running: 1
```

```
Sleeping: 194
```

```
Stopped: 0
```

```
Zombie: 3
```

```
Host CPU Information (Time in sec)
```

```
-----
```

```
User Time: 574351
```

```
System Time: 0
```

```
Idle Time: 2692218
```

```
I/O Wait Time: 216
```

```
Nice Time: 4609
```

```
Interrupt Service Time: 0
```

```
Host Disk Partitions
```

Device	Mount Point	File System	Options
/dev/sda2	/	ext4	
rw,relatime,i_version,data=ordered			
/dev/sda1	/boot/efi	vfat	

```

rw,noatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro
/dev/sda7                               /config                ext4
rw,noatime,data=ordered
/dev/sda8                               /var/log                ext4
rw,noatime,data=ordered
/dev/sda9                               /mnt/.share             ext4
rw,noatime,discard,data=ordered
/dev/sda5                               /junos                  ext4
rw,noatime,discard,data=ordered
/dev/loop0                             /var/tmp                ext4
rw,relatime,data=ordered

```

#### Host Disk Usage Information

```

-----
Total (MiB):    1469
Used  (MiB):    906
Free  (MiB):    470
Percentage Used: 61.7

```

#### Host Disk I/O Information

```

-----
Read Count: 245805
Write Count: 333782
Read Bytes: 2967304704
Write Bytes: 6147921408
Read Time: 34906
Write Time: 448918

```

#### Host Interfaces

```

-----
Interface      State      MAC
-----
hsxe0          active     30:7c:5e:4c:78:44
hsxe1          active     30:7c:5e:4c:78:45
ctrlbr0        active     02:00:00:00:00:10
docker0        inactive   02:42:f9:e7:08:5f
eth0br         active     4c:96:14:00:00:00
eth1br         inactive   66:7e:98:6c:9d:a7
l3_h_ge_1_0_0  active     ca:6b:5a:fe:39:2c
lo             inactive   00:00:00:00:00:00
sit0           inactive   00:00:00:00
virbr0         active     30:7c:5e:4c:78:43
virbr1         active     be:51:f7:ac:03:1b

```

#### Host Port Statistics

```

-----
Interface Bytes Sent  Bytes Rcvd  Packets Sent Packets Rcvd Errors In Errors
Out Drops In Drops Out
-----
l3_h_ge_1_0_0 0 648 0 8 0 0
0 0
ovs-sys-br 0 0 0 0 0 0
0 0
vnet0 2573491477 117345734 2448205 1790887 0 0
0 0
vnet1 670930985 585788796 7585078 8400542 0 0
0 0
vnet2 454043208 224389433 2873376 416585 0 0
0 0

```

vnet3	7129616	9814	137213	231	0	0
0	0					
eth0	224389433	464747548	416585	2889060	0	0
9829	0					
lo	61305	61305	920	920	0	0
0	0					
virbr1	2475291351	90762062	1008399	1774468	0	0
0	0					
irb	0	0	0	0	0	0
0	0					
hsxe1	0	0	0	0	0	0
0	0					
hsxe0	0	0	0	0	0	0
0	0					
docker0	0	0	0	0	0	0
0	0					
dcapi-tap	0	0	0	0	0	0
0	0					
sit0	0	0	0	0	0	0
0	0					
flowd_h_mgmt	387545386	426690199	5662328	5294853	0	0
0	0					
virbr0-nic	0	0	0	0	0	0
0	0					
virbr0	3021873	1067179	4573	6153	0	0
0	0					
jdm-hbme1	1785562	33378	34145	404	0	0
0	0					
jdm-hbme2	41904	72344	321	323	0	0
0	0					
eth0br	0	401858893	0	2755416	0	0
226	0					
ctrlbr0	243770080	159923150	2283092	2738720	0	0
0	0					
eth1br	0	0	0	0	0	0
0	0					
ovs-netdev	0	0	0	0	0	0
0	0					

## show system visibility memory

<b>Syntax</b>	show system visibility memory
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices.
<b>Description</b>	Display the details about virtual memory and shared memory for a Junos OS platform.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show system visibility cpu on page 113</a></li> <li>• <a href="#">show system visibility host on page 116</a></li> <li>• <a href="#">show system visibility network on page 126</a></li> <li>• <a href="#">show system visibility vnf on page 131</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show system visibility memory on page 124</a>
<b>Output Fields</b>	Table 16 on page 124 lists the output fields for the <b>show system visibility memory</b> command. Output fields are listed in the approximate order in which they appear.

Table 16: show system visibility memory Output Fields

Field Name	Field Description
<b>Fields for Memory Information—Virtual Memory</b>	
<b>Total</b>	The total amount of available virtual memory, in kibibytes (KiBs).
<b>Used</b>	The total amount of used virtual memory, in kibibytes (KiBs).
<b>Available</b>	The total amount of available virtual memory, in kibibytes (KiBs).
<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—Memory Limits</b>	
<b>User VNFs Total Memory Limit</b>	The total memory limit for user VNFs, in kibibytes (KiBs).

## Sample Output

### show system visibility memory

