



Getting Started with NFX150 Network Services Platform



Modified: 2018-12-12

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

Getting Started with NFX150 Network Services Platform
Copyright © 2018 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 |
| | Opening a Case with JTAC | xvi |
| Chapter 1 | Overview | 17 |
| | NFX150 System Overview | 17 |
| | Overview | 17 |
| | NFX150 Models | 18 |
| | Junos OS Releases Supported on NFX Series Hardware | 20 |
| | NFX Product Compatibility | 21 |
| | Hardware Compatibility | 21 |
| | Software Version Compatibility | 21 |
| | Benefits and Uses of NFX150 | 23 |
| | NFX150 Feature Overview | 23 |
| | Software Architecture | 23 |
| | Interfaces | 25 |
| | Physical Interfaces | 25 |
| | Virtual Interfaces | 26 |
| | LTE Interface | 26 |
| | Interface Mapping | 26 |
| | Supported Features | 28 |
| | Licensing | 29 |
| Chapter 2 | Initial Configuration | 31 |
| | Initial Configuration on NFX150 Devices | 31 |
| | Factory-Default Settings | 31 |
| | Enabling Basic Connectivity | 32 |
| | Establish the Connection | 33 |
| | Provisioning the NFX150 Device Using Network Activator | 34 |
| | Understanding Network Activator | 34 |
| | Provisioning the NFX150 Device | 35 |
| | Configuring the LTE Module on NFX Devices | 36 |
| | Configuring the LTE Module for Primary Mode | 37 |
| | Configuring the LTE Module for Dial-on-Demand Mode | 38 |

| | | |
|------------------|--|-----------|
| | Configuring the LTE Module for Backup Mode | 40 |
| | Upgrading the Modem Firmware on NFX Devices Through Over-the-Air (OTA) | 42 |
| Chapter 3 | Configuring Interfaces | 45 |
| | Mapping Interfaces on NFX150 Devices | 45 |
| | Mapping Physical Interfaces to Virtual Interfaces on NFX150 Devices | 45 |
| | Mapping Physical Ports to VNF Interfaces Through SR-IOV | 46 |
| | Mapping Layer 3 Dataplane Interfaces to OVS | 46 |
| | Configuring the In-Band Management Interface for NFX150 | 46 |
| | ADSL2 and ADSL2+ Interfaces on NFX150 Devices | 47 |
| | ADSL Interface Overview | 47 |
| | ADSL2 and ADSL2+ | 48 |
| | Configuring ADSL SFP Interface Using VLANs on NFX150 Network Services Platform | 48 |
| | Configuring ADSL SFP Interface Without Using VLANs on NFX150 Network Services Platform | 50 |
| | VDSL2 Interfaces on NFX150 Devices | 51 |
| | VDSL Interface Overview | 51 |
| | VDSL2 Vectoring Overview | 52 |
| | VDSL2 Network Deployment Topology | 52 |
| | VDSL2 Interface Support on NFX Series Devices | 53 |
| | VDSL2 Interface Compatibility with ADSL Interfaces | 54 |
| | VDSL2 Interfaces Supported Profiles | 54 |
| | VDSL2 Interfaces Supported Features | 54 |
| | Configuring VDSL SFP Interface Using VLANs on NFX150 Network Services Platform | 55 |
| | Configuring VDSL SFP Interface Without Using VLANs on NFX150 Network Services Platform | 56 |
| Chapter 4 | Configuring Security | 59 |
| | IP Security on NFX Devices | 59 |
| | Overview | 59 |
| | Configuring Security | 61 |
| | Configuring Interfaces | 61 |
| | Configuring Routing Options | 62 |
| | Configuring Security IKE | 62 |
| | Configuring Security IPsec | 64 |
| | Configuring Security Policies | 66 |
| | Configuring Security Zones | 67 |
| | UTM on NFX Devices | 67 |
| | Application Security on NFX Devices | 68 |
| | Intrusion Detection and Prevention on NFX Devices | 69 |
| | Integrated User Firewall Support on NFX Devices | 69 |
| Chapter 5 | Configuring VNFs | 71 |
| | Configuring VNFs on NFX150 Devices | 71 |
| | VNF Configuration Overview | 71 |
| | Loading the VNF Image | 72 |
| | Preparing the Bootstrap Configuration | 72 |

| | | |
|------------------|--|------------|
| | Launching a VNF | 72 |
| | Verify that the VNF Launched Successfully | 73 |
| | Allocating a CPU for a VNF | 73 |
| | Allocating Memory for a VNF | 74 |
| | Configuring VNF Storage Devices | 74 |
| | Configuring VNF Interfaces and VLANs | 75 |
| | Launching a VNF Using an XML Descriptor File | 77 |
| | Managing VNFs on NFX150 Devices | 77 |
| | Managing VNF States | 78 |
| | Managing VNF MAC Addresses | 78 |
| | Managing MTU | 79 |
| | Accessing a VNF from JCP | 80 |
| | Viewing List of VNFs | 80 |
| | Displaying the VNF Details | 80 |
| | Deleting a VNF | 80 |
| | Configuring Analyzer VNF and Port-mirroring | 81 |
| Chapter 6 | Configuring Service Chaining | 83 |
| | Service Chaining on NFX150 Devices | 83 |
| | Understanding Service Chaining | 83 |
| | Configuring Service Chaining Using VLANs | 84 |
| | Configuring Service Chaining Using DHCP Services on VLANs | 84 |
| | Example: Configuring Service Chaining Using VLANs on NFX150 Network Services Platform | 85 |
| | Example: Configuring Service Chaining Using SR-IOV on NFX150 Network Services Platform | 89 |
| | Example: Configuring Service Chaining Using a Custom Bridge | 94 |
| | Example: Configuring Service Chaining for LAN-WAN Routing | 100 |
| | Example: Configuring Cross Connect on NFX150 Devices | 103 |
| | Example: Configuring Service Chaining for LAN Routing | 111 |
| | Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices | 113 |
| Chapter 7 | Troubleshooting | 121 |
| | Recovering the Root Password for NFX150 | 121 |
| | Troubleshooting Interfaces on NFX150 Devices | 124 |
| | Monitoring Interface Status and Traffic on NFX Devices | 124 |
| Chapter 8 | Configuration Statements | 125 |
| | cross-connect | 126 |
| | forwarding-options | 127 |
| | interfaces | 128 |
| | phone-home | 129 |
| | security | 131 |
| | virtualization-options | 134 |
| | vlans | 135 |
| | vmhost | 136 |

| | | |
|------------------|---|------------|
| Chapter 9 | Operational Commands | 139 |
| | request vmhost cleanup | 140 |
| | request vmhost file-copy | 141 |
| | request vmhost halt | 142 |
| | request vmhost power-off | 143 |
| | request vmhost reboot | 144 |
| | request vmhost software add | 145 |
| | show system visibility cpu | 146 |
| | show system visibility host | 148 |
| | show system visibility memory | 154 |
| | show system visibility network | 156 |
| | show system visibility vnf | 160 |
| | show vmhost connections | 165 |
| | show vmhost control-plane | 167 |
| | show vmhost crash | 168 |
| | show vmhost forwarding-options analyzer | 169 |
| | show vmhost memory | 170 |
| | show vmhost status | 171 |
| | show vmhost storage | 172 |
| | show vmhost uptime | 174 |
| | show vmhost version | 175 |
| | show vmhost vlans | 176 |

List of Figures

| | | |
|------------------|--|-----------|
| Chapter 1 | Overview | 17 |
| | Figure 1: NFX150 Compact Model (Without LTE Support) | 18 |
| | Figure 2: NFX150 Rack-Mount Model | 18 |
| | Figure 3: NFX150 Architecture | 24 |
| Chapter 2 | Initial Configuration | 31 |
| | Figure 4: Connecting the Interfaces on an NFX150-S1 Device | 34 |
| | Figure 5: Workflow for Initial Provisioning of the NFX150 | 35 |
| Chapter 3 | Configuring Interfaces | 45 |
| | Figure 6: Typical VDSL2 End-to-End Connectivity and Topology Diagram | 53 |
| | Figure 7: Backward-Compatible ADSL Topology (ATM DSLAM) | 53 |
| Chapter 4 | Configuring Security | 59 |
| | Figure 8: Scenario for Integrated User Firewall | 70 |
| Chapter 6 | Configuring Service Chaining | 83 |
| | Figure 9: Service Chaining Using VLANs | 86 |
| | Figure 10: Service Chaining Using SR-IOV—Device Infrastructure | 90 |
| | Figure 11: Service Chaining Using a Custom Bridge | 95 |
| | Figure 12: Service Chaining Using a Custom Bridge | 100 |
| | Figure 13: Configuring Cross-Connect | 105 |
| | Figure 14: Service Chaining for LAN Routing | 111 |
| | Figure 15: Configuring Cross-Connect | 114 |

List of Tables

| | | |
|------------------|--|------------|
| | About the Documentation | xi |
| | Table 1: Notice Icons | xiii |
| | Table 2: Text and Syntax Conventions | xiv |
| Chapter 1 | Overview | 17 |
| | Table 3: NFX150-C (Compact Models) | 18 |
| | Table 4: NFX150-S1 Models | 19 |
| | Table 5: Software Compatibility Details with only vSRX Installed | 21 |
| | Table 6: Software Compatibility Details with vSRX and Cloud CPE Solution | 22 |
| | Table 7: Software Compatibility Details with only vSRX Installed | 22 |
| | Table 8: Interfaces on the NFX150 | 27 |
| | Table 9: Default Mapping of Physical Ports to Virtual Ports on NFX150 | 27 |
| | Table 10: Default Mapping of Physical Ports to Virtual Ports for the Expansion Module | 28 |
| | Table 11: Features Supported on NFX150 | 29 |
| Chapter 2 | Initial Configuration | 31 |
| | Table 12: Security Policies | 31 |
| | Table 13: Interface Mapping | 31 |
| | Table 14: LTE Interfaces | 32 |
| Chapter 3 | Configuring Interfaces | 45 |
| | Table 15: Standard Bandwidths of DSL Operating Modes | 47 |
| | Table 16: VDSL2 Annex A and Annex B Features | 53 |
| | Table 17: Supported Profiles on the VDSL2 Interfaces | 54 |
| Chapter 4 | Configuring Security | 59 |
| | Table 18: IPsec Features Supported on NFX150 | 60 |
| Chapter 5 | Configuring VNFs | 71 |
| | Table 19: VNF Memory Allocation | 74 |
| Chapter 9 | Operational Commands | 139 |
| | Table 20: show system visibility cpu Output Fields | 146 |
| | Table 21: show system visibility host Output Fields | 148 |
| | Table 22: show system visibility memory Output Fields | 154 |
| | Table 23: show system visibility network Output Fields | 156 |
| | Table 24: show system visibility vnf Output Fields | 160 |
| | Table 25: show vmhost connections Output Fields | 165 |
| | Table 26: show vmhost forwarding-options analyzer Output Fields | 169 |
| | Table 27: show vmhost vlans Output Fields | 176 |

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 |
|--------------------------------|---|--|
| Bold text like this | Represents text that you type. | To enter configuration mode, type the configure command: user@host> configure |
| Fixed-width text like this | Represents output that appears on the terminal screen. | user@host> show chassis alarms No alarms currently active |
| <i>Italic text like this</i> | <ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies guide names. Identifies RFC and Internet draft titles. | <ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS CLI User Guide</i> RFC 1997, <i>BGP Communities Attribute</i> |
| <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@# set system domain-name <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"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE. |
| < > (angle brackets) | Encloses optional keywords or variables. | stub <default-metric metric>; |
| (pipe symbol) | Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity. | broadcast multicast (string1 string2 string3) |
| # (pound sign) | Indicates a comment specified on the same line as the configuration statement to which it applies. | rsvp { # Required for dynamic MPLS only |
| [] (square brackets) | Encloses a variable for which you can substitute one or more values. | community name members [community-ids] |
| 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. | } |

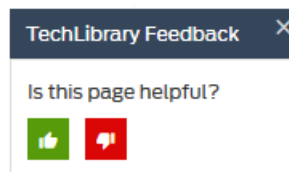
Table 2: Text and Syntax Conventions (continued)

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

Documentation Feedback

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

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



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

Requesting Technical Support

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

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <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/>
- Open a case online in the CSC Case Management tool: <https://www.juniper.net/cm/>

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

Opening a Case with JTAC

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

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

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

CHAPTER 1

Overview

- [NFX150 System Overview on page 17](#)
- [NFX150 Feature Overview on page 23](#)

NFX150 System Overview

- [Overview on page 17](#)
- [NFX150 Models on page 18](#)
- [Junos OS Releases Supported on NFX Series Hardware on page 20](#)
- [NFX Product Compatibility on page 21](#)
- [Benefits and Uses of NFX150 on page 23](#)

Overview

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

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

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

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

Figure 1: NFX150 Compact Model (Without LTE Support)

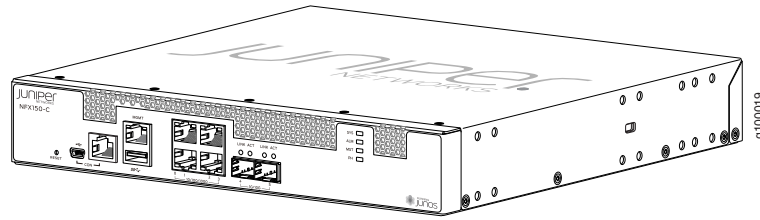
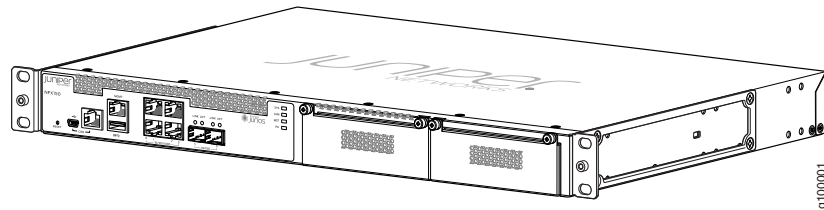


Figure 2: NFX150 Rack-Mount Model



For details on the various models, see “NFX150 Models” on page 18.

NFX150 Models

The NFX150 is available in seven models. [Table 3 on page 18](#) lists the NFX150-C models and [Table 4 on page 19](#) lists the NFX150-S1 models.

