



Junos OS

Junos Node Slicing Feature Guide



Modified: 2019-03-19

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.

Junos OS Junos Node Slicing Feature Guide
Copyright © 2019 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

	About the Documentation	xi
	Documentation and Release Notes	xi
	Using the Examples in This Manual	xi
	Merging a Full Example	xii
	Merging a Snippet	xii
	Documentation Conventions	xiii
	Documentation Feedback	xv
	Requesting Technical Support	xv
	Self-Help Online Tools and Resources	xvi
	Creating a Service Request with JTAC	xvi
Chapter 1	Junos Node Slicing Overview	17
	Understanding Junos Node Slicing	17
	Junos Node Slicing Overview	17
	Benefits of Junos Node Slicing	18
	Components of Junos Node Slicing	18
	Base System (BSYS)	20
	Guest Network Function (GNF)	20
	Juniper Device Manager (JDM)	21
	Abstracted Fabric Interface	21
	Understanding Abstracted Fabric Interface Bandwidth	22
	Features Supported on Abstracted Fabric Interfaces	22
	Abstracted Fabric Interface Restrictions	24
	Mastership Behavior of BSYS and GNF	25
	BSYS Mastership	25
	GNF Mastership	25
	Junos Node Slicing Administrator Roles	26
	Multiversion Software Interoperability Overview	26
	Licensing for Junos Node Slicing	26
Chapter 2	Setting Up Junos Node Slicing	29
	Minimum Hardware and Software Requirements for Junos Node Slicing	29
	MX Series Router	29
	x86 Servers (External Server Model)	30
	Server Hardware Resource Requirements (Per GNF)	30
	Shared Server Hardware Resource Requirements (External Server Model)	31
	Server Software Requirements (External Server Model)	32
	Preparing for Junos Node Slicing Setup	33
	Connecting the Servers and the Router	34
	x86 Server CPU BIOS Settings	35

x86 Server Linux GRUB Configuration	36
Updating Intel X710 NIC Driver for x86 Servers	38
Installing Additional Packages for JDM	39
Completing the Connection Between the Servers and the Router	40
Setting Up Junos Node Slicing	41
Configuring an MX Series Router to Operate in BSYS Mode (External Server Model)	41
Installing JDM RPM Package on x86 Servers Running RHEL (External Server Model)	42
Installing JDM Ubuntu Package on x86 Servers Running Ubuntu 16.04 (External Server Model)	43
Configuring JDM on the x86 Servers (External Server Model)	45
Configuring Non-Root Users in JDM (Junos Node Slicing)	47
Configuring JDM interfaces (External Server Model)	48
Configuring MX Series Router to Operate in In-Chassis Mode	51
Installing and Configuring JDM for In-Chassis Model	52
Installing JDM RPM Package on MX Series Router (In-Chassis Model)	52
Configuring JDM (In-Chassis Model)	54
Configuring Guest Network Functions	56
Chassis Configuration Hierarchy at BSYS and GNF	59
Configuring Abstracted Fabric Interfaces Between a Pair of GNFs	59
Class of Service on Abstracted Fabric Interfaces	60
SNMP Trap Support: Configuring NMS Server (External Server Model)	61
Sample Configuration for Junos Node Slicing	63
Sample JDM Configuration (External Server Model)	63
Sample JDM Configuration (In-Chassis Model)	65
Sample BSYS Configuration with Abstracted Fabric Interface	66
Sample Abstracted Fabric Configuration at GNF with Class of Service	67
Sample Output for Abstracted Fabric Interface State at a GNF	68
Setting Up YANG-Based Orchestration of GNFs	69
Installing YANG Package on BSYS	70
Sharing the SSH Identities Between BSYS and JDM	70
Configuring the BSYS to Communicate with JDM	71
Supported XML RPCs: YANG-Based Abstraction of GNFs	71
Disabling the YANG-based Orchestration of GNFs	74
Chapter 3 Upgrading and Managing Junos Node Slicing	75
Junos Node Slicing Upgrade	75
Upgrading Junos Node Slicing	76
Upgrading JDM for External Server Model	77
Upgrading JDM for In-Chassis Model	77

	Upgrading GNF and BSYS	78
	Downgrading JDM for External Server Model	78
	Downgrading JDM for In-Chassis Model	81
	Unified ISSU Support	82
	Managing Multiversion Software Interoperability	82
	Viewing Software Incompatibility Alarms	84
	Viewing Incompatibilities Between Software Versions	84
	Restarting External Servers	85
	Updating Host OS on the External Servers	86
	Applying Security Updates to Host OS	86
	Steps to Apply Host OS Security Updates	87
	Applying Security Patches for Ubuntu Container	88
	Managing Junos Node Slicing	89
	Deleting Guest Network Functions	89
	Disabling Junos Node Slicing	90
Chapter 4	Configuration Statements for BSYS	93
	af-name	94
	control-plane-bandwidth-percent (Node Slicing)	95
	description (GNF)	96
	description (Abstracted Fabric)	97
	fpcs (Node Slicing)	98
	network-slices	99
	gnf	100
	guest-network-functions	101
	peer-gnf	102
	vmhost resize vjunos compact	102
Chapter 5	Operational Commands for BSYS	103
	request vmhost jdm add (In-Chassis Model)	104
	request vmhost jdm delete (In-Chassis Model)	105
	request vmhost jdm start (In-Chassis Model)	106
	request vmhost jdm login (In-Chassis Model)	107
	request vmhost jdm stop (In-Chassis Model)	108
	show vmhost status (In-Chassis Model)	109
	show vmhost jdm status (In-Chassis Model)	111
	show chassis network-slices	112
	show chassis fpc pic-status (Node Slicing)	115
	show chassis fpc (Node Slicing)	117
	show chassis adc (Node Slicing)	119
	show chassis network-slices fpcs	121
	show system anomalies gnf-id (BSYS)	122
Chapter 6	Operational Commands for GNF	125
	show chassis gnf	126
	show chassis gnf	128
	show chassis hardware (GNF)	130
	show chassis fpc (GNF)	133
	show chassis fpc pic-status (GNF)	135
	show chassis adc (GNF)	136

	show interfaces (Abstracted Fabric)	138
	show system anomalies (GNF)	153
Chapter 7	Configuration Statements for JDM	155
	virtual-network-functions	156
	server	157
	interfaces (Junos Node Slicing)	158
	routing-options (Junos Node Slicing)	159
	system login (Junos Node Slicing)	160
	root-login (JDM)	161
Chapter 8	Operational Commands for JDM	163
	Generic Guidelines for Using JDM Server Commands	163
	clear log (JDM)	165
	monitor list (JDM)	166
	monitor start (JDM)	167
	monitor stop (JDM)	169
	request server authenticate-peer-server	170
	request virtual-network-functions	171
	show virtual-network-functions	172
	show version vnf	179
	show version (JDM)	181
	show system cpu (JDM)	183
	show system storage (JDM)	185
	show system memory (JDM)	187
	show system network (JDM)	189
	restart (JDM)	193

List of Figures

Chapter 1	Junos Node Slicing Overview	17
	Figure 1: GNFs on External Server	19
	Figure 2: In-chassis Junos Node Slicing	20
	Figure 3: Abstracted Fabric Interface	22
	Figure 4: Mastership Behavior of GNF and BSYS (External Server Model)	25
Chapter 2	Setting Up Junos Node Slicing	29
	Figure 5: MX2020 Router—External x86 Server Connectivity	34

List of Tables

	About the Documentation	xi
	Table 1: Notice Icons	xiii
	Table 2: Text and Syntax Conventions	xiv
Chapter 1	Junos Node Slicing Overview	17
	Table 3: Supported Abstracted Fabric-capable MPCs	23
Chapter 2	Setting Up Junos Node Slicing	29
	Table 4: GNF Resource Template (External Server Model)	31
	Table 5: GNF Resource Templates for In-Chassis Model	31
	Table 6: Shared Server Resources Requirements (External Server Model)	32
	Table 7: Predefined Login Classes	47
	Table 8: Supported XML RPCs to Manage GNFs	72
Chapter 5	Operational Commands for BSYS	103
	Table 9: show chassis network-slices Output Fields	112
	Table 10: show chassis fpc Output Fields	117
	Table 11: show chassis adc Output Fields	119
	Table 12: show chassis network-slices fpcs Output Fields	121
	Table 13: show system anomalies gnf-id Output Fields	122
Chapter 6	Operational Commands for GNF	125
	Table 14: show chassis gnf Output Fields	126
	Table 15: show chassis gnf Output Fields	128
	Table 16: show chassis hardware Output Fields	130
	Table 17: show chassis fpc Output Fields	133
	Table 18: show chassis adc Output Fields	136
	Table 19: show interfaces (Abstracted Fabric) Output Fields	139
	Table 20: show system anomalies gnf-id Output Fields	153
Chapter 8	Operational Commands for JDM	163
	Table 21: monitor start Output Fields	167
	Table 22: show virtual network functions Output Fields	172
	Table 23: show system cpu Output Fields	183
	Table 24: show system storage Output Fields	185
	Table 25: show system memory Output Fields	187
	Table 26: show system network Output Fields	189

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 <i>metric</i> >;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	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.	