```
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

## Memory Limits:

-----  
User VNFs Total Memory Limit (KiB): 5242880

## show system visibility network

<b>Syntax</b>	show system visibility network
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	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.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show system visibility cpu on page 113</a></li> <li>• <a href="#">show system visibility host on page 116</a></li> <li>• <a href="#">show system visibility memory on page 124</a></li> <li>• <a href="#">show system visibility vnf on page 131</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show system visibility network (NFX150) on page 127</a> <a href="#">show system visibility network (NFX250 (NG)) on page 129</a>
<b>Output Fields</b>	<a href="#">Table 17 on page 126</a> lists the output fields for the <b>show system visibility network</b> command. Output fields are listed in the approximate order in which they appear.

*Table 17: show system visibility network Output Fields*

Field Name	Field Description
Fields for List of VNF MAC Addresses	
VNF	The name of the VNF.
MAC	The MAC address of the VNF.
Fields for List of VNF Internal IP Addresses	
VNF	The name of the VNF.
IP	The IP address of the VNF.
Fields for List of VNF Virtual Functions	
VNF	The name of the VNF.
PF	The names of the Physical Functions available.

Table 17: show system visibility network Output Fields (continued)

Field Name	Field Description
VF	The names of the Virtual Functions available for each Physical Function.
Fields for List of Free Virtual Functions	
PF	The names of the Physical Functions available.
VF	The names of the Virtual Functions available for each Physical Function.
Fields for List of VNF Interfaces	
VNF	The name of the VNF.
Interface	The name of the interface.
Type	The type of interface.
Source	The connectivity source.
Model	The connectivity model.
MAC	The MAC address of the VNF.

## Sample Output

### show system visibility network (NFX150)

```

user@host> show system visibility network

VNF MAC Addresses
-----
VNF                                     MAC
-----
centos1_ethdef0                        00:00:5E:00:53:9E
centos1_ethdef1                        00:00:5E:00:53:9F
centos1_eth2                           00:00:5E:00:53:A0
centos1_eth3                           00:00:5E:00:53:A1
centos2_ethdef0                        00:00:5E:00:53:A2
centos2_ethdef1                        00:00:5E:00:53:A3
centos2_eth2                           00:00:5E:00:53:A4
centos2_eth3                           00:00:5E:00:53:A5

VNF Internal IP Addresses
-----
VNF                                     IP
-----
centos1                                192.0.2.103
centos2                                192.0.2.102

VNF Virtual Functions
-----
VNF                                     PF      VF
-----

```

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

<b>Syntax</b>	<code>show system visibility vnf <i>vnf name</i></code>
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	<p>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.</p> <p>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.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show system visibility cpu on page 113</a></li> <li>• <a href="#">show system visibility host on page 116</a></li> <li>• <a href="#">show system visibility memory on page 124</a></li> <li>• <a href="#">show system visibility network on page 126</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show system visibility vnf on page 134</a>
<b>Output Fields</b>	<a href="#">Table 18 on page 131</a> lists the output fields for the <b>show system visibility vnf</b> command. Output fields are listed in the approximate order in which they appear.