Table 3: NFX150-C (Compact Models)

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

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

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

Table 4: NFX150-S1 Models

| | NFX150-S1 | NFX150-S1E |
|-------------|--------------------------|--------------------------|
| CPU | 2.2 GHz 8-core Intel CPU | 2.2 GHz 8-core Intel CPU |
| RAM | 16 GB | 32 GB |
| Storage | 200 GB SSD | 200 GB SSD |
| Form Factor | 1 RU | 1 RU |

Table 4: NFX150-S1 Models (continued)

| | NFX150-S1 | NFX150-S1E |
|-----------------------------|--|--|
| Ports | Four 10/100/1000BASE-T RJ-45 ports which can be used as either access ports or uplinks | Four 10/100/1000BASE-T RJ-45 ports which can be used as either access ports or uplinks |
| | Two 1-Gigabit Ethernet/10-Gigabit Ethernet SFP+ ports | Two 1-Gigabit Ethernet/10-Gigabit Ethernet SFP+ ports |
| | One 10/100/1000BASE-T RJ-45 management port | One 10/100/1000BASE-T RJ-45 management port |
| | Console ports (RJ-45 and mini-USB) | Console ports (RJ-45 and mini-USB) |
| | One USB 3.0 port | One USB 3.0 port |
| LTE support | Yes (LTE expansion module) | Yes (LTE expansion module) |
| Expansion module support | Yes | Yes |
| Supported expansion modules | <ul style="list-style-type: none"> NFX-EM-6T2SFP— Expansion module with six 1-Gigabit Ethernet RJ-45 ports and two 1-Gigabit Ethernet SFP ports NFX-LTE-AE—Expansion module with a LTE modem supporting the frequency bands in Europe and North America. NFX-LTE-AA—Expansion module with a LTE modem supporting the frequency bands in Asia, Australia, and New Zealand. | <ul style="list-style-type: none"> NFX-EM-6T2SFP— Expansion module with six 1-Gigabit Ethernet RJ-45 ports and two 1-Gigabit Ethernet SFP ports NFX-LTE-AE—Expansion module with a LTE modem supporting the frequency bands in Europe and North America. NFX-LTE-AA—Expansion module with a LTE modem supporting the frequency bands in Asia, Australia, and New Zealand. |

NOTE: You can install only one expansion module on the NFX150-S1 devices. The expansion module must be installed in the first slot, which is next to the chassis LEDs.

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

Junos OS Releases Supported on NFX Series Hardware

The following table provides details of Junos OS software releases supported on the NFX Series platforms:

| 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) | NFX150 Software Download Page |
| NFX250 | 15.1X53-D45, and 15.1X53-D47 17.2R1 or later | nfx-2 (jinstall-host-nfx-2-flex-x86-64-<release-number>-secure-signed.tgz) | NFX250 Software Download Page |

NFX Product Compatibility

- [Hardware Compatibility on page 21](#)
- [Software Version Compatibility on page 21](#)

Hardware Compatibility

To obtain information about the components that are supported on the 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 NFX150 and NFX250 platforms:



NOTE: Starting in Junos OS Release 18.1R1, NFX series supports the same version of platform software and vSRX. For example, see [Table 5 on page 21](#) and [Table 7 on page 22](#).

NFX150 Software Version Compatibility

This section lists the vSRX software releases that are compatible with the Junos OS releases on the NFX150 platform:

Table 5: Software Compatibility Details with only vSRX Installed

| NFX150 Junos OS Release | vSRX |
|-------------------------|--------|
| 18.1R1 | 18.1R1 |
| 18.1R2 | 18.1R2 |
| 18.1R3 | 18.1R3 |
| 18.2R1 | 18.2R1 |
| 18.3R1 | 18.3R1 |

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 platform:

Table 6: 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-D61 | 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 |

Table 7: Software Compatibility Details with only vSRX Installed

| NFX250 Junos OS Release | vSRX |
|-------------------------|---------------|
| 15.1X53-D40.3 | 15.1X49-D40.6 |
| 15.1X53-D41.6 | 15.1X49-D40.6 |
| 15.1X53-D45.3 | 15.1X49-D61 |
| 15.1X53-D47.4 | 15.1X49-D78.3 |
| 17.2R1 | 15.1X49-D75 |
| 17.3R1 | 15.1X49-D100 |
| 15.1X53-D471 | 15.1X49-D143 |
| 18.1R1 | 18.1R1 |
| 18.1R2 | 18.1R2 |
| 18.1R3 | 18.1R3 |
| 18.2R1 | 18.2R1 |
| 18.3R1 | 18.3R1 |

Benefits and Uses of NFX150

The NFX150 Network Services Platform provides these benefits:

- Highly scalable, supporting multiple Juniper and third-party VNFs on a single device. The modular software architecture provides high performance and scalability for routing, switching, and security enhanced by carrier-class reliability.
- Integrated security, routing, and switching functionality in a single control plane simplifies management and deployment.
- Supports a variety of flexible deployments. A distributed services deployment model ensures high availability, performance, and compliance. The NFX150 provides an open framework that supports industry standards, protocols, and seamless API integration.
- In addition to Ethernet connections, Wireless WAN support through the LTE module provides more flexibility in deployments.
- Supports advanced security features such as IPsec connectivity, applications detection, and filtering for malicious traffic.
- The Secure Boot feature safeguards device credentials, automatically authenticates system integrity, verifies system configuration, and enhances overall platform security.
- Automated configuration eliminates complex device setup and delivers a plug-and-play experience.

Related Documentation

- [NFX150 Feature Overview on page 23](#)

NFX150 Feature Overview

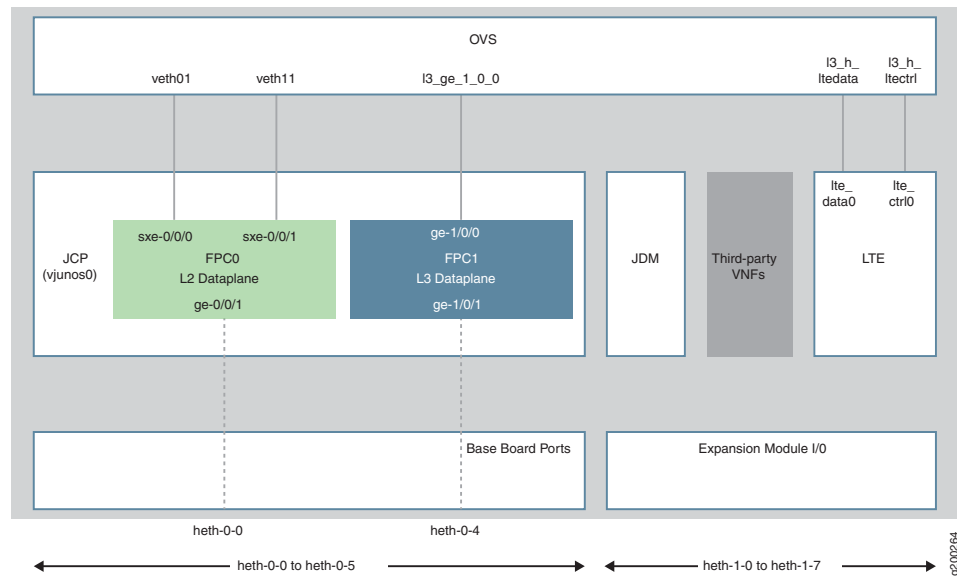
- [Software Architecture on page 23](#)
- [Interfaces on page 25](#)
- [Supported Features on page 28](#)
- [Licensing on page 29](#)

Software Architecture

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

[Figure 3 on page 24](#) illustrates the architecture of the NFX150.

Figure 3: NFX150 Architecture



Key components of the system software include:

- VNF—A VNF is a consolidated offering that contains all the components required for supporting a fully virtualized networking environment. You can configure and use third-party VNFs in service chains.
- Junos Control Plane (JCP)—The JCP is the Junos VM running on the host OS, Wind River Linux. The JCP functions as the single point of management for all the components. The JCP controls the Layer 2 dataplane, which provide the Layer 2 services and the Layer 3 dataplane, which provides the Layer 3 to Layer 7 services.

In addition to chassis management, JCP enables:

- Configuration of advanced security features.
- Management of guest virtualized network functions (VNFs) during their life cycle.
- Installation of third-party VNFs.
- Creation of VNF service chains.
- Management of guest VNF images (their binary files).
- Management of the system inventory and resource usage.
- Management of the LTE interface.
- Juniper Device Manager (JDM)—An application container that manages VNFs and provides infrastructure services. The JDM functions in the background and users cannot access JDM directly.
- L2 Dataplane—The Layer 2 dataplane that manages the Layer 2 traffic. The Layer 2 dataplane forwards the LAN traffic to the NFV backplane, Open vSwitch (OVS). The

Layer 2 dataplane is mapped to the virtual FPC0 on the JCP. By default, all the 1-Gigabit Ethernet physical ports are mapped to the virtual interfaces on the Layer 2 dataplane.

- L3 Dataplane—The Layer 3 dataplane that provides datapath functions for the Layer 3 to Layer 7 services. The Layer 3 dataplane is mapped to the virtual FPC1 on the JCP. By default, the two SFP+ ports on the NFX150 chassis are mapped to the virtual interfaces on the Layer 3 dataplane.
- Linux—The host OS, WindRiver Linux. In Junos OS Release 18.1R1, the WindRiver Linux version is 8.
- Open vSwitch (OVS) bridge—The OVS bridge is a VLAN-aware system bridge, which acts as the NFV backplane to which the VNFs and FPCs connect. Additionally, you can create custom OVS bridges to isolate connectivity between different VNFs.
- LTE—A containerized driver that provides 4G LTE connectivity management. The LTE container is bound to the FPC1 for management.

Interfaces

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

Physical Interfaces

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

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

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

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

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



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

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

Virtual Interfaces

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

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

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

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

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

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

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



NOTE: The NFX150 devices do not support integrated routing and bridging (IRB) interfaces. The IRB functionality is provided by ge-1/0/0, which is always mapped to the service chaining backplane (OVS). Note that this mapping cannot be changed.

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

LTE Interface

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

Interface Mapping

Table 8 on page 27 summarizes the interfaces on the NFX150.

Table 8: Interfaces on the NFX150

| Interface Name | Description |
|----------------------|--|
| heth-0-0 to heth-0-5 | <p>Physical ports on the front panel of the NFX150 device, which can be mapped to Layer 2 or Layer 3 interfaces, or VNFs.</p> <p>Ports heth-0-0 to heth-0-3 are 10 Mbps/100 Mbps/1 Gbps tri-speed copper ports mapped to the LAN ports ge-0/0/0 to ge-0/0/3, respectively.</p> <p>Ports heth-0-4 and heth-0-5 are 10 Gbps SFP+ ports mapped to WAN ports ge-1/0/1 and ge-1/0/2, respectively.</p> |
| heth-1-0 to heth-1-7 | <p>Physical ports on the expansion module of the NFX150-S1 device. These ports are mapped to the ge-0/0/n ports by default.</p> <p>Ports heth-1-0 to heth-1-5 are 10 Mbps/100 Mbps/1 Gbps tri-speed copper ports mapped to the LAN ports ge-0/0/4 to ge-0/0/9, respectively.</p> <p>Ports heth-1-6 and heth-1-7 are 1 Gbps SFP ports mapped to the LAN ports ge-0/0/10 and ge-0/0/11 respectively.</p> |
| ge-0/0/x | <p>Logical Layer 2 interfaces, which can be used for LAN connectivity. The values of x ranges from:</p> <ul style="list-style-type: none"> • 0 to 3 for NFX150 devices without an expansion module • 0 to 11 for NFX150 devices with an expansion module |
| ge-1/0/x | A set of up to 10 logical Layer 3 interfaces. Each of these interfaces can have 4k sub-interfaces. The value of x ranges from 0 to 9. |
| cl-1/1/0 | The LTE cellular interface, which carries the physical layer attributes. |
| dl0 | The LTE dialer interface, which carries Layer 3 and security services. The security flow session contains the dl0 interface as the ingress or egress interface. |
| st0 | Secure tunnel interface used for IPsec VPNs. |
| fxp0 | The out-of-band management interface. |

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

Table 9 on page 27 illustrates the default mapping between the physical and virtual interfaces on a NFX150 device.

Table 9: Default Mapping of Physical Ports to Virtual Ports on NFX150

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

Table 9: Default Mapping of Physical Ports to Virtual Ports on NFX150 (continued)

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

Table 10 on page 28 illustrates the default mapping between the physical ports on the expansion module and the virtual interfaces.

Table 10: Default Mapping of Physical Ports to Virtual Ports for the Expansion Module

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



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

Supported Features

Table 11 on page 29 lists the Junos features supported on NFX150.

Table 11: Features Supported on NFX150

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

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

Licensing

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

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



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

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



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

**Related
Documentation**

- [Initial Configuration on NFX150 Devices on page 31](#)
- [Mapping Interfaces on NFX150 Devices on page 45](#)

CHAPTER 2

Initial Configuration

- [Initial Configuration on NFX150 Devices on page 31](#)
- [Provisioning the NFX150 Device Using Network Activator on page 34](#)
- [Configuring the LTE Module on NFX Devices on page 36](#)
- [Upgrading the Modem Firmware on NFX Devices Through Over-the-Air \(OTA\) on page 42](#)

Initial Configuration on NFX150 Devices

- [Factory-Default Settings on page 31](#)
- [Enabling Basic Connectivity on page 32](#)
- [Establish the Connection on page 33](#)

Factory-Default Settings

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

Table 12: Security Policies

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

Table 13: Interface Mapping

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

Table 14: LTE Interfaces

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

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

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

Enabling Basic Connectivity

1. Ensure that the NFX150 device is powered on.
2. Connect to the console port:
 - a. Plug one end of the Ethernet cable into the console port on your NFX150 device.
 - b. Connect the other end of the Ethernet cable to the RJ-45-to-DB-9 serial port adapter shipped with your device.
 - c. Connect the RJ-45-to-DB-9 serial port adapter to the serial port on the management device. Use the following values to configure the serial port:
Baud rate—9600; Parity—N; Data bits—8; Stop bits—1; Flow control—None.



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

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

3. Use any terminal emulation program, such as HyperTerminal, to connect to the device console. The CLI displays a login prompt.
4. Log in as **root** and enter the password **juniper123**. If the software completes booting before you connect to the console, you might need to press the **Enter** key for the prompt to appear:


```
login: root
password: juniper123
```
5. Start the CLI:


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

6. Enter configuration mode:

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

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

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

8. Enable SSH service for the root user:

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

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

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

10. Commit the configuration:

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

Establish the Connection

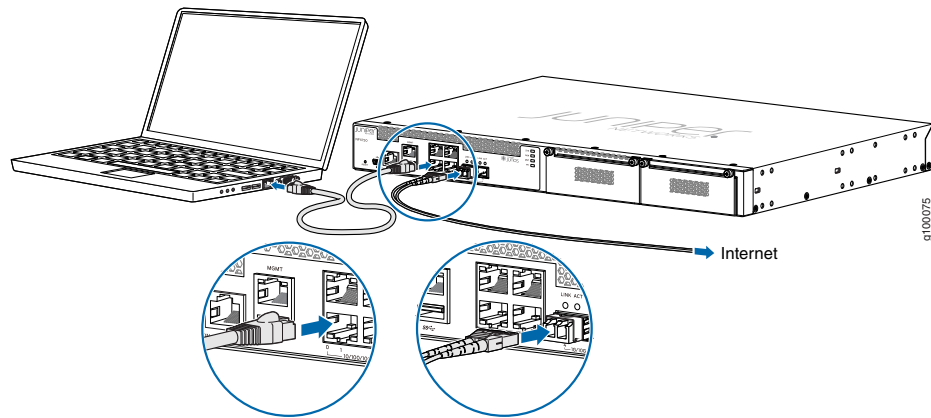
1. Connect the device to the Internet Service Provider (ISP) by using the following step:



NOTE: For information on interface mapping, see [Table 13 on page 31](#) and [Table 14 on page 32](#).