GUI Conventions

Table 2: Text and Syntax Conventions (continued)

Convention	Description	Examples
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<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 Juniper Care or Partner Support Services support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

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

Self-Help Online Tools and Resources

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

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

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

Creating a Service Request with JTAC

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

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

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

CHAPTER 1

Junos Node Slicing Overview

- [Understanding Junos Node Slicing on page 17](#)

Understanding Junos Node Slicing

- [Junos Node Slicing Overview on page 17](#)
- [Components of Junos Node Slicing on page 18](#)
- [Abstracted Fabric Interface on page 21](#)
- [Mastership Behavior of BSYS and GNF on page 25](#)
- [Junos Node Slicing Administrator Roles on page 26](#)
- [Multiversion Software Interoperability Overview on page 26](#)
- [Licensing for Junos Node Slicing on page 26](#)

Junos Node Slicing Overview

Junos Node Slicing enables service providers and large enterprises to create a network infrastructure that consolidates multiple routing functions into a single physical device. It helps leverage the benefits of virtualization without compromising on performance. In particular, Junos Node Slicing enables the convergence of multiple services on a single physical infrastructure while avoiding the operational complexity involved. It provides operational, functional, and administrative separation of functions on a single physical infrastructure that enables the network to implement the same virtualization principles the compute industry has been using for years.

Using Junos Node Slicing, you can create multiple partitions in a single physical MX Series router. These partitions are referred to as guest network functions (GNFs). Each GNF behaves as an independent router, with its own dedicated control plane, data plane, and management plane. This enables you to run multiple services on a single converged MX Series router, while still maintaining operational isolation between them. You can leverage the same physical device to create parallel partitions that do not share the control plane or the forwarding plane, but only share the same chassis, space, and power.

You can also send traffic between GNFs through the switch fabric by using an Abstracted Fabric (**af**) interface, a pseudo interface that behaves as a first class Ethernet interface. An Abstracted Fabric interface facilitates routing control, data, and management traffic between GNFs.

Junos Node Slicing offers two models - an external server model and an in-chassis model. In the external server model, the GNFs are hosted on a pair of industry-standard x86 servers. For the in-chassis model, the GNFs are hosted on the Routing Engines of the MX Series router itself.

Junos Node Slicing supports multiversion software compatibility, thereby allowing the GNFs to be independently upgraded.

Benefits of Junos Node Slicing

- **Converged network**—With Junos Node Slicing, service providers can consolidate multiple network services, such as video edge and voice edge, into a single physical router, while still maintaining operational separation between them. You can achieve both horizontal and vertical convergence. Horizontal convergence consolidates router functions of the same layer to a single router, while vertical convergence collapses router functions of different layers into a single router.
- **Improved scalability**—Focusing on virtual routing partitions, instead of physical devices, improves the programmability and scalability of the network, enabling service providers and enterprises to respond to infrastructure requirements without having to buy additional hardware.
- **Easy risk management**—Though multiple network functions converge on a single chassis, all the functions run independently, benefiting from operational, functional, and administrative separation. Partitioning a physical system, such as Broadband Network Gateway (BNG), into multiple independent logical instances ensures that failures are isolated. The partitions do not share the control plane or the forwarding plane, but only share the same chassis, space, and power. This means failure in one partition does not cause any widespread service outage.
- **Reduced network costs**—Junos Node Slicing enables interconnection of GNFs through internal switching fabrics, which leverages Abstracted Fabric (**af**) interface, a pseudo interface that represents a first class Ethernet interface behavior. With **af** interface in place, companies no longer need to depend on physical interfaces to connect GNFs, resulting in significant savings.
- **Reduced time-to-market for new services and capabilities**—Each GNF can operate on a different Junos software version. This advantage enables companies to evolve each GNF at its own pace. If a new service or a feature needs to be deployed on a certain GNF, and it requires a new software release, only the GNF involved requires an update. Additionally, with the increased agility, Junos Node Slicing enables service providers and enterprises to introduce highly flexible Everything-as-a-service business model to rapidly respond to ever-changing market conditions.

Components of Junos Node Slicing

Junos Node Slicing allows a single MX Series router to be partitioned to appear as multiple, independent routers. Each partition has its own Junos OS control plane, which runs as a virtual machine (VM), and a dedicated set of line cards. Each partition is called a guest network function (GNF).

The MX Series router functions as the base system (BSYS). The BSYS owns all the physical components of the router, including the line cards and the switching fabric. The BSYS assigns line cards to GNFs.

The Juniper Device Manager (JDM) software orchestrates the GNF VMs. In JDM, a GNF VM is referred to as a virtual network function (VNF). A GNF thus comprises a VNF and a set of line cards.

Through configuration at the BSYS, you can assign line cards of the chassis to different GNFs.

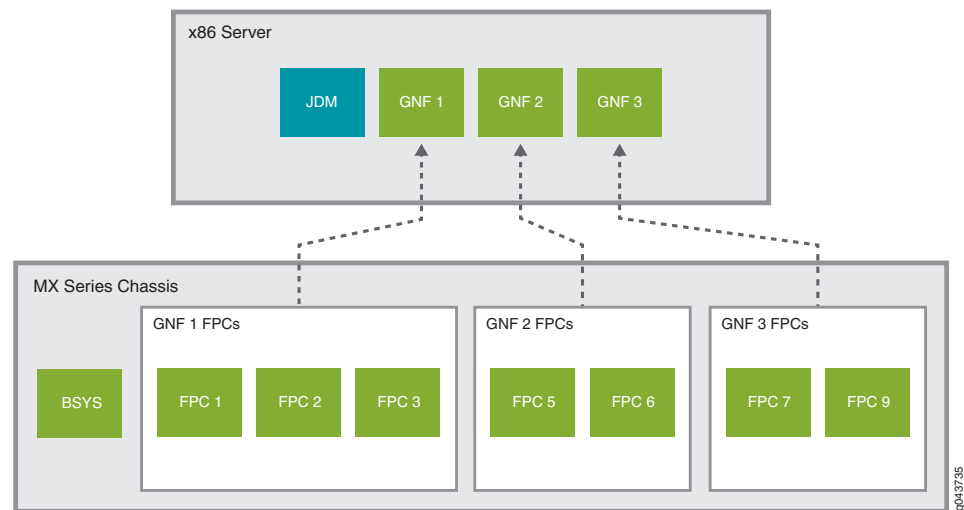
Junos Node Slicing supports two models:

- External server model
- In-chassis model

In the external server model, JDM and VNFs are hosted on a pair of external industry standard x86 servers.

[Figure 1 on page 19](#) shows three GNFs with their dedicated line cards running on an external server.

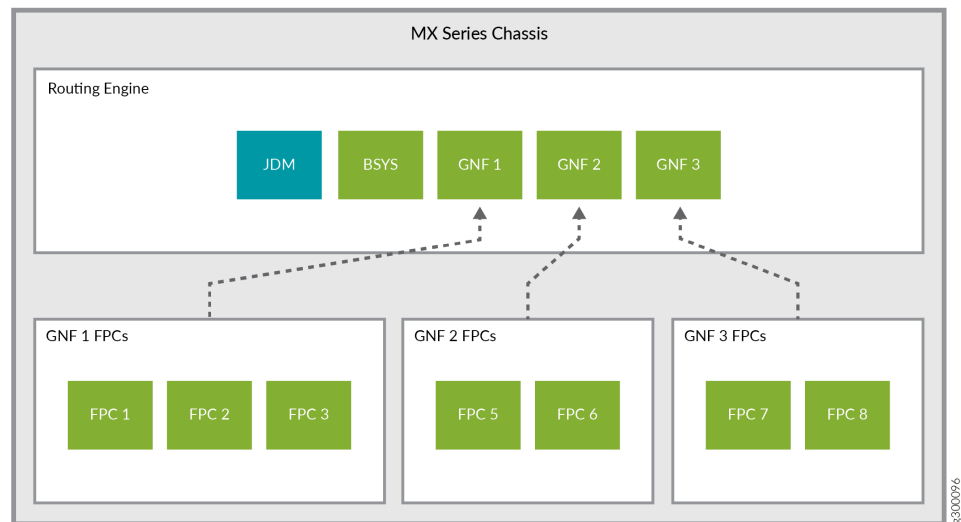
Figure 1: GNFs on External Server



See [“Connecting the Servers and the Router” on page 34](#) for information about how to connect an MX Series router to a pair of external x86 servers.

In the in-chassis model, all components (JDM, BSYS, as well as GNFs) run within the Routing Engine of the MX Series router. See [Figure 2 on page 20](#).

Figure 2: In-chassis Junos Node Slicing



- [Base System \(BSYS\) on page 20](#)
- [Guest Network Function \(GNF\) on page 20](#)
- [Juniper Device Manager \(JDM\) on page 21](#)

Base System (BSYS)

In Junos Node Slicing, the MX Series router functions as the base system (BSYS). The BSYS owns all the physical components of the router, including all line cards and fabric. Through Junos OS configuration at the BSYS, you can assign line cards to GNFs and define Abstracted Fabric (af) interfaces between GNFs. The BSYS software runs on a pair of redundant Routing Engines of the MX Series router.

Guest Network Function (GNF)

A guest network function (GNF) logically owns the line cards assigned to it by the base system (BSYS), and maintains the forwarding state of the line cards. You can configure multiple GNFs on an MX Series router (see [“Configuring Guest Network Functions” on page 56](#)). The Junos OS control plane of each GNF runs as a virtual machine (VM). The Juniper Device Manager (JDM) software orchestrates the GNF VMs. In the JDM context, the GNFs are referred to as virtual network functions (VNF).

A GNF is equivalent to a standalone router. GNFs are configured and administered independently, and are operationally isolated from each other.

Creating a GNF requires two sets of configurations, one to be performed at the BSYS, and the other at the JDM.

A GNF is defined by an ID. This ID must be the same at the BSYS and JDM.

The BSYS part of the GNF configuration comprises giving it an ID and a set of line cards.

The JDM part of the GNF configuration comprises specifying the following attributes:

- A VNF name.
- A GNF ID. This ID must be the same as the GNF ID used at the BSYS.
- The MX Series platform type (for the external server model).
- A Junos OS image to be used for the VNF.
- The VNF server resource template.