*Table 18: show system visibility vnf Output Fields*

Field Name	Field Description
Fields for List of VNFs	
ID	ID of the VNF.
Name	Name of the VNF.
State	State of the VNF.
Fields for VNF Memory Usage	
Name	Name of the VNF.
Maximum Memory	The maximum amount of memory, in kibibytes (KiBs).

Table 18: show system visibility vnf Output Fields (continued)

Field Name	Field Description
Used Memory	The total amount of used memory, in kibibytes (KiBs).
Used 1G Hugepages	The total number of 1G hugepages used.
Used 2M Hugepages	The total number of 2M hugepages used.
Fields for VNF CPU Stats	
Name	Name of the VNF.
CPU Time	The total CPU time, in seconds.
System Time	The amount of system CPU time, in seconds.
User Time	The amount of user CPU time, in seconds.
Fields for List of VNF MAC Addresses	
VNF	Names of the VNFs.
MAC	MAC addresses of the VNFs.
Fields for List of VNF Internal IP Addresses	
VNF	Names of the VNFs.
IP	Internal IP addresses of the VNFs.
Fields for List of Virtual Functions per VNF	
VNF	Names of the VNFs.
PF	The names of the Physical Functions available.
VF	The names of the Virtual Functions available for each Physical Function.
Fields for the VNF Interfaces	
VNF	The name of the VNF.
Interface	The name of the interface.
Type	The type of interface.
Source	The connectivity source.
Model	The connectivity model.
MAC	The MAC address of the VNF.

Table 18: show system visibility vnf Output Fields (continued)

Field Name	Field Description
Fields for List of VNF Disk Information	
VNF	The name of the VNF.
Disk	The name of the disk.
File	The path to the disk.
Fields for List of VNF Disk Usage	
VNF	The name of the VNF.
Disk	The name of the disk.
Read Requests	The number of times a read operation has happened on the disk.
Bytes Read	The number of read bytes on the disk.
Write Requests	The number of times a write operation has happened on the disk.
Bytes Written	The number of bytes written on the disk.
Fields for List of VNF Port Statistics	
VNF	The name of the VNF.
Port	The name of the port.
Rcvd Bytes	The number of bytes received.
Rcvd Packets	The number of packets received.
Rcvd Error	The number of errors received.
Rcvd Drop	The number of drops received.
Trxd Bytes	The number of bytes transferred.
Trxd Packets	The number of packets transferred.
Trxd Error	The number of errors transferred.
Trxd Drop	The number of drops transferred.

## Sample Output

### show system visibility vnf

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

#### List of VNFs

ID	Name	State
4	vnf-name1	Running

#### VNF Memory Usage

Name	Maximum Memory (KiB)	Used Memory (KiB)
Used 1G Hugepages	Used 2M Hugepages	
vnf-name1	104857	104857
0	256	

#### VNF CPU Statistics (Time in ms)

Name	CPU Time	System Time	User Time
vnf-name1	617598	257020	53510

#### VNF MAC Addresses

VNF	MAC
vnf-name1_ethdef0	00:00:5e:00:53:15
vnf-name1_ethdef1	00:00:5e:00:53:16
vnf-name1_eth2	00:00:5e:00:53:17

#### VNF Internal IP Addresses

VNF	IP
vnf-name1	192.168.1.100

#### VNF Virtual Functions

VNF	PF	VF
sxe0vf9_vfdef0	hsxe0	0000:03:12:2
sxe0vf10_vfdef0	hsxe0	0000:03:12:4
sxe0vf11_vfdef0	hsxe0	0000:03:12:6
sxe0vf12_vfdef0	hsxe0	0000:03:13:0
sxe0vf13_vfdef0	hsxe0	0000:03:13:2
sxe1vf13_vfdef0	hsxe1	0000:03:13:3
sxe1vf14_vfdef0	hsxe1	0000:03:13:5