Connect one of the WAN ports such as **heth-0-4** or **heth-0-5** to the ISP. The device is assigned an IP address by the ISP through DHCP.

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



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



NOTE: The LTE expansion module must be purchased separately.

2. Connect the laptop to one of the front panel LAN ports such as **heth-0-0**, **heth-0-1**, **heth-0-2**, or **heth-0-3**. The laptop is assigned an IP address by the DHCP server running on the interface.
3. Open a browser on your laptop, navigate to www.juniper.net, and verify your connectivity.

Related Documentation

- [Configuring the LTE Module on NFX Devices on page 36](#)

Provisioning the NFX150 Device Using Network Activator

- [Understanding Network Activator on page 34](#)
- [Provisioning the NFX150 Device on page 35](#)

Understanding Network Activator

Network Activator facilitates the initial provisioning of the NFX150 Network Services Platform (referred to as remote devices further in this documentation). The remote devices can download a boot image and initial configuration from servers hosting the Network Activator, using a process that provides full authentication and authorization for all interactions. Once initial provisioning is complete, the remote device can be managed and monitored by a Management Server. Network Activator uses a distributed architecture to support remote devices.

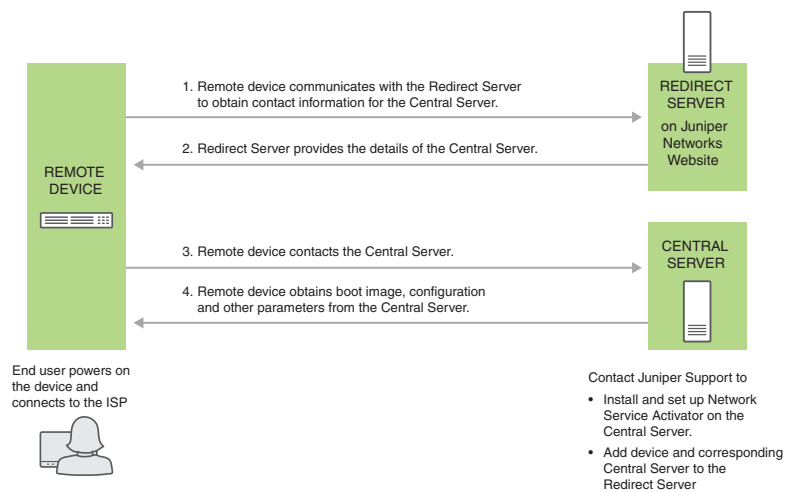
The initial provisioning process involves the following components:

- Remote device—Sends requests to Juniper's Redirect Server, which is the default configuration on the NFX150.
- Central server—Hosts the Network Activator and communicates with remote devices. The central server is located at a central geographical location for the service provider.
- Redirect server—Provides authentication and authorization for remote devices to access their assigned central servers through use of ITU-T X.509 private key infrastructure (PKI) and digital certificates. The Redirect Server resides at Juniper Networks.

Provisioning the NFX150 Device

Figure 5 on page 35 illustrates the workflow of the initial provisioning.

Figure 5: Workflow for Initial Provisioning of the NFX150



NOTE: Contact Juniper Support to install and set up Network Service Activator on the central server, and to add the device and corresponding central server to the Redirect Server.

In detail, the provisioning workflow proceeds as follows:

1. The end user powers on the remote device, and connects it to the ISP through the WAN ports.
2. The 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 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 device sends a request to the central server for the URL of the boot image and the location of the initial configuration. The central server responds with the requested information.
6. The device fetches the boot image and configuration from the central server.
7. The devices upgrades to the boot image (if the boot image is different from the image running on the NFX150) and applies the configuration to start and become operational.

Configuring the LTE Module on NFX Devices

The LTE module can be configured in three modes:

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



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

- Dial-on-demand—The LTE module initiates a connection when it receives interesting traffic. You define interesting traffic using the dialer filter. To configure dial-on-demand using a dialer filter, you first configure the dialer filter and then apply the filter to the dialer interface.
- Backup—The LTE module connects to the 3G/4G network when the primary connection fails.

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

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

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

- Username and password
- Access Point Name (APN)

- Authentication (Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP))
- [Configuring the LTE Module for Primary Mode on page 37](#)
- [Configuring the LTE Module for Dial-on-Demand Mode on page 38](#)
- [Configuring the LTE Module for Backup Mode on page 40](#)

Configuring the LTE Module for Primary Mode

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

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

To configure the LTE module as a primary interface:

1. Configure the dialer interface:

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

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

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



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

3. Configure the profile.

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



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

4. Verify that the profile is configured successfully:

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

5. Activate the SIM card:

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

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

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



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

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

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

Configuring the LTE Module for Dial-on-Demand Mode

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

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

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



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

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

1. Configure the dialer interface:

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

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

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

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

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



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

4. Create the dialer filter rule:

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

5. Set the default route:

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

6. Configure the profile.

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



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

7. Verify that the profile is configured successfully:

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

8. Activate the SIM card:

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

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

```

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

```



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

10. Verify the configuration by sending traffic to the destination address. The traffic is routed to the dI0 interface and if it matches the dialer filter rule, then the dI0 is triggered to dial.
11. Verify the status of the wireless network and dialer interface:

```

user@host# run show modem wireless network
user@host# run show interfaces dI0.0

```

Configuring the LTE Module for Backup Mode

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

To configure the LTE module as a backup interface:

1. Configure the dialer interface:

```

user@host# set interfaces dI0 unit 0 family inet negotiate-address
user@host# set interfaces dI0 unit 0 family inet6 negotiate-address
user@host# set interfaces dI0 unit 0 dialer-options pool dialer-pool-number
user@host# set interfaces dI0 unit 0 dialer-options dial-string dial-number

```

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

```

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

```



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

3. Configure the profile.

```

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

```




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

- Verify that the profile is configured successfully:

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

- Activate the SIM card:

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

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

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



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

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

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

- Verify the status of the wireless network and dialer interface:

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

Related Documentation

- [Upgrading the Modem Firmware on NFX Devices Through Over-the-Air \(OTA\) on page 42](#)

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

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



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

1. Enable OTA upgrade on the LTE module:

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

```
Set FOTA on modem succeeded
```

2. Initiate the firmware upgrade:

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

```
Launch FOTA upgrade succeeded
```

3. Verify the firmware upgrade status:

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

```
LTE mPIM firmware details
  Product name: Junos LTE mPIM
  Serial number: D23F4349-10FA-41AA-A538-03648DE
  Hardware version: AcceleratedConcepts/porter
  Firmware version: 17.11.13
  MAC: 00:00:5e:00:53:82
  System uptime: 4632 seconds
Wireless modem firmware details
  Modem firmware version:
9999999_9904609_SWI9X30C_02.24.05.06_00_GENERIC_002.026_000
  Modem Firmware build date: 19/05/2017
  Card type: MC7455
  Modem manufacturer: Sierra Wireless, Inc
  Hardware version: 1.0
  Power & Temperature: Normal 3368 mV, Normal 29.00 C
OTA status
  State: Enabled
  New firmware available: No
Number of SIM: 1
Slot of active: 1
Status of SIM 1
  SIM state: SIM present
  Modem PIN security status: Disabled
  SIM status: SIM Okay
```

```
SIM user operation needed: No Op
Retries remaining: 3
```

4. Check the LTE module connection status:

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

```
LTE Connection details
  Connected time: 2880
  IP: 10.12.219.210
  Gateway: 10.12.219.209
  DNS: 123.123.123.123
  IPv6: ::
  Gatewayv6: ::
  DNSv6: ::
  Input bps: 0
  Output bps: 0
  Bytes Received: 1952
  Bytes Transferred: 2164
  Packets Received: 10
  Packets Transferred: 20
Wireless Modem Network Info
  Current Modem Status: Connected
  Current Service Status: Normal
  Current Service Type: PS
  Current Service Mode: LTE
  Current Band: B3
  Network: UNICOM
  Mobile Country Code (MCC): 460
  Mobile Network Code (MNC): 1
  Location Area Code (LAC): 65534
  Routing Area Code (RAC): 0
  Cell Identification: 239907605
  Access Point Name (APN): 3gnet
  Public Land Mobile Network (PLMN): CHN-UNICOM
  Physical Cell ID (PCI): 452
  International Mobile Subscriber Identification (IMSI): *****
  International Mobile Equipment Identification (IMEI/MEID): *****
  Integrate Circuit Card Identity (ICCID): 89860117811046631207
  Reference Signal Receiving Power (RSRP): -71
  Reference Signal Receiving Quality (RSRQ): -8
  Signal to Interference-plus-Noise Ratio (SINR): 19
  Signal Noise Ratio (SNR): 22
  Energy per Chip to Interference (ECIO): 0
```

Related Documentation • [Configuring the LTE Module on NFX Devices on page 36](#)

CHAPTER 3

Configuring Interfaces

- [Mapping Interfaces on NFX150 Devices on page 45](#)
- [Configuring the In-Band Management Interface for NFX150 on page 46](#)
- [ADSL2 and ADSL2+ Interfaces on NFX150 Devices on page 47](#)
- [VDSL2 Interfaces on NFX150 Devices on page 51](#)

Mapping Interfaces on NFX150 Devices

- [Mapping Physical Interfaces to Virtual Interfaces on NFX150 Devices on page 45](#)
- [Mapping Physical Ports to VNF Interfaces Through SR-IOV on page 46](#)
- [Mapping Layer 3 Dataplane Interfaces to OVS on page 46](#)

Mapping Physical Interfaces to Virtual Interfaces on NFX150 Devices

The NFX150 devices are shipped with the default mapping as described in “[NFX150 Feature Overview](#)” on [page 23](#). You can change the mapping to suit your needs. Any of the physical ports can be mapped to the ge-0/0/x or ge-1/0/x interfaces. Any change in port mapping configuration will automatically reset the affected FPC.

- To map a physical port to a virtual interface, use the following command:

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

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

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

- To map a physical port to the FPC0, use the following command:

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

- To map a physical port to the FPC1, use the following commands:

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

```
root# set interfaces virtual-interface-name unit 0 vlan-id vlan-id
root# set interfaces virtual-interface-name unit 0 family inet address ip-address
```

Mapping Physical Ports to VNF Interfaces Through SR-IOV

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

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

Mapping Layer 3 Dataplane Interfaces to OVS

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

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

Related Documentation

- [Configuring the In-Band Management Interface for NFX150 on page 46](#)

Configuring the In-Band Management Interface for NFX150

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 NFX150 Devices

- [ADSL Interface Overview on page 47](#)
- [Configuring ADSL SFP Interface Using VLANs on NFX150 Network Services Platform on page 48](#)
- [Configuring ADSL SFP Interface Without Using VLANs on NFX150 Network Services Platform on page 50](#)

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 15 on page 47](#).

Table 15: 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

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

Supported security devices with Mini-PIMs can use PPP over Ethernet over ATM (PPPoEoA) and PPP over ATM (PPPoA) to connect through ADSL lines only.

- [ADSL2 and ADSL2+ on page 48](#)

ADSL2 and ADSL2+

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

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

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

Configuring ADSL SFP Interface Using VLANs on NFX150 Network Services Platform



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

To configure ADSL SFP interfaces on NFX150 devices:

1. Connect to the host.

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

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

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

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

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

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

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


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

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

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

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

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

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



NOTE:

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

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

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

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

- Commit the configuration.

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

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

```
[edit]
```

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

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

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



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

To configure ADSL SFP interfaces on NFX150 devices:

1. Connect to the host.

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

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

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

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

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

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

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



NOTE:

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

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

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

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

5. Commit the configuration.

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

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

```
[edit]
user@host# show interfaces heth-0-4

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

Related Documentation

VDSL2 Interfaces on NFX150 Devices

- [VDSL Interface Overview on page 51](#)
- [VDSL2 Network Deployment Topology on page 52](#)
- [VDSL2 Interface Support on NFX Series Devices on page 53](#)
- [Configuring VDSL SFP Interface Using VLANs on NFX150 Network Services Platform on page 55](#)
- [Configuring VDSL SFP Interface Without Using VLANs on NFX150 Network Services Platform on page 56](#)

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 52](#)

VDSL2 Vectoring Overview

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

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

VDSL2 Network Deployment Topology

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

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

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

1. Connect an end-user device such as a LAN, hub, or PC through an Ethernet interface to the customer premises equipment (CPE) (for example, an NFX device).
2. Connect the CPE to a DSLAM.
3. The VDSL2 interface uses either Gigabit Ethernet or fiber as second mile to connect to the Broadband Remote Access Server (B-RAS) as shown in [Figure 6 on page 53](#).
4. The ADSL interface uses either Gigabit Ethernet (in case of IP DSLAM) as the "second mile" to connect to the B-RAS or OC3/DS3 ATM as the second mile to connect the B-RAS as shown in [Figure 7 on page 53](#).



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

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

Figure 6 on page 53 shows a typical VDSL2 network topology.

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

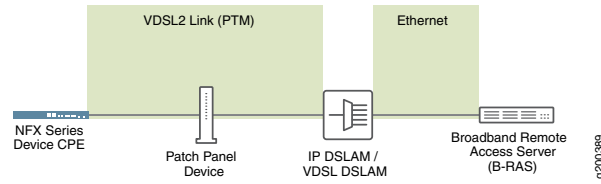
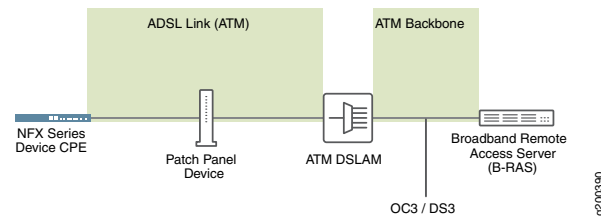


Figure 7 on page 53 shows a backward-compatible ADSL topology using ATM DSLAM.

Figure 7: 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 16 on page 53. (Platform support depends on the Junos OS release in your installation.)

Table 16: 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 heth-0-4 and heth-0-5.



NOTE:

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

VDSL2 Interfaces Supported Profiles

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

[Table 17 on page 54](#) lists the different profiles supported on the VDSL2 interfaces and their properties.

Table 17: Supported Profiles on the VDSL2 Interfaces

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

VDSL2 Interfaces Supported Features

The following features are supported on the VDSL2 interfaces:

- ADSL2 and ADSL2+ backward compatibility with Annex A, Annex M support
- PTM or EFM (802.3ah) support
- Operation, Administration, and Maintenance (OAM) support for ADSL2 and ADSL2+ modes

- Multilink Point-to-Point Protocol (MLPPP) (supported only when the VDSL2 Mini-PIM is operating in ADSL2 mode)
- MTU size of 1514 bytes (maximum) in VDSL2 mode and 1496 bytes in ADSL mode.
- Support for maximum of 10 permanent virtual connections (PVCs) (only in ADSL2 and ADSL2+ mode)

Configuring VDSL SFP Interface Using VLANs on NFX150 Network Services Platform



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

To configure VDSL SFP interfaces on NFX150 devices:

1. Connect to the host.

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

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

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

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

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

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

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

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

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

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

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

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

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

**NOTE:**

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

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

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

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

7. Commit the configuration.

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

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

```
[edit]
```

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

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

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



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

To configure VDSL SFP interfaces on NFX150 devices:

1. Connect to the host.