The server resource template defines the number of dedicated (physical) CPU cores and the size of DRAM to be assigned to a GNF. For a list of predefined server resource templates available for GNFs, see the *Server Hardware Resource Requirements (Per GNF)* section in “[Minimum Hardware and Software Requirements for Junos Node Slicing](#)” on [page 29](#).

After a GNF is configured, you can access it by connecting to the virtual console port of the GNF. Using the Junos OS CLI at the GNF, you can then configure the GNF system properties such as hostname and management IP address, and subsequently access it through its management port.

Juniper Device Manager (JDM)

The Juniper Device Manager (JDM), a virtualized Linux container, enables provisioning and management of the GNF VMs.

JDM supports Junos OS-like CLI, NETCONF for configuration and management and SNMP for monitoring.



NOTE: In the in-chassis model, JDM does not support SNMP.

A JDM instance is hosted on each of the x86 servers in the external server model, and on each Routing Engine for the in-chassis model. The JDM instances are typically configured as peers that synchronize the GNF configurations: when a GNF VM is created on one server, its backup VM is automatically created on the other server or Routing Engine.

An IP address and an administrator account need to be configured on the JDM. After these are configured, you can directly log in to the JDM.

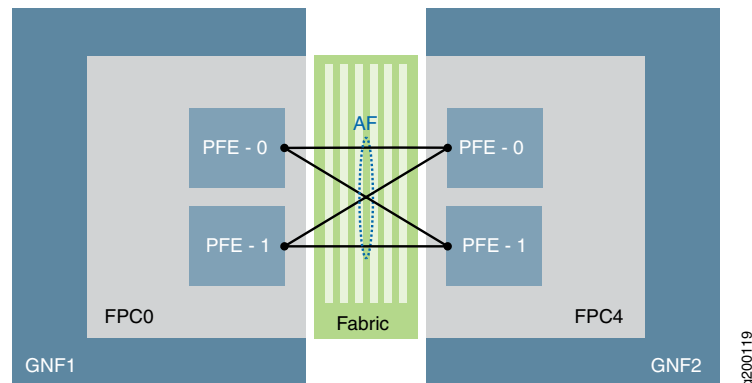
- See Also**
- [Junos Node Slicing Overview on page 17](#)
 - [Mastership Behavior of BSYS and GNF on page 25](#)

Abstracted Fabric Interface

Abstracted Fabric (**af**) interface is a pseudo interface that represents a first class Ethernet interface behavior. An **af** interface facilitates routing control and management traffic between guest network functions (GNFs) through the switch fabric. An **af** interface is created on a GNF to communicate with its peer GNF when the two GNFs are configured to be connected to each other. Abstracted Fabric interfaces must be created at BSYS.

The bandwidth of the **af** interfaces changes dynamically based on the insertion or reachability of the remote line card/MPC. Because the fabric is the communication medium between GNFs, **af** interfaces are considered to be the equivalent WAN interfaces. See [Figure 3 on page 22](#).

Figure 3: Abstracted Fabric Interface



Understanding Abstracted Fabric Interface Bandwidth

An Abstracted Fabric (**af**) interface connects two GNFs through the fabric and aggregates all the Packet Forwarding Engines (PFEs) that connect the two GNFs. An **af** interface can leverage the sum of the bandwidth of each Packet Forwarding Engine belonging to the **af** interface.

For example, if GNF1 has one MPC8 (which has four Packet Forwarding Engines with 240 Gbps capacity each), and GNF1 is connected with GNF2 and GNF3 using **af** interfaces (af1 and af2), the maximum **af** interface capacity on GNF1 would be $4 \times 240 \text{ Gbps} = 960 \text{ Gbps}$.

GNF1—af1—GNF2

GNF1—af2—GNF3

Here, af1 and af2 share the 960 Gbps capacity.

For information on the bandwidth supported on each MPC, see [Table 3 on page 23](#).

Features Supported on Abstracted Fabric Interfaces

Abstracted Fabric interfaces support the following features:

- Load balancing based on the remote GNF line cards present
- Class of service (CoS) support:
 - Inet-precedence classifier and rewrite
 - DSCP classifier and rewrite

- MPLS EXP classifier and rewrite
- DSCP v6 classifier and rewrite for IP v6 traffic
- Support for OSPF, IS-IS, BGP, OSPFv3 protocols, and L3VPN



NOTE: The non-**af** interfaces support all the protocols that work on Junos OS.

- Multicast forwarding
- Graceful Routing Engine switchover (GRES)
- MPLS applications where the **af** interface acts as a core interface (L3VPN, VPLS, L2VPN, L2CKT, EVPN, and IP over MPLS)
- The following protocol families are supported:
 - IPv4 Forwarding
 - IPv6 Forwarding
 - MPLS
 - ISO
 - CCC
- Junos Telemetry Interface (JTI) sensor support
- Starting in Junos OS Release 19.1R1, guest network functions (GNFs) support Ethernet VPNs (EVPN) with Virtual Extensible LAN protocol (VXLAN) encapsulation. This support is available with non-**af** (that is, physical) interface and **af** interface as the core facing interface.
- With the **af** interface configuration, GNFs support **af**-capable MPCs. [Table 3 on page 23](#) lists the **af**-capable MPCs, the number of PFEs supported per MPC, and the bandwidth supported per MPC.

Table 3: Supported Abstracted Fabric-capable MPCs

MPC	Initial Release	Number of PFEs	Total Bandwidth
MPC7E-MRATE	17.4R1	2	480G (240*2)
MPC7E-10G	17.4R1	2	480G (240*2)
MX2K-MPC8E	17.4R1	4	960G (240*4)
MX2K-MPC9E	17.4R1	4	1.6T (400*4)
MPC2E	19.1R1	2	80 (40*2)
MPC2E NG	17.4R1	1	80G

Table 3: Supported Abstracted Fabric-capable MPCs (continued)

MPC	Initial Release	Number of PFEs	Total Bandwidth
MPC2E NG Q	17.4R1	1	80G
MPC3E	19.1R1	1	130G
MPC3E NG	17.4R1	1	130G
MPC3E NG Q	17.4R1	1	130G
32x10GE MPC4E	19.1R1	2	260G (130*2)
2x100GE + 8x10GE MPC4E	19.1R1	2	260G (130*2)
MPC5E-40G10G	18.3R1	2	240G (120*2)
MPC5EQ-40G10G	18.3R1	2	240G (120*2)
MPC5E-40G100G	18.3R1	2	240G (120*2)
MPC5EQ-40G100G	18.3R1	2	240G (120*2)
MX2K-MPC6E	18.3R1	4	520G (130*4)
Multiservices MPC (MS-MPC)	19.1R1	1	120G
16x10GE MPC	19.1R1	4	160G (40*4)



NOTE:

- A GNF that does not have the **af** interface configuration supports all the MPCs that are supported by a standalone MX Series router. For the list of supported MPCs, see [MPCs Supported by MX Series Routers](#).
- We recommend that you set the MTU settings on the **af** interface to align to the maximum allowed value on the XE/GE interfaces. This ensures minimal or no fragmentation of packets over the **af** interface.

Abstracted Fabric Interface Restrictions

The following are the current restrictions of Abstracted Fabric interfaces:

- Configurations such as single endpoint **af** interface, **af** interface-to-GNF mapping mismatch or multiple **af** interfaces mapping to same remote GNF are not checked during commit on the BSYS. Ensure that you have the correct configurations.
- Bandwidth allocation is static, based on the MPC type.

- Abstracted Fabric interfaces do not support the Hyper mode.
- There can be minimal traffic drops (both transit and host) during the offline/restart of an MPC hosted on a remote GNF.
- Interoperability between MPCs that are **af**-capable and the MPCs that are not **af**-capable is not supported.

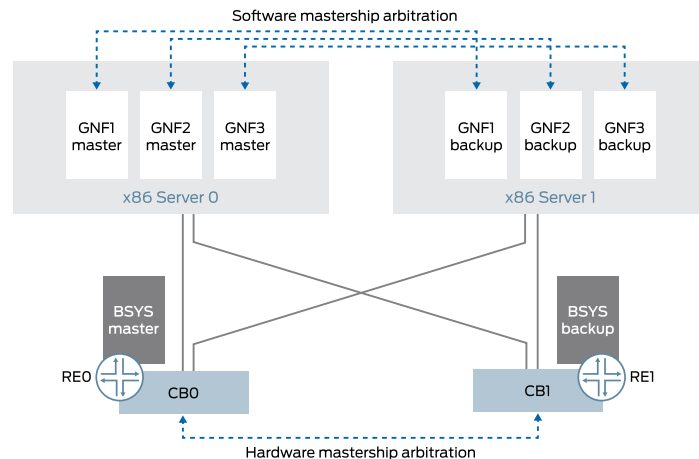
See Also • [Configuring Abstracted Fabric Interfaces Between a Pair of GNFs on page 59](#)

Mastership Behavior of BSYS and GNF

The following sections address the mastership behavior of BSYS and GNF in the context of Routing Engine redundancy.

[Figure 4 on page 25](#) shows the mastership behavior of GNF and BSYS with Routing Engine redundancy.

Figure 4: Mastership Behavior of GNF and BSYS (External Server Model)



BSYS Mastership

The BSYS Routing Engine mastership arbitration behavior is identical to that of Routing Engines on MX Series routers.

GNF Mastership

The GNF VM mastership arbitration behavior is similar to that of MX Series Routing Engines. Each GNF runs as a master-backup pair of VMs. A GNF VM that runs on **server0** (or **re0** for in-chassis) is equivalent to Routing Engine slot 0 of an MX Series router, and the GNF VM that runs on **server1** (or **re1** for in-chassis) is equivalent to Routing Engine slot 1 of an MX Series router.