#### VNF Interfaces

VNF	Interface	Type	Source	Model	MAC
vnf-name1	vnet5	network	default	virtio	00:00:5e:00:53:15
vnf-name1	vnet6	bridge	eth0br	virtio	00:00:5e:00:53:16
vnf-name1	vhostuser	--		virtio	00:00:5e:00:53:17

## VNF Disk Information

VNF	Disk	File
vnf-name1	vda	/var/third-party/images/media-vsrx-vmdisk-15.1X49-D40.6.qcow2.md5

## VNF Disk Usage

VNF	Disk	Read Req	Read Bytes	Write Req	Write Bytes
vnf-name1	vda	0	0	0	0

## VNF Port Statistics

VNF	Port	Rcvd Bytes	Rcvd Packets	Rcvd Error	Rcvd Drop
Trxd Bytes	Trxd Packets	Trxd Error	Trxd Drop		
vnf-name1	vnet5	0	0	0	252654
0	0	0	0	0	
vnf-name1	vnet6	0	0	0	8893085
0	0	0	0	0	

## show vmhost connections

<b>Syntax</b>	show vmhost connections
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Display the details for the cross-connect connections. The NFX150 and NFX250 (NG) supports VLAN PUSH, POP, and SWAP operations.
<b>Options</b>	<p><b>name</b>—Display the details of a specific connection.</p> <p><b>down</b>—Display the details of connections that are not operational.</p> <p><b>up</b>—Display the details of connections that are operational.</p> <p><b>up-down</b>—Display the details of both operational and non-operational connections.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">vmhost</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show vmhost connections on page 136</a>
<b>Output Fields</b>	Table 19 on page 136 lists the output fields for the <b>show vmhost connections</b> command. Output fields are listed in the approximate order in which they appear.

Table 19: show vmhost connections Output Fields

Field Name	Field Description
Connection	Displays the type of the cross-connect.
Function	Displays the name of the virtual network function.
Interface	Specifies an interface on which the connection is established.
Status	Displays the status of the connection.

## Sample Output

### show vmhost connections

```
user@host> show vmhost connections
```



Connection	Function	Interface	Vlan	Status
phy_cc	system centos1	sxe0 eth2	200 500	up
push_pop_cc	centos1 centos2	eth2 eth3	none none	down
swap_cc	centos1 centos2	eth2 eth2	300 400	up
vlan_cc	centos1 centos2	eth2 eth2	100 100	up

## 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 138](#)

### 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

<b>Syntax</b>	show vmhost crash
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Display host OS crash information.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show vmhost crash on page 139</a>

### 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

<b>Syntax</b>	<b>show vmhost forwarding-options analyzer <i>analyzer-name</i></b>
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Displays information about the VNF analyzers that are configured for port mirroring on a Junos OS platform.
<b>Options</b>	<b><i>analyzer-name</i></b> —Displays the details of a specific analyzer on the device.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show vmhost forwarding-options analyzer on page 140</a>
<b>Output Fields</b>	<a href="#">Table 20 on page 140</a> lists the output fields for the <b>show vmhost forwarding-options analyzer</b> command. Output fields are listed in the approximate order in which they appear.

*Table 20: 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 142](#)

**Output Fields**

### Sample Output

`show vmhost memory`

```
user@host> show vmhost memory
```

```
Memory Controller Information
```

```
-----
```

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

## show vmhost mode

<b>Syntax</b>	show vmhost mode
<b>Release Information</b>	Command introduced in Junos OS Release 19.1R1 for NFX150 and NFX250 (NG) devices.
<b>Description</b>	Display the CPU and memory allocations for various components.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show vmhost mode (Throuput mode) on page 143</a> <a href="#">show vmhost mode (Hybrid mode) on page 144</a> <a href="#">show vmhost mode (Compute mode) on page 145</a>

## Sample Output

### show vmhost mode (Throuput 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	0	0
Juniper Device Manager	0	0
LTE	0	-
NFV Backplane Control Path	0	0
NFV Backplane Data Path	1	1
Layer 2 Control Path	0	0
Layer 2 Data Path	2	2
Layer 3 Control Path	0	0
Layer 3 Data Path	3	3