```
user@host> configure
```



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

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

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

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

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

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

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



NOTE:

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

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

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

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

5. Commit the configuration.

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

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

```
[edit]
user@host# show interfaces heth-0-4

Physical interface: heth-0-4, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Fiber, MTU: 9192, VDSL2 mode, Speed: 1Gbps, Duplex: Full-duplex, Auto-negotiation: Disabled
```

```
VDSL status:  
  Modem status : Showtime (Auto)  
  VDSL profile  : Profile-8b  
  Device flags  : Present Running  
  Current address: d8:b1:22:33:5e:51, Hardware address: d8:b1:22:33:5e:51
```

**Related
Documentation**

CHAPTER 4

Configuring Security

- [IP Security on NFX Devices on page 59](#)
- [UTM on NFX Devices on page 67](#)
- [Application Security on NFX Devices on page 68](#)
- [Intrusion Detection and Prevention on NFX Devices on page 69](#)
- [Integrated User Firewall Support on NFX Devices on page 69](#)

IP Security on NFX Devices

- [Overview on page 59](#)
- [Configuring Security on page 61](#)

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 18 on page 60 lists the IPsec features supported on NFX150 devices.

Table 18: IPsec Features Supported on NFX150

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

Table 18: IPsec Features Supported on NFX150 (continued)

| Features | Reference |
|---------------------|-----------------------------------|
| NAT-T | Understanding NAT-T |
| Dead Peer Detection | Understanding Dead Peer Detection |

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 61](#)
- [Configuring Routing Options on page 62](#)
- [Configuring Security IKE on page 62](#)
- [Configuring Security IPsec on page 64](#)
- [Configuring Security Policies on page 66](#)
- [Configuring Security Zones on page 67](#)

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 69](#)
- [Integrated User Firewall Support on NFX Devices on page 69](#)

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 69](#)

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 67](#)

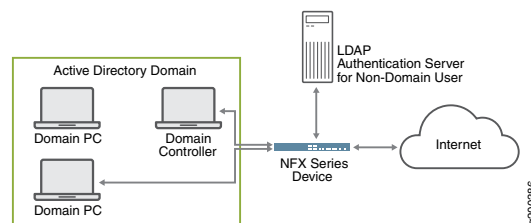
Integrated User Firewall Support on NFX Devices

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 8 on page 70](#) 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 8: 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 67](#)

CHAPTER 5

Configuring VNFs

- [Configuring VNFs on NFX150 Devices on page 71](#)
- [Managing VNFs on NFX150 Devices on page 77](#)
- [Configuring Analyzer VNF and Port-mirroring on page 81](#)

Configuring VNFs on NFX150 Devices

NFX150 devices support launching and management of Virtual Network Functions (VNFs) from the Junos Control Plane (JCP). JCP supports the creation and management of third-party Virtual Machines (VMs).



NOTE: NFX150 does not support vSRX and VMX as VMs.

- [VNF Configuration Overview on page 71](#)
- [Loading the VNF Image on page 72](#)
- [Preparing the Bootstrap Configuration on page 72](#)
- [Launching a VNF on page 72](#)
- [Verify that the VNF Launched Successfully on page 73](#)
- [Allocating a CPU for a VNF on page 73](#)
- [Allocating Memory for a VNF on page 74](#)
- [Configuring VNF Storage Devices on page 74](#)
- [Configuring VNF Interfaces and VLANs on page 75](#)
- [Launching a VNF Using an XML Descriptor File on page 77](#)

VNF Configuration Overview

The configuration process for the VNFs includes the following tasks:

1. Log in to the JCP.
2. Load the VNF image.
3. Launch the VNF.
4. Verify that the VNF launched successfully.

5. Allocate a CPU for the VNF.
6. Allocate the memory for the VNF.
7. Configure storage devices for the VNF.
8. Configure Interfaces and VLANs for the VNF.

Loading the VNF Image

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



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

Preparing the Bootstrap Configuration

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

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

Launching a VNF

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

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

- To launch a VNF using an image:



NOTE: Only Qcow2 and Raw image types are supported.

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



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

- (Optional) To specify a UUID for the VNF:

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


uuid is an optional parameter, and it is recommended to allow the system to allocate a UUID for the VNF.



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

Verify that the VNF Launched Successfully

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

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

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

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

Allocating a CPU for a VNF

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

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

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

2. Pin a virtual CPU to a physical CPU

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

3. Commit the configuration:

```
user@host# commit
```

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



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

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

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

Allocating Memory for a VNF

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

By default, a certain amount of memory is allocated for VNFs. [Table 19 on page 74](#) lists the VNF memory allocation for the different NFX150 models.

Table 19: VNF Memory Allocation

| Model | Memory Allocation |
|-----------------|-------------------|
| NFX150-C-S1 | 5 GB |
| NFX150-C-S1-AE | 5 GB |
| NFX150-C-S1-AA | 5 GB |
| NFX150-C-S1E-AE | 13 GB |
| NFX150-C-S1E-AA | 13 GB |
| NFX150-S1 | 13 GB |
| NFX150-S1E | 29 GB |

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

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



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

Configuring VNF Storage Devices

The NFX150 supports the following storage options for VNFs:

- CD-ROM
- Disk
- USB

To add a virtual CD or to update the source file of a virtual CD, enter the following command:

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

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

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

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

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

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

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

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



NOTE:

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

Configuring VNF Interfaces and VLANs

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

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

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

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

2. To create VLAN:

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

3. To attach a VNF interface to a VLAN:

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



NOTE:

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

4. 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 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 when the VNF is in running state.

5. To delete a VNF interface:

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

**NOTE:**

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

Launching a VNF Using an XML Descriptor File

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

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



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

Managing VNFs on NFX150 Devices

- [Managing VNF States on page 78](#)
- [Managing VNF MAC Addresses on page 78](#)
- [Managing MTU on page 79](#)
- [Accessing a VNF from JCP on page 80](#)
- [Viewing List of VNFs on page 80](#)
- [Displaying the VNF Details on page 80](#)
- [Deleting a VNF on page 80](#)

Managing VNF States

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

1. To disable an autostart of a VNF on a VNF configuration commit:

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

2. To manually start a VNF:

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

3. To stop a VNF:

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

4. To restart a VNF:

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

5. To access the console of an active VNF:

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

6. To access a VNF through SSH:

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

7. 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 a 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. You can configure a MAC address other than that 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 launched.

1. 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
```

2. 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 start 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 VNF interfaces.

Managing MTU

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.



NOTE: MTU configuration is supported only on VLAN interfaces.

1. To configure 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 MTU if the VNF does not support hot-plugging functionality.

2. To delete MTU of a VNF interface:

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



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

**NOTE:**

- MTU size can be either 1500 bytes or 2048 bytes.
- The maximum number of VLAN interfaces on the OVS that can be configured in the system is limited to 25.
- The maximum size of the MTU for a VNF interface is 2048 bytes.

Accessing a VNF from JCP

You can access a VNF from JCP by using the following command:

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

Viewing 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 |
| 11057 | LTE | Running | alive |

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

Displaying the VNF Details

To display VNF details:

```
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 the VNF.

Configuring Analyzer VNF and Port-mirroring

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



NOTE:

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

To configure the analyzer VNF and enable port-mirroring:

1. Configure the analyzer VNF:

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

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

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

Related Documentation

- [Configuring VNFs on NFX150 Devices on page 71](#)

CHAPTER 6

Configuring Service Chaining

- [Service Chaining on NFX150 Devices on page 83](#)
- [Example: Configuring Service Chaining Using VLANs on NFX150 Network Services Platform on page 85](#)
- [Example: Configuring Service Chaining Using SR-IOV on NFX150 Network Services Platform on page 89](#)
- [Example: Configuring Service Chaining Using a Custom Bridge on page 94](#)
- [Example: Configuring Service Chaining for LAN-WAN Routing on page 100](#)
- [Example: Configuring Cross Connect on NFX150 Devices on page 103](#)
- [Example: Configuring Service Chaining for LAN Routing on page 111](#)
- [Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices on page 113](#)

Service Chaining on NFX150 Devices

- [Understanding Service Chaining on page 83](#)
- [Configuring Service Chaining Using VLANs on page 84](#)
- [Configuring Service Chaining Using DHCP Services on VLANs on page 84](#)

Understanding Service Chaining

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

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

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

Configuring Service Chaining Using VLANs



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

To configure service chaining:

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

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

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

- Create a VLAN with a VLAN ID:

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

- Create a VLAN using a list of VLAN IDs:

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

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

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

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

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

Configuring Service Chaining Using DHCP Services on VLANs

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

To configure service chaining:

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

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



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

2. Specify the IP address pool to be used:

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

3. Attach an interface from FPC1 to the VLAN:

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

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

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

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

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

6. Enable the DHCP client on the VNF.

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

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

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

- [Requirements on page 85](#)
- [Overview on page 85](#)
- [Configuration on page 86](#)

Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

Before you configure service chaining, be sure you have:

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

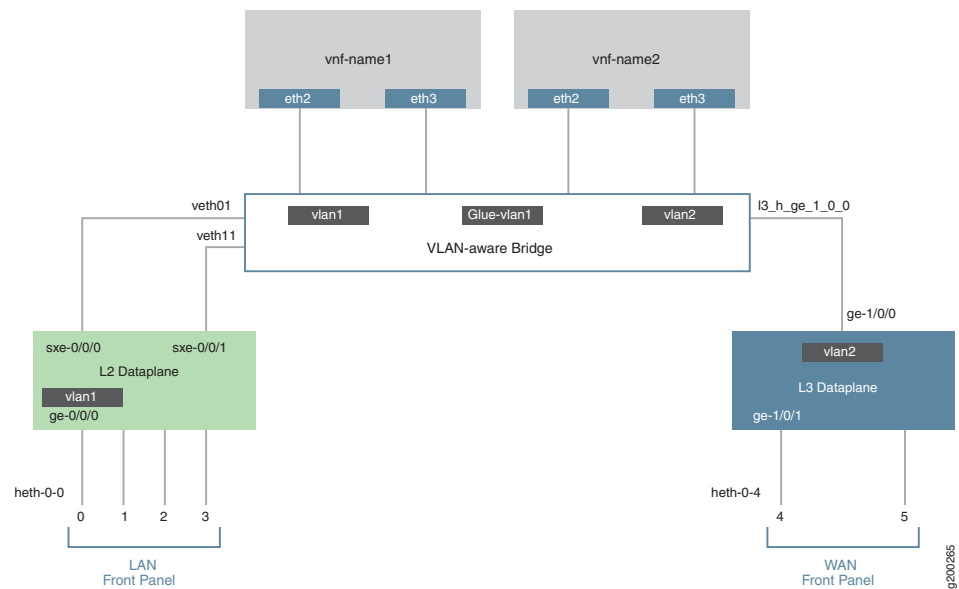
Overview

Service chaining on a device running the disaggregated Junos OS allows multiple services, or virtual network functions (VNFs), to be applied to traffic as it flows through the device. This example explains how to configure the various layers of the device to enable traffic to enter the device, flow through two service VNFs, and exit the device.

Topology

This example uses a single device running the disaggregated Junos OS, as shown in [Figure 9 on page 86](#).

Figure 9: Service Chaining Using VLANs



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

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

Configuration

- [Configuring the Interfaces on page 86](#)
- [Configuring the VNF Interfaces and Creating the Service Chain on page 89](#)

Configuring the Interfaces

Step-by-Step Procedure

To configure the interfaces:

1. Connect to the JCP.

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

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

```

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

```

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

```

user@jcp# set vlans vlan1 vlan-id 77

```

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

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

```

user@jcp# set interfaces ge-0/0/2.0 family ethernet-switching vlan members vlan1

```

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

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

```

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

```

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

```

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

```

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

```

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

```

8. Commit the configuration.

```

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

```

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

```

[edit]
user@host# show interfaces ge-0/0/2

```

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

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

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

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

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

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

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

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

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


Configuring the VNF Interfaces and Creating the Service Chain

Step-by-Step Procedure

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

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

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

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

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

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

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

4. Finally, configure the IP addresses and static routes for each interface of the VNFs as shown in [Figure 9 on page 86](#).

Related Documentation

- [Understanding Service Chaining on Disaggregated Junos OS Platforms](#)
- [Disaggregated Junos OS VMs](#)
- [Understanding Virtio Usage](#)

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

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

- [Requirements on page 90](#)
- [Overview on page 90](#)
- [Configuration on page 91](#)

Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

Before you configure service chaining, be sure you have:

- Installed and launched the relevant VNFs

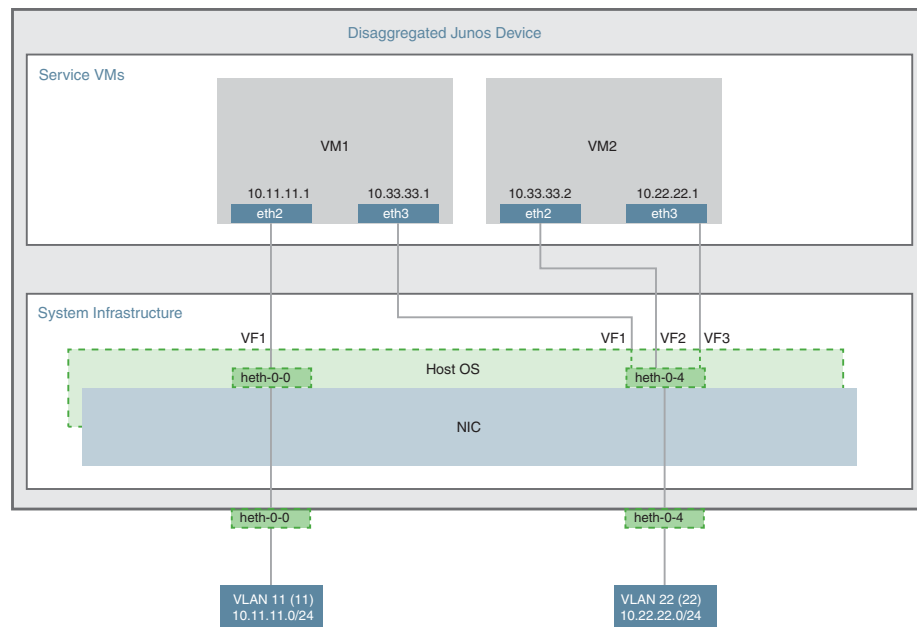
Overview

Service chaining on a device running the disaggregated Junos OS allows multiple services, or virtual network functions (VNFs), to be applied to traffic as it flows through the device. This example explains how to configure the various layers of the device to enable traffic to enter the device, flow through two service VNFs, and exit the device.

Topology

This example uses a single device running the disaggregated Junos OS, as shown in [Figure 10 on page 90](#).

Figure 10: Service Chaining Using SR-IOV—Device Infrastructure



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

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

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

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

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

Configuration

This example describes:

- [Configuring the Packet Forwarding Engine Interfaces on page 91](#)
- [Creating the Service Chain on page 94](#)

Configuring the Packet Forwarding Engine Interfaces

CLI Quick Configuration

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