The GNF mastership is independent of the BSYS mastership and that of other GNFs. The GNF mastership arbitration is done through Junos OS. Under connectivity failure conditions, GNF mastership is handled conservatively.

The GNF mastership model is the same for both external server and in-chassis models.



NOTE: As with the MX Series Routing Engines, you must configure graceful Routing Engine switchover (GRES) at each GNF. This is a prerequisite for the backup GNF VM to automatically take over the mastership when the master GNF VM fails or is rebooted.

Junos Node Slicing Administrator Roles

The following administrator roles enable you to carry out the node slicing tasks:

- **BSYS administrator**—Responsible for the physical chassis, as well as for GNF provisioning (assignment of line cards to GNFs). Junos OS CLI commands are available for these tasks.
- **GNF administrator**—Responsible for configuration, operation, and management of Junos OS at the GNF. All regular Junos OS CLI commands are available to the GNF administrator for these tasks.
- **JDM administrator**—Responsible for the JDM server port configuration (for the external server model), and for the provisioning and life-cycle management of the GNF VMs (VNFs). JDM CLI commands are available for these tasks.

Multiversion Software Interoperability Overview

Starting from Junos OS Release 17.4R1, Junos Node Slicing supports multiversion software compatibility, enabling the BSYS to interoperate with a guest network function (GNF) which runs a Junos OS version that is higher than the software version of the BSYS. This feature supports a range of up to two versions between GNF and BSYS. That is, the GNF software can be two versions higher than the BSYS software. Both BSYS and GNF must meet a minimum version requirement of Junos OS Release 17.4R1.



NOTE: The restrictions in multiversion support are also applicable to the unified ISSU upgrade process.

While JDM software versioning does not have a similar restriction with respect to the GNF or BSYS software versions, we recommend that you regularly update the JDM software. A JDM upgrade does not affect any of the running GNFs.

Licensing for Junos Node Slicing

Operating Junos Node Slicing requires licenses for the GNFs and Abstracted Fabric interfaces to be installed at the BSYS. Running a GNF without a license installed at the BSYS will result in the following syslog message and minor alarm:

```
CHASSISD_LICENSE_EVENT: License Network-Slices: Failed to get valid license('216')
```

```
'gnf-creation'
```

Minor alarm set, 1 Guest network functions creation for JUNOS requires a license.

Please contact Juniper Networks if you have queries pertaining to Junos Node Slicing licenses.

**Related
Documentation**

- [Junos Node Slicing Upgrade on page 75](#)
- [Configuring Abstracted Fabric Interfaces Between a Pair of GNFs on page 59](#)

CHAPTER 2

Setting Up Junos Node Slicing

- [Minimum Hardware and Software Requirements for Junos Node Slicing on page 29](#)
- [Preparing for Junos Node Slicing Setup on page 33](#)
- [Setting Up Junos Node Slicing on page 41](#)
- [Setting Up YANG-Based Orchestration of GNFs on page 69](#)

Minimum Hardware and Software Requirements for Junos Node Slicing

To set up Junos Node Slicing using the external server model, you need an MX Series router and a pair of industry standard x86 servers. The x86 servers host the Juniper Device Manager (JDM) along with the GNF VMs.

To set up Junos Node Slicing using the in-chassis model, you need an MX Series router with MX Series Routing Engines that support x86 virtualization and have sufficient resources to host JDM and GNF VMs.

MX Series Router

The following routers support Junos Node Slicing:

- MX2010
- MX2020
- MX480
- MX960
- MX2008

**NOTE:**

- For the MX960 and MX480 routers, the Control Boards must be SCBE2; and the Routing Engines must be interoperable with SCBE2 (RE-S-1800, RE-S-X6-64G, RE-S-2X00x6-128G, RE-MX2X00x8-128G).
 - To configure in-chassis Junos Node Slicing, the MX Series router must have one of the following types of Routing Engines installed:
 - RE-S-2X00x6-128 (used in MX480 and MX960 routers)
 - RE-MX200X8-128G (used in MX2010 and MX2020 routers)
-

x86 Servers (External Server Model)

Ensure that both the servers have similar (preferably identical) hardware configuration.

The x86 server hardware resource requirements comprise:

- Per-GNF resource requirements (CPU, memory, and storage).
- Shared resource requirements (CPU, memory, storage and network ports).

The server hardware requirements are thus the sum of the requirements of the individual GNFs, and the shared resource requirements. The server hardware requirements are a function of how many GNFs you plan to use.

x86 CPU:

- Must be Intel Haswell-EP or newer.

BIOS:

- Must have hyperthreading disabled.
- Must have performance mode enabled.

Storage:

- Must be local to the server.
- Must be solid-state drive (SSD)-based.

The storage space for GNFs is allocated from the following:

- / (root), which must have a minimum available storage space of 50 GB.
- /vm-primary, which must have a minimum available storage space of 350 GB.

We recommend hardware RAID1 configuration for storage resiliency. We also recommend that you set up /vm-primary as a Linux partition.

Server Hardware Resource Requirements (Per GNF)

Each GNF must be associated with a resource template, which defines the number of dedicated CPU cores and the size of DRAM to be assigned for that GNF.

Table 4 on page 31 lists the GNF resource templates available for configuring Junos Node Slicing on external servers:

Table 4: GNF Resource Template (External Server Model)

Template	CPU cores	DRAM (GB)
2core-16g	2	16
4core-32g	4	32
6core-48g	6	48
8core-64g	8	64



NOTE: Each GNF requires a minimum of 64 GB storage.

Table 5 on page 31 lists the GNF resource templates available for configuring in-chassis Junos Node Slicing:

Table 5: GNF Resource Templates for In-Chassis Model

Template	CPU cores	DRAM (GiB)
1core-16g	1	16
1core-32g	1	32
1core-48g	1	48
2core-16g	2	16
2core-32g	2	32
2core-48g	2	48
4core-32g	4	32
4core-48g	4	48

Shared Server Hardware Resource Requirements (External Server Model)

Table 6 on page 32 lists the server hardware resources that are shared between all the guest network functions (GNFs) on a server:



NOTE: These requirements are in addition to the per-GNF requirements mentioned in the Server Hardware Resource Requirements (Per GNF) section.

Table 6: Shared Server Resources Requirements (External Server Model)

Component	Specification
CPU	<ul style="list-style-type: none"> Four cores to be allocated for JDM and Linux host processing.
Memory	<ul style="list-style-type: none"> Minimum of 32 GB DRAM for JDM and Linux host processing.
Storage	<ul style="list-style-type: none"> Minimum of 64 GB storage for JDM and Linux host.
Network Ports	<ul style="list-style-type: none"> Two 10-Gbps Ethernet interfaces for control plane connection between the server and the router. <ul style="list-style-type: none"> Minimum—1 PCIe NIC card with Intel X710 dual port 10-Gbps Direct Attach, SFP+, Converged Network Adapter, PCIe 3.0, x8 Recommended—2 NIC cards of the above type. Use one port from each card to provide redundancy at the card level. One Ethernet interface (1/10 Gbps) for Linux host management network. One Ethernet interface (1/10 Gbps) for JDM management network. One Ethernet interface (1/10 Gbps) for GNF management network. (This port is shared by all the GNFs on that server). Serial port or an equivalent interface (iDRAC, IPMI) for server console access.

Server Software Requirements (External Server Model)

The x86 servers must have the following installed:

- Red Hat® Enterprise Linux® (RHEL) 7.3 or Ubuntu 16.04 LTS - with virtualization packages.

To enable virtualization for RHEL, choose "Virtualization Host" for the Base Environment and "Virtualization Platform" as an Add-On from the Software Selection screen during installation.



NOTE:

- The hypervisor supported is KVM.

- Install additional packages required for Intel X710 NIC Driver and JDM. For more information, see the ["Updating Intel X710 NIC Driver for x86 Servers" on page 38](#) and ["Installing Additional Packages for JDM" on page 39](#) sections.
- Ensure that you have the latest X710 NIC driver (2.4.10 or later version) and firmware (18.5.17 or later version) installed. For more details, see ["Updating Intel X710 NIC Driver for x86 Servers" on page 38](#).

The servers must also have the BIOS setup as described in ["x86 Server CPU BIOS Settings" on page 35](#) and the Linux GRUB configuration as described in ["x86 Server Linux GRUB Configuration" on page 36](#).

Ensure that the host OS is up to date.

**NOTE:**

- The x86 servers require internet connectivity for you to be able to perform host OS updates and install the additional packages.
- Ensure that you have the same host OS software version on both the servers.

**NOTE:** The following software packages are required to set up Junos Node Slicing:

- JDM package
- Junos OS image for GNFs
- Junos OS package for BSYS
- Junos OS vmhost package for REMX2K-X8-64G and RE-S-X6-64G Control Board-Routing Engine based BSYS

Related Documentation

- [Components of Junos Node Slicing on page 18](#)
- [Connecting the Servers and the Router on page 34](#)

Preparing for Junos Node Slicing Setup



NOTE: Topics in this section apply only to Junos Node Slicing set up using the external server model. For the in-chassis Junos Node Slicing, proceed to “[Configuring MX Series Router to Operate in In-Chassis Mode](#)” on page 51.

Before setting up Junos Node Slicing (external server model), you need to perform a few preparatory steps, such as connecting the servers and the router, installing additional packages, configuring x86 server Linux GRUB, and setting up the BIOS of the x86 server CPUs.