#### 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

## show vmhost status

**Syntax** show vmhost status

**Release Information** Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

**Description** Display the virtualization status and status of all the CPUs.

**Required Privilege Level** view

**List of Sample Output** [show vmhost status on page 146](#)

### Sample Output

show vmhost status

```
user@host> show vmhost status
```

```
Virtualization status :
```

```
-----
kvm_status      : ok
libvirt_status  : ok
qemu_status     : ok
```

```
CPU Status [Since Boot Time]:
```

```
-----
CPU      %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice
%idle

Load Avg : 4.04  0.00  4.74  0.01  0.00  0.01  0.00  0.30  0.00
90.90
cpu0      : 8.26  0.00  15.91  0.06  0.00  0.06  0.00  2.47  0.00
73.23
cpu1      : 24.73 0.00  22.95  0.00  0.00  0.00  0.00  0.00  0.00
52.32
cpu2      : 0.00  0.00  0.01  0.00  0.00  0.00  0.00  0.02  0.00
99.97
cpu3      : 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
100.00
cpu4      : 0.00  0.00  0.00  0.00  0.00  0.02  0.00  0.00  0.00
99.98
cpu5      : 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
100.00
cpu6      : 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
100.00
cpu7      : 0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
100.00

Device: tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
-----
sda      2.15    7.60          30.04        4057951    16046703
```



## show vmhost storage

**Syntax** `show vmhost storage`

**Release Information** Command introduced in Junos OS Release 18.1R1 for NFX150 devices.  
Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.

**Description** Display the vmhost storage information.

**Required Privilege Level** view

**Related Documentation**

- vmhost*

**List of Sample Output** [show vmhost storage on page 148](#)

### 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

<b>Syntax</b>	show vmhost uptime
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Display the current time and information such as how long the host OS has been running, number of users, and average load.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <i>vmhost</i></li></ul>
<b>List of Sample Output</b>	<a href="#">show vmhost uptime on page 150</a>

## Sample Output

### show vmhost uptime

```
user@host> show vmhost uptime
```

```
Vmhost Current time: 2018-04-09 09:15:28+00:00
Vmhost Uptime:
    09:15:28 up 6 days, 4:42, 0 users, load average: 0.38, 0.48, 0.45
```

## show vmhost version

<b>Syntax</b>	show vmhost version
<b>Release Information</b>	Command introduced in Junos OS Release 18.1R1 for NFX150 devices. Command introduced in Junos OS Release 18.4R1 for NFX250 (NG) devices.
<b>Description</b>	Display host version information including Linux host kernel version and host software version.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show vmhost connections (NFX150) on page 151</a> <a href="#">show vmhost connections (NFX250 (NG)) on page 151</a>

### 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

<b>Syntax</b>	show vmhost vlans
<b>Release Information</b>	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.
<b>Description</b>	Display details about the vmhost VLANs.
<b>Options</b>	<p><b>vlan-name</b>—Display information for a specified VLAN.</p> <p><b>brief   detail   extensive</b> —Display the specified level of output.</p> <p><b>instance</b>—Display information for a specified instance.</p> <p><b>interface</b>—Name of interface for which the table is displayed.</p> <p><b>logical-system</b>—Name of logical system.</p>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>vmhost</i></li> </ul>
<b>List of Sample Output</b>	<a href="#">show vmhost vlans on page 153</a>
<b>Output Fields</b>	<a href="#">Table 21 on page 152</a> describes the output fields for the <a href="#">show vmhost forwarding-options analyzers</a> <b>show vmhost vlans</b> command. Output fields are listed in the approximate order in which they appear.

Table 21: show vmhost vlans Output Fields

Field Name	Field Description
vlan-name	Display information for a specified VLAN
brief	Display brief output
detail	Display detailed output
extensive	Display extensive output
instance	Display information for a specified instance
interface	Name of interface for which to display table
logical-system	Name of logical system



## Sample Output

show vmhost vlans

```
root@host> show vmhost vlans
```

Routing instance	VLAN name	Tag	Interfaces
vmhost	test-1	56	centos1_eth2.0
----			