```
[edit]
user@jcp#

set vlans Vlan11 vlan-id 11
set interfaces ge-0/0/0 family ethernet-switching vlan member Vlan11
set interfaces sxe-0/0/0.0 family ethernet-switching interface-mode trunk
set interfaces sxe-0/0/0.0 family ethernet-switching vlan member Vlan11
set vlans Vlan22 vlan-id 22
set interfaces ge-1/0/1 family ethernet-switching interface-mode trunk
set interfaces ge-1/0/1 family ethernet-switching vlan member Vlan22
set interfaces sxe-0/0/1.0 family ethernet-switching interface-mode trunk
set interfaces sxe-0/0/1.0 family ethernet-switching vlan member Vlan22
```

Step-by-Step Procedure

To configure the Packet Forwarding Engine interfaces:

1. Connect to the JCP.

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

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

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

3. Configure the Packet Forwarding Engine's LAN-side front panel port, and add it to the LAN-side VLAN.

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

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

4. Configure the Packet Forwarding Engine's LAN-side internal-facing interface as a trunk port, and add it to the LAN-side VLAN.

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

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

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

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

6. Configure the Packet Forwarding Engine's WAN-side front panel port as a trunk port, and add it to the WAN-side VLAN.

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

```
user@jcp# set interfaces ge-1/0/1 family ethernet-switching interface-mode trunk
user@jcp# set interfaces ge-1/0/1 family ethernet-switching vlan members Vlan22
```

7. Configure the Packet Forwarding Engine's WAN-side internal-facing interface as a trunk port, and add it to the WAN-side VLAN.

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

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

8. Commit the configuration.

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

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

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

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

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

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

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

```
}

```

Creating the Service Chain

Step-by-Step Procedure

To configure the VNF interfaces and create the service chain:

1. Configure *vnf-name1*'s LAN-side interface as a Layer 3 interface, and map it to the LAN-side NIC interface. Include the virtual function (VF) setting to specify direct NIC-to-VM connectivity. VNF must use the interfaces from eth0 through eth9.

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

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

2. Configure the *vnf-name1*'s WAN-side interface from the eth3 VNF interface as shown in [Figure 10 on page 90](#).

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

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

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

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

Related Documentation

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

Example: Configuring Service Chaining Using a Custom Bridge

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

- [Requirements on page 95](#)
- [Overview on page 95](#)
- [Configuration on page 95](#)
- [Verifying the Configuration on page 97](#)

Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

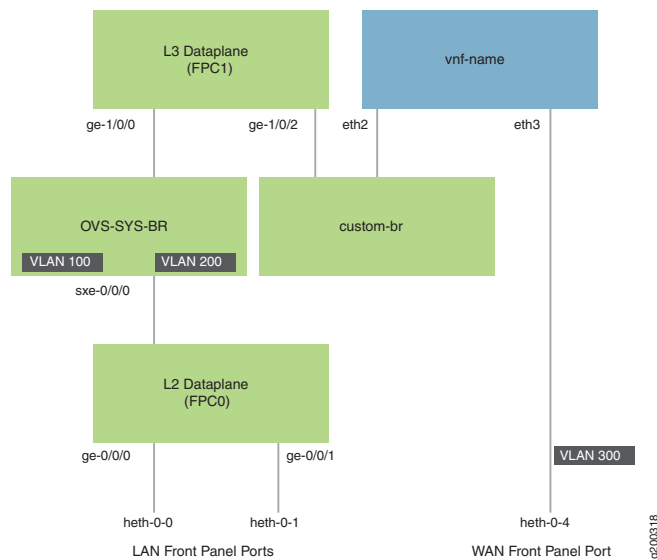
Overview

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

Topology

This example uses the topology shown in [Figure 11 on page 95](#).

Figure 11: Service Chaining Using a Custom Bridge



Configuration

- [Create VLANs and the Custom Bridge on page 96](#)
- [Map the Interfaces on page 96](#)
- [Configure the Layer 2 Datapath on page 96](#)
- [Configure the Layer 3 Datapath on page 97](#)
- [Configure the VNF on page 97](#)

Create VLANs and the Custom Bridge

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.


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


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

Map the Interfaces

Step-by-Step Procedure

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


```
user@host# set vmhost virtualization-options interfaces ge-0/0/0 mapping interface heth-0-0
```
2. Map the heth-0-1 physical port to the FPC0 interface.


```
user@host# set vmhost virtualization-options interfaces ge-0/0/1 mapping interface heth-0-1
```
3. Map the FPC1 interface ge-1/0/1 to the system bridge OVS.


```
user@host# set vmhost virtualization-options interfaces ge-1/0/1
```
4. Map the FPC1 interface ge-1/0/2 to the custom bridge.


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

Configure the Layer 2 Datapath

Step-by-Step Procedure

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


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


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


Configure the Layer 3 Datapath

Step-by-Step Procedure

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


```
user@host# set interfaces ge-1/0/1 vlan-tagging
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 100
user@host# set interfaces ge-1/0/1 unit 0 family inet address 192.0.2.1/24
```
2. Configure VLAN tagging on ge-1/0/2:


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

Configure the VNF

Step-by-Step Procedure

1. Launch the VNF:


```
user@host# set virtual-network-functions vnf-name image /var/public/centos-updated1.img
user@host# set virtual-network-functions vnf-name image image-type raw
```
2. Specify the number of CPUs required for the VNF:


```
user@host# set virtual-network-functions vnf-name virtual-cpu count 1
```
3. Pin a virtual CPU to a physical CPU:


```
user@host# set virtual-network-functions vnf-name virtual-cpu 0 physical-cpu 2
```
4. Create a VNF interface on the custom OVS bridge:


```
user@host# set virtual-network-functions vnf-name interfaces eth2 mapping vlan members
custom-br
```
5. Attach a VNF interface to a physical interface by using the SR-IOV virtual function:


```
user@host# set virtual-network-functions vnf-name interfaces eth3 mapping interface
heth-0-4 virtual-function vlan-id 300
```
6. Specify the memory allocation for the VNF:


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

Verifying the Configuration

- [Verify the Control Plane Configuration on page 97](#)
- [Verify the Data Plane Configuration on page 98](#)

Verify the Control Plane Configuration

Purpose Verify the control plane configuration:

Action To verify the control plane configuration:

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

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

| Routing instance | VLAN name | Tag | Interfaces |
|------------------|-----------|-----|------------|
| vmhost | custom-br | | ge-1/0/2.0 |
| vmhost | vlan200 | 200 | |

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

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

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

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

To view more details of the VNF:

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

```
Virtual Network Function Information
```

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

Verify the Data Plane Configuration

Purpose Verify the data plane configuration.

Action To verify the data plane configuration:

- Verify the status of the physical ports.

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

```

Physical interface: heth-0-0, Enabled, Physical link is Up
Link-level type: Ethernet, Media type: Copper, MTU: 9192, Speed: 1Gbps, Duplex:
Full-duplex, Auto-negotiation: Enabled
Device flags   : Present Running
Current address: 00:00:5e:00:53:8d, Hardware address: 00:00:5e:00:53:8d
Input packets : 272469
Output packets: 674
MAC statistics:
  Input bytes: 17438016, Input packets: 272469, Output bytes: 48658, Output
packets: 674
VF statistics:
  VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
    Input bytes: 17433984, Input packets: 272406, Output bytes: 48658, Output
packets: 674, Multicast packets: 272406
  VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
  VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
  VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0

```

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

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

For example:

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

```

Physical interface: ge-0/0/0, Enabled, Physical link is Up
Interface index: 144, SNMP ifIndex: 518
Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 1000mbps,
BPDU Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,

MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
Flow control: Enabled
Device flags   : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
Link flags     : None
CoS queues    : 8 supported, 8 maximum usable queues
Current address: 00:00:5e:00:53:43, Hardware address: 00:00:5e:00:53:43
Last flapped  : 2018-04-18 05:38:22 UTC (1d 22:28 ago)
Statistics last cleared: Never
Input rate      : 0 bps (0 pps)
Output rate     : 0 bps (0 pps)
Input errors: 0, Output errors: 0
Active alarms : None
Active defects : None
PCS statistics                               Seconds
  Bit errors                                0
  Errored blocks                             0
Ethernet FEC statistics                       Errors
  FEC Corrected Errors                       0
  FEC Uncorrected Errors                     0
  FEC Corrected Errors Rate                   0
  FEC Uncorrected Errors Rate                 0

```

```

PRBS Statistics : Disabled
Interface transmit statistics: Disabled

Logical interface ge-0/0/0.0 (Index 333) (SNMP ifIndex 524)
Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
Input packets : 125892
Output packets: 22
Protocol eth-switch, MTU: 9192
Flags: Is-Primary

```

Example: Configuring Service Chaining for LAN-WAN Routing

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

- [Requirements on page 100](#)
- [Overview on page 100](#)
- [Configuration on page 101](#)
- [Verification on page 102](#)

Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

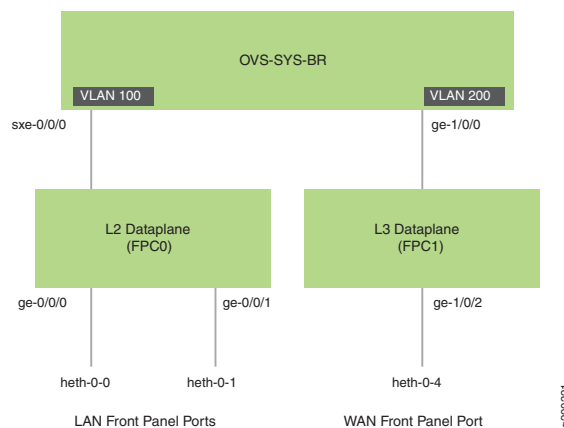
Overview

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

Topology

This example uses the topology shown in [Figure 11 on page 95](#).

Figure 12: Service Chaining Using a Custom Bridge



Configuration

- [Map the Interfaces on page 101](#)
- [Configure the Layer 2 Datapath on page 101](#)
- [Configure the Layer 3 Datapath on page 101](#)

Map the Interfaces

Step-by-Step Procedure

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

```
user@host# set vmhost virtualization-options interfaces ge-0/0/0 mapping interface heth-0-0
```
2. Map the heth-0-1 physical port to the FPC0 interface.

```
user@host# set vmhost virtualization-options interfaces ge-0/0/1 mapping interface heth-0-1
```
3. Map the FPC1 interface ge-1/0/2 to the physical port heth-0-4.

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

Configure the Layer 2 Datapath

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

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

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

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

Configure the Layer 3 Datapath

Step-by-Step Procedure

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

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

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

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

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

Verification

- [Verifying the Status of the Interfaces on page 102](#)

Verifying the Status of the Interfaces

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

Action To verify the status of the interfaces:

- Verify the status of the physical ports.

```
user@host> show interfaces heth-0-0 statistics
Physical interface: heth-0-0, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Copper, MTU: 9192, Speed: 1Gbps, Duplex:
  Full-duplex, Auto-negotiation: Enabled
  Device flags   : Present Running
  Current address: 00:00:5e:00:53:8d, Hardware address: 00:00:5e:00:53:8d
    Input packets : 272469
    Output packets: 674
  MAC statistics:
    Input bytes: 17438016, Input packets: 272469, Output bytes: 48658, Output
  packets: 674
  VF statistics:
    VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
    Input bytes: 17433984, Input packets: 272406, Output bytes: 48658, Output
packets: 674, Multicast packets: 272406
    VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
    VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
  Multicast packets: 0
```

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

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

For example:

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

```

Physical interface: ge-0/0/0, Enabled, Physical link is Up
Interface index: 144, SNMP ifIndex: 518
Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 1000mbps,
BPDU Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,

MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
Flow control: Enabled
Device flags   : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
Link flags     : None
CoS queues    : 8 supported, 8 maximum usable queues
Current address: 00:00:5e:00:53:43, Hardware address: 00:00:5e:00:53:43
Last flapped   : 2018-04-18 05:38:22 UTC (2d 10:07 ago)
Statistics last cleared: Never
Input rate   : 0 bps (0 pps)
Output rate  : 0 bps (0 pps)
Input errors: 0, Output errors: 0
Active alarms : None
Active defects : None
PCS statistics                               Seconds
  Bit errors                                0
  Errored blocks                            0
Ethernet FEC statistics                       Errors
  FEC Corrected Errors                      0
  FEC Uncorrected Errors                    0
  FEC Corrected Errors Rate                  0
  FEC Uncorrected Errors Rate                0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled

Logical interface ge-0/0/0.0 (Index 333) (SNMP ifIndex 524)
Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
Input packets : 147888
Output packets: 22
Protocol eth-switch, MTU: 9192
Flags: Is-Primary

```

Example: Configuring Cross Connect on NFX150 Devices

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

- [Requirements on page 104](#)
- [Overview on page 104](#)
- [Configuration on page 105](#)
- [Verifying the Configuration on page 107](#)

Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

Overview

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



NOTE: This feature does not support unidirectional traffic flow.

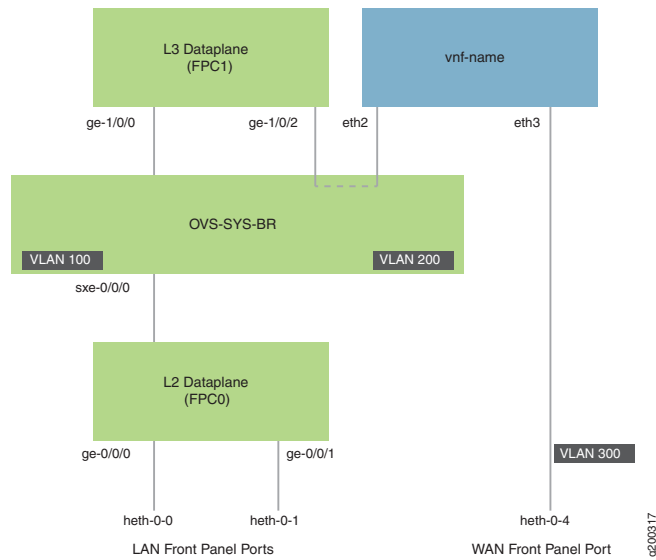
The **Cross-connect** feature supports the following:

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

Topology

This example uses the topology shown in [Figure 11 on page 95](#).

Figure 13: Configuring Cross-Connect



Configuration

- Create VLANs on page 105
- Map the Interfaces on page 105
- Configure the Layer 2 Datapath on page 106
- Configure the Layer 3 Datapath on page 106
- Configure the VNF on page 106
- Configure Cross-Connect on page 107

Create VLANs

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

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

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

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

Map the Interfaces

Step-by-Step Procedure

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

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

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

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

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

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

Configure the Layer 2 Datapath

Step-by-Step Procedure

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

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

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

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

Configure the Layer 3 Datapath

Step-by-Step Procedure

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

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

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

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

Configure the VNF

Step-by-Step Procedure

1. Launch the VNF:

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

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

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

3. Pin a virtual CPU to a physical CPU:

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

4. Create a host VLAN:

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

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

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

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

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

7. Specify the memory allocation for the VNF:

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

Configure Cross-Connect

Step-by-Step Procedure

1. Configure cross-connect:

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

Verifying the Configuration

- [Verify the Control Plane Configuration on page 107](#)
- [Verify the Data Plane Configuration on page 108](#)

Verify the Control Plane Configuration

Purpose Verify the control plane configuration:

Action To verify the control plane configuration:

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

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

| Routing instance | VLAN name | Tag | Interfaces |
|------------------|-----------|-----|------------|
| vmhost | custom-br | | ge-1/0/2.0 |
| vmhost | vlan200 | 200 | |

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

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

| ID | Name | State | Liveliness |
|----|----------|---------|------------|
| 2 | vnf-name | Running | alive |

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

To view more details of the VNF:

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