- [Connecting the Servers and the Router on page 34](#)
- [x86 Server CPU BIOS Settings on page 35](#)
- [x86 Server Linux GRUB Configuration on page 36](#)
- [Updating Intel X710 NIC Driver for x86 Servers on page 38](#)
- [Installing Additional Packages for JDM on page 39](#)
- [Completing the Connection Between the Servers and the Router on page 40](#)

Connecting the Servers and the Router

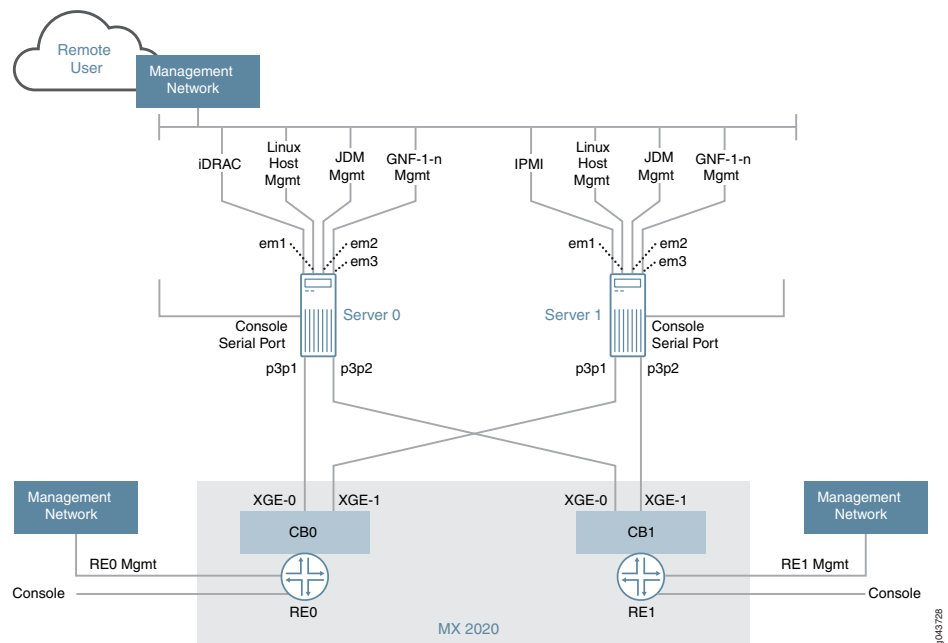
To set up Junos Node Slicing, you must directly connect a pair of external x86 servers to the MX Series router. Besides the management port for the Linux host, each server also requires two additional ports for providing management connectivity for the JDM and the GNF VMs, respectively, and two ports for connecting to the MX Series router.



NOTE: Do not connect the loopback cable to external CB port when Junos Node Slicing is enabled on the MX series router. Also, ensure that the external CB port is not connected to the other CB's external port.

Figure 5 on page 34 shows how an MX2020 router is connected to a pair of x86 external servers.

Figure 5: MX2020 Router—External x86 Server Connectivity



According to the example in Figure 5 on page 34, **em1**, **em2**, and **em3** on the x86 servers are the ports that are used for the management of the Linux host, the JDM and the GNFs, respectively. **p3p1** and **p3p2** on each server are the two 10-Gbps ports that are connected to the Control Boards of the MX Series router.



NOTE: The names of interfaces on the server, such as **em1**, **p3p1** might vary according to the server hardware configuration.

For more information on the XGE ports of the MX Series router Control Board (CB) mentioned in [Figure 5 on page 34](#), see:

- [SCBE2-MX Description](#) (for MX960 and MX480)
- [REMX2K-X8-64G and REMX2K-X8-64G-LT CB-RE Description](#) (for MX2010 and MX2020)



NOTE: Use the `show chassis ethernet-switch` command to view these XGE ports. In the command output on MX960, refer to the port numbers 24 and 26 to view these ports on the SCBE2. In the command output on MX2010 and MX2020, refer to the port numbers 26 and 27 to view these ports on the Control Board-Routing Engine (CB-RE).

x86 Server CPU BIOS Settings

For Junos Node Slicing, the BIOS of the x86 server CPUs should be set up such that:

- Hyperthreading is disabled.
- The CPU cores always run at their rated frequency.
- The CPU cores are set to reduce jitter by limiting C-state use.

To find the rated frequency of the CPU cores on the server, run the Linux host command `lscpu`, and check the value for the field **Model name**. See the following example:

```
Linux server0:~# lscpu
```

```
..
Model name:      Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz
..
```

To find the frequency at which the CPU cores are currently running, run the Linux host command `grep MHz /proc/cpuinfo` and check the value for each CPU core.

On a server that has the BIOS set to operate the CPU cores at their rated frequency, the observed values for the CPU cores will all match the rated frequency (or be very close to it), as shown in the following example.

```
Linux server0:~# grep MHz /proc/cpuinfo
```

```
...
cpu MHz : 2499.902
cpu MHz : 2500.000
cpu MHz : 2500.000
cpu MHz : 2499.902
...
```

On a server that does not have the BIOS set to operate the CPU cores at their rated frequency, the observed values for the CPU cores do not match the rated frequency, and the values could also vary with time (you can check this by rerunning the command).

```
Linux server0:~# grep MHz /proc/cpuinfo
```

```
...
cpu MHz      : 1200.562
cpu MHz      : 1245.468
cpu MHz      : 1217.625
cpu MHz      : 1214.156
```

To set the x86 server BIOS system profile to operate the CPU cores at their rated frequency, reduce jitter, and disable hyperthreading, consult the server manufacturer, because these settings vary with server model and BIOS versions.

Typical BIOS system profile settings to achieve this include:

- **Logical processor:** set to Disabled.
- **CPU power management:** set to Maximum performance.
- **Memory frequency:** set to Maximum performance.
- **Turbo boost:** set to Disabled.
- **C-states and C1E state:** set to Disabled.
- **Energy efficient policy:** set to Performance.
- **Monitor/Mwait:** set to Disabled.

A custom BIOS system profile might be required to set these values.

x86 Server Linux GRUB Configuration

In Junos Node Slicing, each GNF VM is assigned dedicated CPU cores. This assignment is managed by Juniper Device Manager (JDM). On each x86 server, JDM requires that all CPU cores other than CPU cores 0 and 1 be reserved for Junos Node Slicing – and in effect, that these cores be isolated from other applications. CPU cores 2 and 3 are dedicated for GNF virtual disk and network I/O. CPU cores 4 and above are available for assignment to GNF VMs. To reserve these CPU cores, you must set the **isolcpus** parameter in the Linux GRUB configuration as described in the following procedure:

For x86 servers running Red Hat Enterprise Linux (RHEL) 7.3, perform the following steps:

1. Determine the number of CPU cores on the x86 server. Ensure that hyperthreading has already been disabled, as described in [“x86 Server CPU BIOS Settings” on page 35](#). You can use the Linux command **lscpu** to find the total number of CPU cores, as shown in the following example:

```
Linux server0:~# lscpu
```

```
...
Cores per socket: 12
```

```
Sockets: 2
```

```
...
```

Here, there are 24 cores (12 x 2). The CPU cores are numbered as core 0 to core 23.

2. As per this example, the **isolcpus** parameter must be set to 'isolcpus=2-23' (isolate all CPU cores other than cores 0 and 1 for use by JDM and the GNF VMs).

To set the **isolcpus** parameter in the Linux GRUB configuration file, follow the procedure described in the section *Isolating CPUs from the process scheduler* in [this Red Hat document](#). A summary of the section is as follows:

- a. Edit the Linux GRUB file **/etc/default/grub** to append the **isolcpus** parameter to the variable **GRUB_CMDLINE_LINUX**, as shown in the following example:

```
GRUB_CMDLINE_LINUX=
"crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet
isolcpus=2-23"
```

- b. Run the Linux shell command **grub2-mkconfig** to generate the updated GRUB file as shown below:

If you are using legacy BIOS, issue the following command:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

If you are using UEFI, issue the following command:

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

- c. Reboot the x86 server.
- d. Verify that the **isolcpus** parameter has now been set, by checking the output of the Linux command **cat /proc/cmdline**, as shown in the following example:

```
# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-327.36.3.el7.x86_64 ... quiet isolcpus=2-23
```

For x86 servers running Ubuntu 16.04, perform the following steps:

1. Determine the number of CPU cores on the x86 server. Ensure that hyperthreading has already been disabled, as described in x86 Server CPU BIOS Settings. You can use the Linux command **lscpu** to find the total number of CPU cores.
2. Edit the **/etc/default/grub** file to append the **isolcpus** parameter to the variable **GRUB_CMDLINE_LINUX_DEFAULT**, as shown in the following example:

```
GRUB_CMDLINE_LINUX_DEFAULT=
"intel_pstate=disable processor.ignore_ppc=1 isolcpus=2-15"
```

3. To update the changes, run **update-grub**.

4. Reboot the server.
5. Verify that the `isolcpus` parameter has now been set, by checking the output of the Linux command `cat /proc/cmdline`.

Updating Intel X710 NIC Driver for x86 Servers

If you are using Intel X710 NIC, ensure that you have the latest driver (2.4.10 or later) installed on the x86 servers, and that X710 NIC firmware version is 18.5.17 or later.

You need to first identify the X710 NIC interface on the servers. For example, this could be `p3p1`.

You can check the NIC driver version by running the Linux command `ethtool -i interface`. See the following example:

```
root@Linux server0# ethtool -i p3p1

driver: i40e
version: 2.4.10
firmware-version: 5.05 0x80002899 18.5.17
...
```

Refer to the [Intel support page](#) for instructions on updating the driver.



NOTE: Updating the host OS may replace the Intel X710 NIC driver. Therefore, ensure that the host OS is up to date prior to updating the Intel X710 NIC driver.