```
Virtual Network Function Information
```

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

Verify the Data Plane Configuration

Purpose Verify the data plane configuration.

Action To verify the data plane configuration:

- Verify the status of the physical ports.

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

```
Physical interface: heth-0-0, Enabled, Physical link is Up
  Link-level type: Ethernet, Media type: Copper, MTU: 9192, Speed: 1Gbps, Duplex:
  Full-duplex, Auto-negotiation: Enabled
  Device flags   : Present Running
  Current address: 00:00:5e:00:53:8d, Hardware address: 00:00:5e:00:53:8d
Input packets : 311143
Output packets: 674
  MAC statistics:
    Input bytes: 19913152, Input packets: 311143, Output bytes: 48658, Output
  packets: 674
  VF statistics:
    VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
    Input bytes: 19909120, Input packets: 311080, Output bytes: 48658, Output
packets: 674, Multicast packets: 311080
    VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
```

```

    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
    VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
    VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
    Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0

```

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

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

For example:

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

```

Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 144, SNMP ifIndex: 518
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 1000mbps, BPDU
Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled, Flow
control: Enabled
  Device flags      : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags       : None
  CoS queues       : 8 supported, 8 maximum usable queues
  Current address: 00:00:5e:00:53:43, Hardware address: 00:00:5e:00:53:43
  Last flapped    : 2018-04-18 05:38:22 UTC (6d 00:28 ago)
  Statistics last cleared: Never
  Input rate      : 0 bps (0 pps)
  Output rate     : 0 bps (0 pps)
  Input errors: 0, Output errors: 0
  Active alarms  : None
  Active defects : None
  PCS statistics
    Bit errors          Seconds
    Errored blocks      0
  Ethernet FEC statistics
    FEC Corrected Errors      0
    FEC Uncorrected Errors    0
    FEC Corrected Errors Rate 0
    FEC Uncorrected Errors Rate 0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

Logical interface ge-0/0/0.0 (Index 333) (SNMP ifIndex 524)
  Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
  Input packets : 311115
  Output packets: 22
  Protocol eth-switch, MTU: 9192
  Flags: Trunk-Mode

```

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

```

Physical interface: ge-1/0/2, Enabled, Physical link is Up
  Interface index: 158, SNMP ifIndex: 536

```

```

Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

Remote fault: Online
Device flags : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
CoS queues : 8 supported, 8 maximum usable queues
Current address: 00:00:5e:00:53:5d, Hardware address: 00:00:5e:00:53:5d
Last flapped : 2018-04-23 06:03:29 UTC (1d 00:04 ago)
Statistics last cleared: Never
Input rate : 0 bps (0 pps)
Output rate : 0 bps (0 pps)
Input errors: 0, Output errors: 0
Active alarms : None
Active defects : None
PCS statistics                               Seconds
  Bit errors                                0
  Errored blocks                            0
Ethernet FEC statistics                       Errors
  FEC Corrected Errors                      0
  FEC Uncorrected Errors                    0
  FEC Corrected Errors Rate                  0
  FEC Uncorrected Errors Rate                0
PRBS Statistics : Disabled
Interface transmit statistics: Disabled

Logical interface ge-1/0/2.0 (Index 342) (SNMP ifIndex 538)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.200 ] Encapsulation: ENET2
  Input packets : 0
  Output packets: 0
  Security: Zone: untrust
  Allowed host-inbound traffic : dns dhcp tftp https
  Protocol inet, MTU: 1500
  Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 0, Curr new
hold cnt: 0, NH drop cnt: 0
  Flags: Sendbroadcast-pkt-to-re
  Addresses, Flags: Is-Preferred Is-Primary
  Destination: 203.0.113/24, Local: 203.0.113.2, Broadcast: 203.0.113.255

  Protocol multiservice, MTU: Unlimited

Logical interface ge-1/0/2.32767 (Index 343) (SNMP ifIndex 545)
  Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2

  Input packets : 0
  Output packets: 0
  Security: Zone: Null
  Protocol multiservice, MTU: Unlimited
  Flags: None

```

- Related Documentation**
- [Example: Configuring Cross-Connect Using a Custom Bridge on NFX150 Devices on page 113](#)

Example: Configuring Service Chaining for LAN Routing

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

- [Requirements on page 111](#)
- [Overview on page 111](#)
- [Configuration on page 111](#)

Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

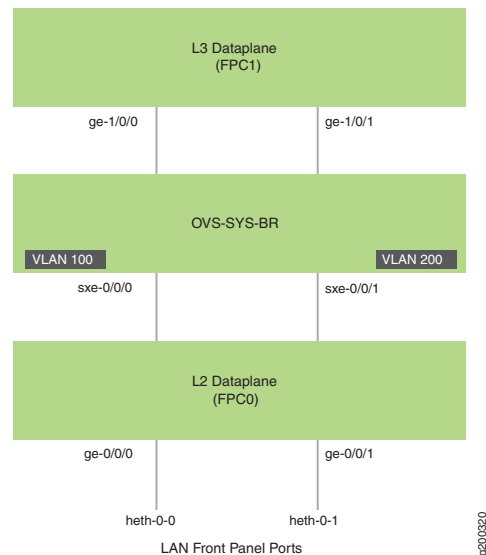
Overview

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

Topology

This example uses the topology shown in [Figure 11 on page 95](#).

Figure 14: Service Chaining for LAN Routing



Configuration

- [Map the Interfaces on page 112](#)
- [Configure the Layer 2 Datapath on page 112](#)
- [Configure the Layer 3 Datapath on page 112](#)

Map the Interfaces

Step-by-Step Procedure

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

```
user@jcp# set vmhost virtualization-options interfaces ge-0/0/1 mapping interface heth-0-0
```
2. Map the heth-0-1 physical port to the FPC0 interface.

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

Configure the Layer 2 Datapath

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

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

```
user@jcp# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
user@jcp# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan100
user@jcp# set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode trunk
user@jcp# set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100
```
3. Configure the internal-facing interfaces as trunk ports and add them to the LAN-side VLAN. The internal-facing interfaces are typically trunk ports, as they must support traffic from multiple front panel ports and VLANs.

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

Configure the Layer 3 Datapath

Step-by-Step Procedure

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

```
user@jcp# set interfaces ge-1/0/1 vlan-tagging
user@jcp# set interfaces ge-1/0/1 unit 0 vlan-id 100
user@jcp# set interfaces ge-1/0/1 unit 0 family inet address 192.0.2.1/24
```
2. Configure VLAN tagging on ge-1/0/2:

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


- Related Documentation**
- [Example: Configuring Service Chaining for LAN-WAN Routing on page 100](#)

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

This example shows how to configure cross-connect using a custom bridge on NFX150 devices.

- [Requirements on page 113](#)
- [Overview on page 113](#)
- [Configuration on page 114](#)
- [Verifying the Configuration on page 116](#)

Requirements

This example uses the following hardware and software components:

- NFX150 running Junos OS Release 18.1R1

Overview

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



NOTE: This feature does not support unidirectional traffic flow.

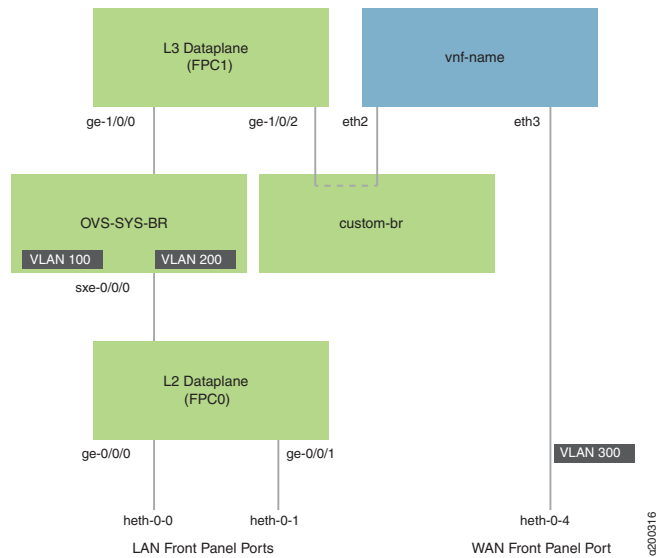
The **Cross-connect** feature supports the following:

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

Topology

This example uses the topology shown in [Figure 11 on page 95](#).

Figure 15: Configuring Cross-Connect



Configuration

- [Create VLANs on page 114](#)
- [Map the Interfaces on page 114](#)
- [Configure the Layer 2 Datapath on page 115](#)
- [Configure the Layer 3 Datapath on page 115](#)
- [Configure the VNF on page 115](#)
- [Configure Cross-Connect on page 116](#)

Create VLANs

Step-by-Step Procedure

1. Configure VLANs for the LAN-side interfaces.

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

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

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

Map the Interfaces

Step-by-Step Procedure

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

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

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

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

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

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

Configure the Layer 2 Datapath

Step-by-Step Procedure

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

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

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

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

Configure the Layer 3 Datapath

Step-by-Step Procedure

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

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

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

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

Configure the VNF

Step-by-Step Procedure

1. Launch the VNF:

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

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

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

3. Pin a virtual CPU to a physical CPU:

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

4. Create a host VLAN:

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

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

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

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

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

7. Specify the memory allocation for the VNF:

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

Configure Cross-Connect

Step-by-Step Procedure

1. Configure cross-connect:

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

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

Verifying the Configuration

- [Verify the Control Plane Configuration on page 116](#)
- [Verify the Data Plane Configuration on page 117](#)

Verify the Control Plane Configuration

Purpose Verify the control plane configuration:

Action To verify the control plane configuration:

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

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

| Routing instance | VLAN name | Tag | Interfaces |
|------------------|-----------|-----|------------|
| vmhost | custom-br | | ge-1/0/2.0 |
| vmhost | vlan200 | 200 | |

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

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

| ID | Name | State | Liveliness |
|----|----------|---------|------------|
| 2 | vnf-name | Running | alive |

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

To view more details of the VNF:

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

```
Virtual Network Function Information
```

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

Verify the Data Plane Configuration

Purpose Verify the data plane configuration.

Action To verify the data plane configuration:

- Verify the status of the physical ports.

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

```
Physical interface: heth-0-0, Enabled, Physical link is Up
Link-level type: Ethernet, Media type: Copper, MTU: 9192, Speed: 1Gbps, Duplex:
Full-duplex, Auto-negotiation: Enabled
Device flags   : Present Running
Current address: 00:00:5e:00:53:8d, Hardware address: 00:00:5e:00:53:8d
Input packets : 311143
Output packets: 674
MAC statistics:
Input bytes: 19913152, Input packets: 311143, Output bytes: 48658, Output
packets: 674
VF statistics:
VF Number: 0, PCI Address: 0000:02:10:1, Mapped to: ge-0/0/0
Input bytes: 19909120, Input packets: 311080, Output bytes: 48658, Output
packets: 674, Multicast packets: 311080
VF Number: 1, PCI Address: 0000:02:10:5, Mapped to: ge-0/0/0
Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
VF Number: 2, PCI Address: 0000:02:11:1, Mapped to: ge-0/0/0
Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0
```

```

VF Number: 3, PCI Address: 0000:02:11:5, Mapped to: ge-0/0/0
  Input bytes: 0, Input packets: 0, Output bytes: 0, Output packets: 0,
Multicast packets: 0

```

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

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

For example:

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

```

Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 144, SNMP ifIndex: 518
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 1000mbps, BPDU
Error: None, Loop Detect PDU Error: None, Ethernet-Switching Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled, Flow
control: Enabled
  Device flags      : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags       : None
  CoS queues       : 8 supported, 8 maximum usable queues
  Current address: 00:00:5e:00:53:43, Hardware address: 00:00:5e:00:53:43
  Last flapped    : 2018-04-18 05:38:22 UTC (6d 00:28 ago)
  Statistics last cleared: Never
Input rate      : 0 bps (0 pps)
Output rate     : 0 bps (0 pps)
  Input errors: 0, Output errors: 0
  Active alarms  : None
  Active defects : None
  PCS statistics                               Seconds
    Bit errors                                0
    Errored blocks                            0
  Ethernet FEC statistics                       Errors
    FEC Corrected Errors                      0
    FEC Uncorrected Errors                    0
    FEC Corrected Errors Rate                  0
    FEC Uncorrected Errors Rate                0
  PRBS Statistics : Disabled
  Interface transmit statistics: Disabled

Logical interface ge-0/0/0.0 (Index 333) (SNMP ifIndex 524)
  Flags: Up SNMP-Traps 0x24024000 Encapsulation: Ethernet-Bridge
Input packets : 311115
Output packets: 22
  Protocol eth-switch, MTU: 9192
  Flags: Trunk-Mode

```

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

```

Physical interface: ge-1/0/2, Enabled, Physical link is Up
  Interface index: 158, SNMP ifIndex: 536
  Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, BPDU Error: None, Loop Detect PDU Error: None,
  Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,

```

```

Remote fault: Online
Device flags : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
CoS queues : 8 supported, 8 maximum usable queues
Current address: 00:00:5e:00:53:5d, Hardware address: 00:00:5e:00:53:5d
Last flapped : 2018-04-23 06:03:29 UTC (1d 00:04 ago)
Statistics last cleared: Never
Input rate : 0 bps (0 pps)
Output rate : 0 bps (0 pps)
Input errors: 0, Output errors: 0
Active alarms : None
Active defects : None
PCS statistics
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-1/0/2.0 (Index 342) (SNMP ifIndex 538)
Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.200 ] Encapsulation: ENET2
Input packets : 0
Output packets: 0
Security: Zone: untrust
Allowed host-inbound traffic : dns dhcp tftp https
Protocol inet, MTU: 1500
Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 0, Curr new
hold cnt: 0, NH drop cnt: 0
Flags: 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

Protocol multiservice, MTU: Unlimited

Logical interface ge-1/0/2.32767 (Index 343) (SNMP ifIndex 545)
Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2

Input packets : 0
Output packets: 0
Security: Zone: Null
Protocol multiservice, MTU: Unlimited
Flags: None

```

Related Documentation • [Example: Configuring Cross Connect on NFX150 Devices on page 103](#)

CHAPTER 7

Troubleshooting

- [Recovering the Root Password for NFX150 on page 121](#)
- [Troubleshooting Interfaces on NFX150 Devices on page 124](#)

Recovering the Root Password for NFX150

The root password on your Junos OS-enabled router 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 your asynchronous terminal emulation application (such as Microsoft Windows Hyperterminal) and select the appropriate port to use.

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 (if necessary) plugging the power cords into the device's power supply, or turning on the power to the device by switching on the AC power outlet 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. For example:

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

15. At the first prompt, enter the new root password:

```
New password: ABC123
```

16. At the second prompt, re-enter the new root password.

```
Retype new password: ABC123
```

17. After you have finished configuring the password, commit the configuration.

```
[edit]
root# commit
commit complete
```

18. Exit the configuration mode in the CLI.

```
[edit]
root@host# exit
root@host>
```

19. Exit the 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 NFX150 Devices

- [Monitoring Interface Status and Traffic on NFX Devices on page 124](#)

Monitoring Interface Status and Traffic on NFX Devices

Purpose View the interface status to monitor interface bandwidth utilization and traffic statistics.

Action To view interface status for an interface, enter **show interfaces *interface-name***

For example:

```
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
```

CHAPTER 8

Configuration Statements

- [cross-connect on page 126](#)
- [forwarding-options on page 127](#)
- [interfaces on page 128](#)
- [phone-home on page 129](#)
- [security on page 131](#)
- [virtualization-options on page 134](#)
- [vlans on page 135](#)
- [vmhost on page 136](#)

cross-connect

| | |
|---------------------------------|---|
| Syntax | <pre>cross-connect <i>cc-name</i> { virtual-interface <i>int-name</i> { vlan-id <i>vlan-id</i>; } virtual-network-function <i>vnf-name</i> { interface <i>interface</i>; vlan-id <i>vlan-id</i>; } }</pre> |
| Hierarchy Level | [edit vmhost] |
| Release Information | Statement introduced in Junos OS Release 18.1R1 for the NFX150 device. |
| Description | Enables traffic switching between any two OVS interfaces such as virtual interfaces or physical interfaces that are connected to the OVS. |
| Options | <p>virtual-interface—Configures virtual interface as switch port.</p> <p>virtual-network-function—Configures virtual network function as switch port.</p> <p>vlan-id —IEEE 802.1q VLAN identifier for VLAN that ranges from 1 to 4094.</p> <p>int-name—Name of the virtual interface.</p> <p>vnf-name—Name of the virtual network functions.</p> |
| Required Privilege Level | set vmhost—To view this statement in the configuration. |
| Related Documentation | <ul style="list-style-type: none">• show vmhost connections on page 165 |

forwarding-options

```

Syntax forwarding-options {
  analyzer analyzer-name {
    output;
    input {
      egress {
        virtual-network-function vnf-name {
          interface int-name;
        }
      }
    }
    ingress {
      virtual-network-function vnf-name {
        interface int-name;
      }
    }
  }
}

```

Hierarchy Level [edit vmhost]

Release Information Statement introduced in Junos OS Release 18.1R1 for the NFX150 devices.

Description Configure options to control packet forwarding.

Options

- analyzer**—Analyzer options
- analyzer-name***—Name of the analyzer.
- input**—Port for incoming packets.
- output**—Port for outgoing mirrored packets.
- egress**—Port to monitor outgoing traffic.
- ingress**—Port to monitor incoming traffic.
- interface**—Interface configured for virtual network functions.

Required Privilege Level set vmhost—To view this statement in the configuration.

Related Documentation

- [vmhost on page 136](#)

interfaces

```
Syntax interfaces int-name {
  ether-options {
    speed {
      auto-negotiation;
    }
  }
  dsl-sfp-options {
    adsl-options {
      encap;
      vci;
      vpi;
    }
    vdsl-options {
      carrier;
      profile;
    }
  }
}
```

Hierarchy Level [edit vmhost]

Release Information Statement introduced in Junos OS Release 18.1R1 for the NFX150 Network Services Platform.

Description Configures interfaces on NFX150 devices.

Options

- ether-options**—Configure options for Ethernet interface.
- dsl-sfp-options**—Configure options for DSL SFP interface.
- speed**—Specify the speed of an Ethernet interface.
- adsl-options**—Configure ADSL options for SFP interface.
- vdsl-options**—Configure VDSL options for SFP interface.
- auto-negotiation**—Enable auto-negotiation. With auto-negotiation enabled, the interface automatically detects the speed in which to operate.

Required Privilege Level set vmhost—To view this statement in the configuration.

Related Documentation

- [vmhost on page 136](#)

phone-home

Syntax

```

phone-home {
  server;
  upgrade-image-before-configuration;
  traceoptions {
    no-remote-trace;
    file {
      file-name;
      size file-size;
    }
    flag {
      all;
      config;
      function;
      misc;
      socket;
      state-machine;
    }
  }
}

```

Hierarchy Level [edit system services]

Release Information Statement introduced in Junos OS Release 15.1X53-D45 for the NFX250 device. Statement introduced in Junos OS Release 18.1R1 for the NFX150 device.

Description An option that is used for initial boot up and configuration of the device when the client device is switched on.

The default configuration for phone-home is as follows:

```

user@jdm# set system phone-home server https://redirect.juniper.net
user@jdm# set system phone-home upgrade-image-before-configuration

```

Options **server**—Name of the server.

upgrade-image-before-configuration—Upgrades the image before applying the configuration received from the Network Activator.

traceoptions—Options that are available for the phone-home trace operation

no-remote-trace—

file-name—Name of file in which the trace information is available

file-size—Maximum size of a trace file.

flag—Parameters that can be specified for the phone-home trace operation.

Required Privilege Level system—To view this statement in the configuration.

security


```

Syntax security {
  ike {
    gateway gateway-name {
      aaa {
        client {
          username user-name password password;
        }
      }
      local-identity {
        hostname;
        distinguished-name;
        inet;
        inet6;
        key-id string-key-id;
        user-at-hostname;
      }
      remote-identity {
        hostname;
        distinguished-name;
        inet;
        inet6;
        key-id string-key-id;
        user-at-hostname;
      }
    }
  }
  ipsec {
    proposal ipsec-proposal-name {
      protocol {
        esp;
      }
      authentication-algorithm {
        hmac-md5-96;
        hmac-sha-256-128;
        hmac-sha1-96;
      }
      encryption-algorithm {
        3des-cbc;
        aes-128-cbc;
        aes-192-cbc;
        aes-256-cbc;
        des-cbc;
      }
      lifetime-seconds lifetime-in-seconds;
    }
    policy ipsec-policy-name {
      perfect-forward-secrecy {
        keys {
          group1;
          group14;
          group2;
        }
      }
    }
  }
}

```

```
        group5;
    }
}
proposals proposal-name;
}
vpn vpn-name {
    bind-interface interface-name;
    ike {
        gateway remote-gateway-name;
        ipsec-policy ipsec-policy-name;
    }
    traffic-selector {
        traffic-selector-name1 {
            local-ip local-traffic-selector-ip-address;
            remote-ip remote-traffic-selector-ip-address;
        }
        traffic-selector-name2 {
            local-ip local-traffic-selector-ip-address;
            remote-ip remote-traffic-selector-ip-address;
        }
    }
}
establish-tunnels {
    immediately;
    on-traffic;
}
}
policies {
    from-zone {
        from-zone-name {
            to-zone to--zone-name {
                policy policy-name {
                    match {
                        source-address source-address;
                        destination-address destination-address;
                        application application;
                        then permit;
                    }
                }
            }
        }
    }
}
trust {
    to-zone to--zone-name;
    trust;
    untrust;
}
untrust {
    to-zone to--zone-name;
    trust;
    untrust;
}
}
}
zones {
    security-zone {
        security-zone-name {
```


virtualization-options

| | |
|---------------------------------|---|
| Syntax | <pre>virtualization-options { interfaces <i>virtual-int-name</i> { mapping { interfaces <i>physical-int-name</i>; } } }</pre> |
| Hierarchy Level | [edit vmhost] |
| Release Information | Statement introduced in Junos OS Release 18.1R1 for the NFX150 Network Services Platform. |
| Description | Configure virtualization options on NFX150 devices. |
| Options | <p>interfaces—Configures virtual interfaces. Virtual layer 2 interfaces range from ge-0/0/0 to ge-0/0/3 for NFX150 devices without an expansion module and ge-0/0/0 to ge-0/0/11 for NFX150 devices with an expansion module. Virtual layer 3 interfaces range from ge-1/0/1 to ge-1/0/9 for NFX150 devices.</p> <p><i>virtual-int-name</i>—Name of the virtual interface.</p> <p>mapping—Maps any physical ports to virtual interfaces.</p> <p><i>physical-int-name</i>—Name of the physical ports. Physical ports on the front panel of the NFX150 device range from heth-0-0 to heth-0-5. Physical ports on the expansion module of the NFX150-S1 device ranges from heth-1-0 to heth-1-7.</p> |
| | <p>.....</p> <p> NOTE: The mapping option is not required when the interface is mapped to the OVS.</p> <p>.....</p> |
| Required Privilege Level | set vmhost—To view this statement in the configuration. |
| Related Documentation | <ul style="list-style-type: none"> • vmhost on page 136 |

vlan

| | |
|---------------------------------|--|
| Syntax | <pre>vlan <i>vlan-name</i> { vlan-id <i>vlan-id</i>; vlan-id-list <i>vlan-id-list</i>; }</pre> |
| Hierarchy Level | [edit vmhost] |
| Release Information | Statement introduced in Junos OS Release 18.1R1 for the NFX150 Network Services Platform. |
| Description | Configures VLAN on NFX150 devices. |
| Options | <p>vlan—Name of the VLAN. The name can include letters, numbers, hyphens (-), and periods (.) and can contain up to 255 characters.</p> <p>forwarding-options—Configure options to control packet forwarding.</p> <p>interfaces—Configure interfaces.</p> <p>virtualization-options—Configure virtualization options.</p> <p>vlan—Configure VLAN.</p> |
| Required Privilege Level | set vmhost—To view this statement in the configuration. |
| Related Documentation | <ul style="list-style-type: none"> • vmhost on page 136 |

vmhost

```

Syntax  vmhost {
        cross-connect cc-name {
            virtual-interface int-name {
                vlan-id vlan-id;
            }
            virtual-network-function vnf-name {
                interface interface;
                vlan-id vlan-id;
            }
        }
        forwarding-options {
            analyzer analyzer-name {
                output;
                input {
                    egress {
                        virtual-network-function vnf-name {
                            interface int-name;
                        }
                    }
                    ingress {
                        virtual-network-function vnf-name {
                            interface int-name;
                        }
                    }
                }
            }
        }
        interfaces int-name {
            ether-options {
                speed {
                    auto-negotiation;
                }
            }
        }
        virtualization-options {
            interfaces virtual-int-name {
                mapping {
                    interfaces physical-int-name;
                }
            }
        }
        vlans vlan-name {
            vlan-id vlan-id;
            vlan-id-list vlan-id-list;
        }
    }

```

Hierarchy Level [\[edit\]](#)

Release Information Statement introduced in Junos OS Release 18.1R1 for the NFX150 Network Services Platform.

Description Configures vmhost on NFX150 devices.

Options **cross-connect** —Enables traffic switching between any two OVS interfaces such as virtual interfaces or physical interfaces that are connected to the OVS.

forwarding-options—Configure options to control packet forwarding.

interfaces—Configure interfaces.

virtualization-options —Configure virtualization options.

vlan—Configure VLAN.

Required Privilege Level set vmhost—To view this statement in the configuration.

Related Documentation •

CHAPTER 9

Operational Commands

- request vmhost cleanup
- request vmhost file-copy
- request vmhost halt
- 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 status
- show vmhost storage
- show vmhost uptime
- show vmhost version
- show vmhost vlans

request vmhost cleanup

| | |
|---------------------------------|--|
| Syntax | <code>request vmhost cleanup</code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | 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. |
| Required Privilege Level | maintenance |
| Related Documentation | <ul style="list-style-type: none">• vmhost on page 136 |
| Output Fields | When you enter this command, you are provided feedback on the status of your request. |

request vmhost file-copy

| | |
|---------------------------------|---|
| Syntax | <code>request vmhost file-copy (crash log) from-jnode <i>host file-name</i> to-vjunos <i>host file-name</i></code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Copy crash files or log files from the host OS to Junos OS. You can use these files for analysis and debugging purposes. |
| Options | <ul style="list-style-type: none"> <code>crash</code>—Files in <code>/var/crash</code> on the host. <code>from-jnode <i>filename</i></code>—Name of the host file to be copied. <code>log</code>—Files in <code>/var/log</code> on the host. <code>to-vjunos <i>filename</i></code>—Name of the Junos OS file to which the host file is copied. |
| Additional Information | 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. |
| Required Privilege Level | maintenance |
| Related Documentation | <ul style="list-style-type: none"> vmhost on page 136 |
| List of Sample Output | request vmhost file-copy on page 141 |

Sample Output

request vmhost file-copy

```
user@host> request vmhost file-copy log from-jnode daemon.log to-vjunos /var/tmp
:/var/tmp # ls -lrt daemon.log
-rw-r--r--  1 root  wheel  1035126 Mar  4 20:33 daemon.log
```

request vmhost halt

| | |
|---------------------------------|--|
| Syntax | <code>request vmhost halt</code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Stop the host OS and Junos OS running on the device. |
| Required Privilege Level | maintenance |
| Related Documentation | <ul style="list-style-type: none">• vmhost on page 136 |
| List of Sample Output | request vmhost halt on page 142 |

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 power-off

Syntax `request vmhost power-off`

Release Information Command introduced in Junos OS Release 18.1R1 for NFX150 devices.



NOTE: `request vmhost power-on` is not supported on NFX150 devices.

Description Shut down the Junos OS software and the host OS.

Required Privilege Level maintenance

Related Documentation

- [vmhost on page 136](#)

List of Sample Output [request vmhost power-off on page 143](#)

Sample Output

request vmhost power-off

```
user@host> request vmhost power-off
Power-off the vmhost ? [yes,no] (no) yes

Initiating vmhost shutdown... ok
Initiating Junos shutdown... shutdown: [pid 3884]
Shutdown NOW!
ok

*** FINAL System shutdown message from root@host ***

System going down IMMEDIATELY
...
...
```

request vmhost reboot

| | |
|---------------------------------|--|
| Syntax | <code>request vmhost reboot</code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Reboot the Junos OS software and the host OS. |
| Required Privilege Level | maintenance |
| Related Documentation | <ul style="list-style-type: none">• vmhost on page 136 |
| List of Sample Output | request vmhost reboot on page 144 |

Sample Output

request vmhost reboot

```
user@host> request vmhost reboot
Reboot the vmhost ? [yes,no] (no) yes

warning: Rebooting re0
Initiating vmhost reboot... ok
Initiating Junos shutdown... shutdown: [pid 7273]
Shutdown NOW!
ok
Junos shutdown is in progress...