You need the following packages for building the driver:

- For RedHat:
 - `kernel-devel`
 - `Development Tools`
- For Ubuntu:
 - `make`
 - `gcc`

If you are using RedHat, run the following commands to install the packages:

```
root@Linux server0#yum install kernel-devel
root@Linux server0#yum group install "Development Tools"
```

If you are using Ubuntu, run the following commands to install the packages:

```
root@Linux server0# apt-get install make
```

```
root@Linux server0# apt-get install gcc
```



NOTE: After updating the Intel X710 NIC driver, you might notice the following message in the host OS log:

"i40e: module verification failed: signature and/or required key missing - tainting kernel"

Ignore this message. It appears because the updated NIC driver module has superseded the base version of the driver that was packaged with the host OS.

See Also • [Minimum Hardware and Software Requirements for Junos Node Slicing on page 29](#)

Installing Additional Packages for JDM

The x86 servers must have Red Hat Enterprise Linux (RHEL) 7.3 or Ubuntu 16.04 LTS installed.



NOTE: The x86 Servers must have the virtualization packages installed.

For RHEL 7.3, install the following additional packages, which can be downloaded from the [Red Hat Customer Portal](#).

- python-psutil-1.2.1-1.el7.x86_64.rpm
- net-snmp-5.7.2-24.el7.x86_64.rpm
- net-snmp-libs-5.7.2-24.el7.x86_64.rpm
- libvirt-snmp-0.0.3-5.el7.x86_64.rpm

Only for Junos OS Releases 17.4R1 and earlier, and for 18.1R1, if you are running RHEL 7.3, also install the following additional package:

- libstdc++-4.8.5-11.el7.i686.rpm



NOTE:

- The package version numbers shown are the minimum versions. Newer versions might be available in the latest RHEL 7.3 patches.
- The libstdc++ package extension .i686 indicates that it is a 32-bit package.
- For RHEL, we recommend that you install the packages using the yum command.

For Ubuntu 16.04, install the following packages:

- python-psutil

Only for Junos OS Releases 17.4R1 and earlier, and for 18.1R1, if you are running Ubuntu, also install the following additional package:

- libstdc++6:i386



NOTE:

- For Ubuntu, you can use the `apt-get` command to install the latest version of these packages. For example, use:
 - the command `apt-get install python-psutil` to install the latest version of the `python-psutil` package.
 - the command `apt-get install libstdc++6:i386` to install the latest version of the `libstdc++6` package (the extension `:i386` indicates that the package being installed is a 32-bit version).
-

Completing the Connection Between the Servers and the Router

Complete the following steps before you start installing the JDM:

- Ensure that the MX Series router is connected to the x86 servers as described in [“Connecting the Servers and the Router” on page 34](#).
- Power on the two x86 servers and both the Routing Engines on the MX Series router.
- Identify the Linux host management port on both the x86 servers. For example, **em1**.
- Identify the ports to be assigned for the JDM and the GNF management ports. For example, **em2** and **em3**.
- Identify the two 10-Gbps ports that are connected to the Control Boards on the MX Series router. For example, **p3p1** and **p3p2**.

- See Also**
- [Components of Junos Node Slicing on page 18](#)
 - [Minimum Hardware and Software Requirements for Junos Node Slicing on page 29](#)
 - [Connecting the Servers and the Router on page 34](#)
 - [Sample Configuration for Junos Node Slicing on page 63](#)

- Related Documentation**
- [Junos Node Slicing Overview on page 17](#)
 - [Components of Junos Node Slicing on page 18](#)
 - [Minimum Hardware and Software Requirements for Junos Node Slicing on page 29](#)

Setting Up Junos Node Slicing

Before proceeding to perform the Junos Node Slicing setup tasks, if you are using the external server model, you must have completed the procedures described in the chapter “Preparing for Junos Node Slicing Setup” on page 33.

- [Configuring an MX Series Router to Operate in BSYS Mode \(External Server Model\)](#) on page 41
- [Installing JDM RPM Package on x86 Servers Running RHEL \(External Server Model\)](#) on page 42
- [Installing JDM Ubuntu Package on x86 Servers Running Ubuntu 16.04 \(External Server Model\)](#) on page 43
- [Configuring JDM on the x86 Servers \(External Server Model\)](#) on page 45
- [Configuring Non-Root Users in JDM \(Junos Node Slicing\)](#) on page 47
- [Configuring JDM interfaces \(External Server Model\)](#) on page 48
- [Configuring MX Series Router to Operate in In-Chassis Mode](#) on page 51
- [Installing and Configuring JDM for In-Chassis Model](#) on page 52
- [Configuring Guest Network Functions](#) on page 56
- [Chassis Configuration Hierarchy at BSYS and GNF](#) on page 59
- [Configuring Abstracted Fabric Interfaces Between a Pair of GNFs](#) on page 59
- [SNMP Trap Support: Configuring NMS Server \(External Server Model\)](#) on page 61
- [Sample Configuration for Junos Node Slicing](#) on page 63

Configuring an MX Series Router to Operate in BSYS Mode (External Server Model)



NOTE: Ensure that the MX Series router is connected to the x86 servers as described in “Connecting the Servers and the Router” on page 34.

Junos Node Slicing requires the MX Series router to function as the base system (BSYS).

Use the following steps to configure an MX Series router to operate in BSYS mode:

1. Install the Junos OS package for BSYS on both the Routing Engines of the MX Series router.

To download the package:

- a. Go to the [Juniper Support](#) page.
- b. Click **Base System** > *Junos OS version number* > *Junos version number (64-bit High-End)*.
- c. On the **Software Download** page, select the **I Agree** option under End User License Agreement and then click **Proceed**.

- On the MX Series router, run the **show chassis hardware** command and verify that the transceivers on both the Control Boards (CBs) are detected. The following text represents a sample output:

```
root@router> show chassis hardware
```

```
...
CB 0          REV 23   750-040257   CABL4989   Control Board
  Xcvr 0      REV 01   740-031980   ANT00F9   SFP+-10G-SR
  Xcvr 1      REV 01   740-031980   APG0SC3   SFP+-10G-SR
CB 1          REV 24   750-040257   CABX8889   Control Board
  Xcvr 0      REV 01   740-031980   AP41BKS   SFP+-10G-SR
  Xcvr 1      REV 01   740-031980   ALN0PCM   SFP+-10G-SR
```

- On the MX Series router, apply the following configuration statements:

```
root@router# set chassis network-slices guest-network-functions
root@router# set chassis redundancy graceful-switchover
root@router# set chassis network-services enhanced-ip
root@router# set routing-options nonstop-routing
root@router# set system commit synchronize
root@router# commit
```



NOTE: On MX960 routers, you must configure the `network-services mode as enhanced-ip` or `enhanced-ethernet`. On MX2020 routers, the `enhanced-ip` configuration statement is already enabled by default.

The router now operates in BSYS mode.



NOTE: A router in the BSYS mode is not expected to run features other than the ones required to run the basic management functionalities in Junos Node Slicing. For example, the BSYS is not expected to have interface configurations associated with the line cards installed in the system. Instead, guest network functions (GNFs) will have the full-fledged router configurations.

Installing JDM RPM Package on x86 Servers Running RHEL (External Server Model)

Before installing the JDM RPM package for x86 servers, ensure that you have installed the additional packages, as described in [“Installing Additional Packages for JDM” on page 39](#).

Download and install the JDM RPM package for x86 servers running RHEL as follows:

To download the package:

- Go to the [Juniper Support](#) page.
- Click **JDM > Junos OS version number > Juniper Device Manager version number (for Redhat)**.

- c. On the **Software Download** page, select the **I Agree** option under the End User License Agreement and then click **Proceed**.

To install the package on x86 servers running RHEL, perform the following steps on each of the servers:

1. Disable SELINUX and reboot the server. You can disable SELINUX by setting the value for **SELINUX** to **disabled** in the `/etc/selinux/config` file.
2. Install the JDM RPM package (indicated by the `.rpm` extension) by using the following command. An example of the JDM RPM package used is shown below:

```
root@Linux Server0# rpm -ivh jns-jdm-1.0-0-17.4R1.73.x86_64.rpm
```

```
Preparing... ##### [100%]
Detailed log of jdm setup saved in /var/log/jns-jdm-setup.log
Updating / installing...
 1:jns-jdm-1.0-0 ##### [100%]
Setup host for jdm...
Launch libvirt in listening mode
Done Setup host for jdm
Installing /juniper/.tmp-jdm-install/juniper_ubuntu_rootfs.tgz...
Configure /juniper/lxc/jdm/jdm1/rootfs...
Configure /juniper/lxc/jdm/jdm1/rootfs DONE
Created symlink from /etc/systemd/system/multi-user.target.wants/jdm.service
to /usr/lib/systemd/system/jdm.service.
Done Setup jdm
Redirecting to /bin/systemctl restart rsyslog.service
```

Repeat the steps for the second server.

Installing JDM Ubuntu Package on x86 Servers Running Ubuntu 16.04 (External Server Model)

Before installing the JDM Ubuntu package for x86 servers, ensure that you have installed the additional packages. For more details, see [“Installing Additional Packages for JDM” on page 39](#).

Download and install the JDM Ubuntu package for x86 servers running Ubuntu 16.04 as follows:

To download the JDM Ubuntu package:

- a. Go to the [Juniper Support](#) page.
- b. Click **JDM > Junos OS version number > Juniper Device Manager version number (for Debian)**.
- c. On the **Software Download** page, select the **I Agree** option under the End User License Agreement and then click **Proceed**.