*** FINAL System shutdown message from root@ ***

System going down IMMEDIATELY

...
...
```


request vmhost software add

| | |
|---------------------------------|--|
| Syntax | <code>request vmhost software add <i>package-name</i> <no-validate> <reboot></code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Install Junos OS and host software packages on the device. |
| Options | <ul style="list-style-type: none"> no-validate—(Optional) When loading a software package or bundle with a different release, suppress the default behavior of the validate option. reboot—(Optional) After adding the software package or bundle, reboot the system. |
| Required Privilege Level | maintenance |
| Related Documentation | <ul style="list-style-type: none"> vmhost on page 136 |
| List of Sample Output | request vmhost software add on page 145 |

Sample Output

request vmhost software add

```

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 ...
..
..

```

show system visibility cpu

| | |
|---------------------------------|--|
| Syntax | <code>show system visibility cpu</code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Display details such as per CPU statistics, per CPU usage, and CPU pinning for a Junos OS platform. |
| Required Privilege Level | view |
| Related Documentation | <ul style="list-style-type: none"> • show system visibility host on page 148 • show system visibility memory on page 154 • show system visibility network on page 156 • show system visibility vnf on page 160 |
| List of Sample Output | show system visibility cpu on page 147 |
| Output Fields | Table 20 on page 146 lists the output fields for the <code>show system visibility cpu</code> command. Output fields are listed in the approximate order in which they appear. |

Table 20: show system visibility cpu Output Fields

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

Table 20: show system visibility cpu Output Fields (continued)

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

Sample Output

show system visibility cpu

```

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

CPU Usages
-----
CPU Id CPU Usage
-----
0      36.399999999999999
1      66.700000000000003
2       0.0
3       0.0

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

System Component          CPUs
-----
ovs-vswitchd             1

```

show system visibility host

Syntax `show system visibility host`

Release Information Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Description Displays details such as the host uptime, number of tasks, CPU statistics, list of disk partitions, disk usage, disk I/O statistics, list of network interfaces, and per port statistics for a Junos OS platform.

Required Privilege Level view

- Related Documentation**
- [show system visibility cpu on page 146](#)
 - [show system visibility memory on page 154](#)
 - [show system visibility network on page 156](#)
 - [show system visibility vnf on page 160](#)

List of Sample Output [show system visibility host on page 150](#)

Output Fields [Table 21 on page 148](#) lists the output fields for the **show system visibility host** command. Output fields are listed in the approximate order in which they appear.

Table 21: show system visibility host Output Fields

| Field Name | Field Description |
|--|--|
| Field for Host Uptime | |
| Uptime | The time the host has been operational. |
| Fields for Host Tasks | |
| Total | The total number of tasks. |
| Running | The total number of tasks running. |
| Sleeping | The total number of tasks in sleeping state. |
| Stopped | The total number of tasks that are stopped. |
| Zombie | The total number of zombie processes. |
| Fields for Host CPU Information | |
| User Time | The amount of user time, in seconds. |
| System Time | The amount of system time, in seconds. |

Table 21: show system visibility host Output Fields (continued)

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

Table 21: show system visibility host Output Fields (continued)

| Field Name | Field Description |
|--|---------------------------------|
| 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

```

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
veth00         active    00:00:5e:00:53:79
veth01         active    00:00:5e:00:53:87

```

```

veth10          active  00:00:5e:00:53:40
veth11          active  00:00:5e:00:53:65
virbr0          active  00:00:5e:00:53:83
virbr1          active  00:00:5e:00:53:6f

```

Host Port Statistics

```

-----
Interface Bytes Sent  Bytes Rcvd  Packets Sent Packets Rcvd Errors In Errors
Out Drops In Drops Out
-----
13_h_ge_1_0_0 11025      648         74           8           0           0
0 0
veth10 0      11673        0           82          0           0
12 0
veth11 11673     0           82          0           0
0 0
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
13_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 memory

| | |
|---------------------------------|---|
| Syntax | <code>show system visibility memory</code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Display the details about virtual memory and shared memory for a Junos OS platform. |
| Required Privilege Level | view |
| Related Documentation | <ul style="list-style-type: none"> • show system visibility cpu on page 146 • show system visibility host on page 148 • show system visibility network on page 156 • show system visibility vnf on page 160 |
| List of Sample Output | show system visibility memory on page 154 |
| Output Fields | Table 22 on page 154 lists the output fields for the show system visibility memory command. Output fields are listed in the approximate order in which they appear. |

Table 22: show system visibility memory Output Fields

| Field Name | Field Description |
|---|--|
| Fields for Memory Information—Virtual Memory | |
| Total | The total amount of available virtual memory, in kibibytes (KiBs). |
| Used | The total amount of used virtual memory, in kibibytes (KiBs). |
| Available | The total amount of available virtual memory, in kibibytes (KiBs). |
| Free | The total amount of free virtual memory, in kibibytes (KiBs). |
| Percent Used | The percentage of buffer virtual memory used. |
| Fields for Memory Information—Memory Limits | |
| User VNFs Total Memory Limit | 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

Syntax `show system visibility network`

Release Information Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Description Displays details such as the list of MAC addresses assigned to VNF interfaces, the list of internal IP addresses for VNFs, the list of virtual functions used by VNFs, and the list of VNF interfaces for a Junos OS platform.

Required Privilege Level view

Related Documentation

- [show system visibility cpu on page 146](#)
- [show system visibility host on page 148](#)
- [show system visibility memory on page 154](#)
- [show system visibility vnf on page 160](#)

List of Sample Output [show system visibility network on page 157](#)

Output Fields [Table 23 on page 156](#) lists the output fields for the `show system visibility network` command. Output fields are listed in the approximate order in which they appear.

Table 23: show system visibility network Output Fields

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

Table 23: show system visibility network Output Fields (continued)

| Field Name | Field Description |
|--|--|
| 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

```

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          Interface Type   Source           Model           MAC
  VLAN-ID
-----
centos2      centos2_vnet6    network default  virtio          00:00:5e:00:53:a2
--
centos2      centos2_vnet7    bridge eth0br    virtio          00:00:5e:00:53:a3
--
centos2      centos2_eth2     bridge ovs-sys-br virtio          00:00:5e:00:53:a4
199
centos2      centos2_eth3     bridge custom1   virtio          00:00:5e:00:53:a5
--
centos1      centos1_vnet4    network default  virtio          00:00:5e:00:53:9e
--
centos1      centos1_vnet5    bridge eth0br    virtio          00:00:5e:00:53:9f
--
centos1      centos1_eth2     bridge ovs-sys-br virtio          00:00:5e:00:53:a0
100
centos1      centos1_eth3     bridge custom1   virtio          00:00:5e:00:53:a1
--

```

OVS Interfaces

```

-----
NAME          MTU
-----
custom1       1500
centos2_eth3  1500
centos1_eth3  1500
veth11        9200
13_h_ge_1_0_0 9200
veth01        9200

```

```
ovs-sys-br      1500
centos1_eth2    1500
centos2_eth2    1500
```

show system visibility vnf

Syntax `show system visibility vnf vnf name`

Release Information Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Description If a VNF name is not specified, this command displays the details of all VNFs present in the system. Details include VNF memory usage, CPU statistics, the list of network interfaces, the list of disk files, per disk usage, per port I/O statistics, and media information, which includes details about CD-ROM and USB storage devices.

If a VNF name is specified, this command displays the details of that particular VNF. Details include VNF memory usage, CPU statistics, the list of network interfaces, the list of disk files, per disk usage, per port I/O statistics, and media information, which includes details about CD-ROM and USB storage devices.

Required Privilege Level view

- Related Documentation**
- [show system visibility cpu on page 146](#)
 - [show system visibility host on page 148](#)
 - [show system visibility memory on page 154](#)
 - [show system visibility network on page 156](#)

List of Sample Output [show system visibility vnf on page 163](#)

Output Fields [Table 24 on page 160](#) lists the output fields for the **show system visibility vnf** command. Output fields are listed in the approximate order in which they appear.

Table 24: 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 24: 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 24: 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 | | |
| vnf-name1 | vnet6 | 0 | 0 | 0 | 8893085 |
| 0 | 0 | 0 | 0 | | |

show vmhost connections

| | |
|---------------------------------|---|
| Syntax | show vmhost connections |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Display the details for the cross-connect connections. The NFX150 supports VLAN PUSH, POP, and SWAP operations. |
| Options | <p>name—Display the details of a specific connection.</p> <p>down—Display the details of connections that are not operational.</p> <p>up—Display the details of connections that are operational.</p> <p>up-down—Display the details of both operational and non-operational connections.</p> |
| Required Privilege Level | view |
| Related Documentation | <ul style="list-style-type: none"> • vmhost on page 136 |
| List of Sample Output | show vmhost connections on page 165 |
| Output Fields | Table 25 on page 165 lists the output fields for the show vmhost connections command. Output fields are listed in the approximate order in which they appear. |

Table 25: 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.

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

Required Privilege Level view

Related Documentation

- [vmhost on page 136](#)

List of Sample Output [show vmhost control-plane on page 167](#)

Sample Output

show vmhost control-plane

```
user@host> show vmhost control-plane
```

```
Vmhost Control Plane Information
```

```
-----
Name | State | Status
-----
Junos Control Plane | RUNNING | OK
Juniper Device Manager | RUNNING | OK
Layer 2 Infrastructure | RUNNING | OK
Layer 3 Infrastructure | RUNNING | OK
LTE | RUNNING | OK
```

show vmhost crash

| | |
|---------------------------------|--|
| Syntax | show vmhost crash |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Display host OS crash information. |
| Required Privilege Level | view |
| Related Documentation | <ul style="list-style-type: none">• vmhost on page 136 |
| List of Sample Output | show vmhost crash on page 168 |

Sample Output

show vmhost crash

```
user@host> show vmhost crash

-rw-r--r-- 1 root root 306773 Mar 22 10:41
local-node.srxpfe.7439.1521715280.core.tgz
-rw-r--r-- 1 root root 307058 Mar 22 10:42
local-node.srxpfe.8184.1521715324.core.tgz
-rw-r--r-- 1 root root 306999 Mar 22 10:42
local-node.srxpfe.8918.1521715357.core.tgz
-rw-r--r-- 1 root root 315121 Apr 18 05:35
localhost.dummy_flowdapp.3037.1524029709.core.tgz
-rw-r--r-- 1 root root 315033 Apr 18 05:17
localhost.dummy_flowdapp.3432.1524028674.core.tgz
-rw-r--r-- 1 root root 315088 Apr 13 18:11
localhost.dummy_flowdapp.3435.1523643106.core.tgz
```


show vmhost forwarding-options analyzer

| | |
|---------------------------------|--|
| Syntax | <code>show vmhost forwarding-options analyzer <i>analyzer-name</i></code> |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Displays information about the VNF analyzers that are configured for port mirroring on a Junos OS platform. |
| Options | <i>analyzer-name</i> —Displays the details of a specific analyzer on the device. |
| Required Privilege Level | view |
| Related Documentation | <ul style="list-style-type: none"> • vmhost on page 136 |
| List of Sample Output | show vmhost forwarding-options analyzer on page 169 |
| Output Fields | Table 26 on page 169 lists the output fields for the <code>show vmhost forwarding-options analyzer</code> command. Output fields are listed in the approximate order in which they appear. |

Table 26: 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.

Description Display the memory information for the host OS.

Required Privilege Level view

Related Documentation

- [vmhost on page 136](#)

List of Sample Output [show vmhost memory on page 170](#)

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 status

Syntax show vmhost status

Release Information Command introduced in Junos OS Release 18.1R1 for NFX150 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 171](#)

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.

Description Display the vmhost storage information.

Required Privilege Level view

Related Documentation

- [vmhost on page 136](#)

List of Sample Output [show vmhost storage on page 172](#)

Sample Output

show vmhost storage

```
user@host> show vmhost storage
```

```
Vmhost Storage Information
```

```
-----
```

```
Storage Name :sda
```

| ID | Storage S.M.A.R.T attribute | Raw value |
|-----|-----------------------------------|------------|
| 1 | Raw_Read_Error_Rate | 0 |
| 5 | Reallocated_Sector_Ct | 0 |
| 9 | Power_On_Hours | 6562 |
| 12 | Power_Cycle_Count | 72 |
| 160 | Uncorrectable_Sector_Count | 0 |
| 161 | Spare_Blocks | 555 |
| 163 | Number_of_Initial_Invalid_Blocks | 31 |
| 164 | Total_Erase_Count | 72780 |
| 165 | Maximum_Erase_Count | 56 |
| 166 | Minimum_Erase_Count | 0 |
| 167 | Average_Erase_Count | 35 |
| 168 | Maximum_Specified_Erase_Count | 3000 |
| 169 | Power-On_UECC_Count | 56 |
| 192 | Power-Off_Retract_Count | 555 |
| 193 | Dynamic_Remaps | 0 |
| 194 | Temperature_Celsius | 37 |
| 195 | Hardware_ECC_Recovered | 646747 |
| 196 | Reallocated_Event_Count | 0 |
| 198 | Offline_Uncorrectable | 0 |
| 199 | UDMA_CRC_Error_Count | 0 |
| 215 | TRIM_Count | 80433 |
| 235 | Total_Flash_LBAs_Written | 103297788 |
| 237 | Total_Flash_LBAs_Written_Expanded | 0 |
| 241 | Total_LBAs_Written | 4262373185 |
| 242 | Total_LBAs_Read | 2322062690 |

| | | |
|-----|----------------------------------|-----|
| 243 | Total_Host_LBAs_Written_Expanded | 0 |
| 244 | Total_Host_LBAs_Read_Expanded | 0 |
| 248 | SSD_Remaining_Life | 99 |
| 249 | Spare_Blocks_Remaining_Life | 100 |

show vmhost uptime

Syntax `show vmhost uptime`

Release Information Command introduced in Junos OS Release 18.1R1 for NFX150 devices.

Description Display the current time and information such as how long the host OS has been running, number of users, and average load.

Required Privilege Level view

Related Documentation

- [vmhost on page 136](#)

List of Sample Output [show vmhost uptime on page 174](#)

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

| | |
|---------------------------------|---|
| Syntax | show vmhost version |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 devices. |
| Description | Display host version information including Linux host kernel version and host software version. |
| Required Privilege Level | view |
| Related Documentation | <ul style="list-style-type: none">• vmhost on page 136 |
| List of Sample Output | show vmhost connections on page 175 |

Sample Output

show vmhost connections

```
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
```

show vmhost vlans

| | |
|---------------------------------|--|
| Syntax | show vmhost vlans |
| Release Information | Command introduced in Junos OS Release 18.1R1 for NFX150 Network Services Platform. |
| Description | Display details about the vmhost VLANs. |
| Options | <p>vlan-name—Display information for a specified VLAN.</p> <p>brief detail extensive —Display the specified level of output.</p> <p>instance—Display information for a specified instance.</p> <p>interface—Name of interface for which the table is displayed.</p> <p>logical-system—Name of logical system.</p> |
| Required Privilege Level | view |
| Related Documentation | <ul style="list-style-type: none"> • vmhost on page 136 |
| List of Sample Output | show vmhost vlans on page 177 |
| Output Fields | Table 27 on page 176 describes the output fields for the show vmhost forwarding-options analyzer show vmhost vlans command. Output fields are listed in the approximate order in which they appear. |

Table 27: 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 |
| ---- | | | |