To install the JDM package on the x86 servers running Ubuntu 16.04, perform the following steps on each of the servers:

1. Disable **apparmor** and reboot the server.

```
root@Linux Server0# systemctl stop apparmor
```

```
root@Linux Server0# systemctl disable apparmor
```

```
root@Linux Server0# reboot
```

2. Install the JDM Ubuntu package (indicated by the *.deb* extension) by using the following command. An example of the JDM Ubuntu package used is shown below:

```
root@Linux Server0# dpkg -i jns-jdm-1.0-0-17.4R1.13.x86_64.deb
Selecting previously unselected package jns-jdm.
(Reading database ... 71846 files and directories currently installed.)
Preparing to unpack jns-jdm-1.0-0-17.4R1.13.x86_64.deb ...
Unpacking jns-jdm (1.0-0) ...
Setting up jns-jdm (1.0-0) ...
Installing /juniper/.tmp-jdm-install/juniper_ubuntu_latest.tgz...
Configure /juniper/lxc/jdm/jdm1/rootfs...
Configure /juniper/lxc/jdm/jdm1/rootfs DONE
Done Setup jdm
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for systemd (225-1ubuntu9) ...
```

Repeat the steps for the second server.

Configuring JDM on the x86 Servers (External Server Model)

Use the following steps to configure JDM on each of the x86 servers.

1. At each server, start the JDM, and assign identities for the two servers as **server0** and **server1**, respectively, as follows:

On one server, run the following command:

```
root@Linux server0# jdm start server=0
```

Starting JDM

On the other server, run the following command:

```
root@Linux server1# jdm start server=1
```

Starting JDM



NOTE: The identities, once assigned, cannot be modified without uninstalling the JDM and then reinstalling it:

2. Enter the JDM console on each server by running the following command:

```
root@Linux Server0# jdm console
```

```
Connected to domain jdm
Escape character is ^]
* Starting Signal sysvinit that the rootfs is mounted [ OK ]
* Starting Populate /dev filesystem [ OK ]
* Starting Populate /var filesystem [ OK ]
* Stopping Send an event to indicate plymouth is up [ OK ]
* Stopping Populate /var filesystem [ OK ]
* Starting Clean /tmp directory [ OK ]
...
jdm login:
```

3. Log in as the **root** user.
4. Enter the JDM CLI by running the following command:

```
root@jdm% cli
```



NOTE: The JDM CLI is similar to the Junos OS CLI.

5. Set the root password for the JDM.

```
root@jdm# set system root-authentication plain-text-password
```

New Password:



NOTE:

- The JDM root password must be the same on both the servers.
- Starting in Junos OS Release 18.3R1, you can create non-root users in JDM. For more information, see [“Configuring Non-Root Users in JDM \(Junos Node Slicing\)” on page 47](#).
- JDM installation blocks libvirt port access from outside the host.

6. Commit the changes:

```
root@jdm# commit
```

7. Enter **Ctrl-]** to exit from the JDM console.

8. From the Linux host, run the **ssh jdm** command to log in to the JDM shell.

- See Also**
- [Components of Junos Node Slicing on page 18](#)
 - [Minimum Hardware and Software Requirements for Junos Node Slicing on page 29](#)
 - [Connecting the Servers and the Router on page 34](#)
 - [Sample Configuration for Junos Node Slicing on page 63](#)

Configuring Non-Root Users in JDM (Junos Node Slicing)

In the external server model, you can create non-root users on Juniper Device Manager (JDM) for Junos Node Slicing, starting in Junos OS Release 18.3R1. You need a root account to create a non-root user. The non-root users can log in to JDM by using the JDM console or through SSH. Each non-root user is provided a username and assigned a predefined login class.

The non-root users can perform the following functions:

- Interact with JDM.
- Orchestrate and manage Guest Network Functions (GNFs).
- Monitor the state of the JDM, the host server and the GNFs by using JDM CLI commands.



NOTE: The non-root user accounts function only inside JDM, not on the host server.

To create non-root users in JDM:

1. Log in to JDM as a root user.
2. Define a user name and assign the user with a predefined login class.

```
root@jdm# set system login user username class predefined-login-class
```

3. Set the password for the user.

```
root@jdm# set system login user username authentication plain-text-password
```

New Password:

4. Commit the changes.

```
root@jdm# commit
```

[Table 7 on page 47](#) contains the predefined login classes that JDM supports for non-root users:

Table 7: Predefined Login Classes

Login Class	Permissions
super-user	<ul style="list-style-type: none"> • Create, delete, start and stop GNFs. • Start and stop daemons inside the JDM. • Execute all CLIs. • Access the shell.

Table 7: Predefined Login Classes (continued)

Login Class	Permissions
operator	<ul style="list-style-type: none"> Start and stop GNFs. Restart daemons inside the JDM. Execute all basic CLI operational commands (except the ones which modify the GNFs or JDM configuration).
read-only	Similar to operator class, except that the users cannot restart daemons inside JDM.
unauthorized	Ping and traceroute operations.

- See Also**
- [system login \(Junos Node Slicing\) on page 160](#)
 - [Configuring JDM on the x86 Servers \(External Server Model\) on page 45](#)

Configuring JDM interfaces (External Server Model)

In the JDM, you must configure:

- The two 10-Gbps server ports that are connected to the MX Series router.
- The server port to be used as the JDM management port.
- The server port to be used as the GNF management port.

Therefore, you need to identify the following on each server before starting the configuration of the ports:

- The server interfaces (for example, **p3p1** and **p3p2**) that are connected to **CB0** and **CB1** on the MX Series router.
- The server interfaces (for example, **em2** and **em3**) to be used for JDM management and GNF management.

For more information, see the figure [“Connecting the Servers and the Router” on page 34](#).



NOTE:

- You need this information for both **server0** and **server1**.
- These interfaces are visible only on the Linux host.

To configure the x86 server interfaces in JDM, perform the following steps on both the servers:

1. On **server0**, apply the following configuration statements:

```
root@jdm# set groups server0 server interfaces cb0 p3p1
root@jdm# set groups server0 server interfaces cb1 p3p2
root@jdm# set groups server1 server interfaces cb0 p3p1
root@jdm# set groups server1 server interfaces cb1 p3p2
root@jdm# set apply-groups [ server0 server1 ]
root@jdm# commit

root@jdm# set groups server0 server interfaces jdm-management em2
root@jdm# set groups server0 server interfaces vnf-management em3
root@jdm# set groups server1 server interfaces jdm-management em2
root@jdm# set groups server1 server interfaces vnf-management em3
root@jdm# commit
```

2. Repeat the step 1 on **server1**.



NOTE: Ensure that you apply the same configuration on both **server0** and **server1**.

3. Share the **ssh** identities between the two x86 servers.

At both **server0** and **server1**, run the following JDM CLI command:

```
root@jdm> request server authenticate-peer-server
```



NOTE: The `request server authenticate-peer-server` command displays a CLI message requesting you to log in to the peer server using `ssh` to verify the operation. To log in to the peer server, you need to prefix `ip netns exec jdm_nv_ns` to `ssh root@jdm-server1`.

For example, to log in to the peer server from **server0**, exit the JDM CLI, and use the following command from JDM shell:

```
root@jdm:~# ip netns exec jdm_nv_ns ssh root@jdm-server1
```

Similarly, to log in to the peer server from **server1**, use the following command:

```
root@jdm:~# ip netns exec jdm_nv_ns ssh root@jdm-server0
```

4. Apply the configuration statements in the JDM CLI configuration mode to set the JDM management IP address, default route, and the JDM hostname for each JDM instance as shown in the following example.

**NOTE:**

- The management IP address and default route must be specific to your network.
- JDM does not support IPv6, even though IPv6 addresses are themselves configurable.

```

root@jdm# set groups server0 interfaces jmgmt0 unit 0 family inet address 10.216.105.112/21
root@jdm# set groups server1 interfaces jmgmt0 unit 0 family inet address 10.216.105.113/21
root@jdm# set groups server0 routing-options static route 0.0.0.0/0 next-hop 10.216.111.254
root@jdm# set groups server1 routing-options static route 0.0.0.0/0 next-hop 10.216.111.254
root@jdm# set groups server0 system host-name test-jdm-server0
root@jdm# set groups server1 system host-name test-jdm-server1
root@jdm# commit

```

**NOTE:**

- `jmgmt0` stands for the JDM management port. This is different from the Linux host management port. Both JDM and the Linux host management ports are independently accessible from the management network.

5. Run the following JDM CLI command on each server and ensure that all the interfaces are up.

```
root@jdm> show server connections
```

Status	Comments	Component	Interface
up		Host to JDM port	virbr0
up		Physical CB0 port	p3p1
up		Physical CB1 port	p3p2
up		Physical JDM mgmt port	em2
up		Physical VNF mgmt port	em3
up		JDM-GNF bridge	bridge_jdm_vm
up		CB0	cb0
up		CB1	cb1
up		JDM mgmt port	jmgmt0
up		JDM to HOST port	bme1
up		JDM to GNF port	bme2
up		JDM to JDM link0*	cb0.4002
up		JDM to JDM link1	cb1.4002

```
up
```



NOTE: For sample JDM configurations, see “[Sample Configuration for Junos Node Slicing](#)” on page 63.

If you want to modify the server interfaces configured in the JDM, perform the following steps:

1. Stop all running GNFs.

```
root@jdm> request virtual-network-functions gnf-name stop
```

2. From the configuration mode, deactivate the virtual network functions configuration, and then commit the change.

```
root@jdm# deactivate virtual-network-functions
root@jdm# commit
```

3. Configure and commit the new interfaces as described in the step 1 of the main procedure.

4. Reboot the JDM from the shell.

```
root@jdm:~# reboot
```

5. From the configuration mode, activate the virtual network functions configuration, and then commit the change.

```
root@jdm# activate virtual-network-functions
root@jdm# commit
```

Configuring MX Series Router to Operate in In-Chassis Mode



NOTE:

- To configure in-chassis Junos Node Slicing, the MX Series router must have one of the following types of Routing Engines installed:
 - RE-S-2X00x6-128 (used in MX480 and MX960 routers)
 - RE-MX200X8-128G (used in MX2010 and MX2020 routers)

In in-chassis model, the base system (BSYS), Juniper Device Manager (JDM), and all guest network functions (GNFs) run within the Routing Engine of the MX Series router. BSYS and GNFs run on the host as virtual machines (VMs). You need to first reduce the resource footprint of the standalone MX Series router as follows:

1. Ensure that both the Routing Engines (re0 and re1) in the MX Series router have the required VM host package (example: `junos-vmhost-install-mx-x86-64-19.2R1.tgz`) installed. The VM host package should be of 19.1R1 or a later version.
2. Applying the following configuration and then reboot VM host on both the Routing Engines (re0 and re1).

```
user@router# set vmhost resize vjunos compact
user@router# set system commit synchronize
user@router> request vmhost reboot (re0|re1)
```

When this configuration is applied, and following the reboot, the Routing Engine resource footprint of the Junos VM on MX Series router shrinks in order to accommodate GNF VMs. A resized Junos VM, now operating as the base system (BSYS) on the MX Series Routing Engine has the following resources:

- CPU Cores—1 (Physical)
- DRAM—16GB
- Storage—14GB (/var)



NOTE: All files in the /var/ location, including the log files (/var/log) and core files (/var/crash), are deleted when you reboot VM host after configuring the `set vmhost resize vjunos compact` statement. You must save any files currently in /var/log or /var/crash before proceeding with the VM host resize configuration if you want to use them for reference.

Installing and Configuring JDM for In-Chassis Model

Steps listed in this topic apply only to in-chassis Junos Node Slicing configuration.

- [Installing JDM RPM Package on MX Series Router \(In-Chassis Model\) on page 52](#)
- [Configuring JDM \(In-Chassis Model\) on page 54](#)

Installing JDM RPM Package on MX Series Router (In-Chassis Model)

Before installing the Juniper Device Manager (JDM) RPM package on an MX Series router, you must configure the MX Series router to operate in the in-chassis BSYS mode. For more information, see [“Configuring MX Series Router to Operate in In-Chassis Mode” on page 51](#).



NOTE: The RPM package `jns-jdm-vmhost` is meant for in-chassis Junos Node Slicing deployment, while the RPM package `jns-jdm` is used for external servers based Junos Node Slicing deployment.

1. Download the JDM RPM package from the [Juniper Support](#) page.

2. Install the JDM RPM package on both Routing Engines (re0 and re1), by using the command shown in the following example:

```

root@router> request vmhost jdm add jns-jdm-vmhost-18.3-20180930.0.x86_64.rpm

Starting to validate the Package
Finished validating the Package
Starting to validate the Environment
Finished validating the Environment
Starting to copy the RPM package from Admin Junos to vmhost
Finished Copying the RPM package from Admin Junos to vmhost
Starting to install the JDM RPM package
Preparing... #####
Detailed log of jdm setup saved in /var/log/jns-jdm-setup.log
jns-jdm-vmhost #####
Setup host for jdm...
Done Setup host for jdm
Installing
/vm/vm/iapps/jdm/install/juniper/.tmp-jdm-install/juniper_ubuntu_rootfs.tgz...
Configure /vm/vm/iapps/jdm/install/juniper/lxc/jdm/jdm1/rootfs...
Configure /vm/vm/iapps/jdm/install/juniper/lxc/jdm/jdm1/rootfs DONE
Setup Junos cgroups...Done
Done Setup jdm
stopping rsyslogd ... done
starting rsyslogd ... done
Finished installing the JDM RPM package
Installation Successful !
Starting to generate the host public keys at Admin Junos
Finished generating the host public keys at Admin Junos
Starting to copy the host public keys from Admin Junos to vmhost
Finished copying the host public keys from Admin Junos to vmhost
Starting to copy the public keys of Admin junos from vmhost to JDM
Finished copying the public keys of Admin junos from vmhost to JDM
Starting to cleanup the temporary file from Vmhost containing host keys of
Admin Junos
Finished cleaning the temporary file from Vmhost containing host keys of Admin
Junos

```

3. Run the **show vmhost status** command to see the **vJunos Resource Status** on both the Routing Engines.

```

user@router> show vmhost status re0

```

```

bsys-re0:
-----

```

```

Compute cluster: rainier-re-cc
  Compute Node: rainier-re-cn, Online

```

```

vJunos Resource Status: Compact

```

```

user@router> show vmhost status re1

```

```

bsys-re1:
-----

```

```

Compute cluster: rainier-re-cc

```

```
Compute Node: rainier-re-cn, Online
vJunos Resource Status: Compact
```

Configuring JDM (In-Chassis Model)

Use the following steps to configure JDM on both the Routing Engines of an MX Series router:

1. Apply the following command on both the Routing Engines to start JDM:

```
user@router> request vmhost jdm start
```

```
Starting JDM
Starting jdm: Domain jdm defined from
/vm/vm/iapps/jdm//install/juniper/lxc/jdm/current/config/jdm.xml

Domain jdm started
```



NOTE: If hyperthreading is disabled, a warning is displayed when you enter the command `request vmhost jdm start`, as shown in the following example:

```
Warning: Hyperthreading is disabled! Cores: (6) Processors: (6)
Expected: (12)
```

2. Use the command `show vmhost jdm status` to check if the JDM is running.

```
user@router> show vmhost jdm status
```

```
JDM Information
-----
Package      : jns-jdm-vmhost-19.1-B2.x86_64
Status       : Running
PID          : 3088
Free Space   : 62967 (MiB)
```

3. After a few seconds, log in to JDM.

```
root@router> request vmhost jdm login
```

```
*****
* The Juniper Device Manager (JDM) must only be used for orchestrating the *
* Virtual Machines for Junos Node Slicing                               *
*                                                                           *
* Host Linux Distro: Wind River Linux                                   *
* JDM Version: jns-jdm-vmhost-19.1-20181003.dev.common.0.x86_64         *
* Free Disk Space on JDM's root-fs ("/"): 125081(MiB)                   *
*****
Last login: Thu Oct  4 15:26:30 2018 from 192.168.1.1
```

**NOTE:**

- You need to have root user privilege on the BSYS to log in to JDM.
- The in-chassis JDM root account password can be different from Junos root account password.
- It takes approximately 10 seconds for JDM to start. If you enter the request vmhost jdm login command before JDM starts, you might get the following message:

```
ssh_exchange_identification: read: Connection reset by peer
```

4. Enter the JDM CLI by running the following command:

```
root@jdm% cli
```

5. In configuration mode, apply the configurations shown in the following example:



NOTE: The IP addresses shown in the following example are samples. Replace them with the actual IP addresses in your configuration.

```
root@jdm# set groups server0 system host-name host-name
root@jdm# set groups server0 interfaces jmgmt0 unit 0 family inet address 192.0.2.1/24
root@jdm# set groups server0 routing-options static route 0.0.0.0/0 next-hop 192.0.2.2
root@jdm# set groups server1 system host-name host-name
root@jdm# set groups server1 interfaces jmgmt0 unit 0 family inet address 198.51.100.1/24
root@jdm# set groups server1 routing-options static route 0.0.0.0/0 next-hop 198.51.100.2
```

6. In configuration mode, set the root password for the JDM on both the Routing Engines, and commit.

```
root@jdm# set apply-groups [server0 server1]
root@jdm# set system root-authentication plain-text-password
```

```
New password:
```

```
root@jdm# commit
```

**NOTE:**

- The JDM supports root user administration account only.

7. In operation mode, enter the following command on both the Routing Engines to copy the ssh public key to the peer JDM.

```
root@jdm> request server authenticate-peer-server
```

```
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
root@jdm-server1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@jdm-server1'"
and check to make sure that only the key(s) you wanted were added.
```



NOTE: You need to enter the root password of the peer JDM when prompted.

8. In the configuration mode, apply the following commands:

```
root@jdm# set system commit synchronize
root@jdm# set apply-groups [server0 server1]
```



NOTE:

- In in-chassis Junos Node Slicing, you cannot ping or send traffic between the management interfaces of the same Routing Engine (for example, from the Routing Engine 0 of GNF1 to the Routing Engine 0 of GNF2 or from the Routing Engine 0 of GNF1 to JDM).
- In in-chassis mode, you cannot perform an scp operation between the BSYS and the JDM management interfaces.

Configuring Guest Network Functions

Configuring a guest network function (GNF) comprises two tasks, one to be performed at the BSYS and the other at the JDM.



NOTE:

- Before attempting to create a GNF, you must ensure that the servers (or Routing Engines in the case of in-chassis model) have sufficient resources (CPU, memory, storage) for that GNF.
- You need to assign an ID to each GNF. This ID must be the same at the BSYS and the JDM.

At the BSYS, specify a GNF by assigning it an ID and a set of line cards by applying the configuration as shown in the following example:

```
user@router# set chassis network-slices guest-network-functions gnf 1 fpcs 4

user@router# commit
```

In the JDM, the GNF VMs are referred to as virtual network functions (VNFs). A VNF has the following attributes:

- A VNF name.
- A GNF ID. This ID must be the same as the GNF ID used at the BSYS.
- The MX Series platform type.
- A Junos OS image to be used for the GNF.
- The VNF server resource template.

At the JDM, to configure a VNF, perform the following steps:

1. Use the JDM shell command **scp** to retrieve the Junos OS Node Slicing image for GNF and place it in the JDM local directory **/var/jdm-usr/gnf-images** (repeat this step to retrieve the GNF configuration file).

```
root@jdm:~# scp source-location-of-the-gnf-image /var/jdm-usr/gnf-images
root@jdm:~# scp source-location-of-the-gnf-configuration-file /var/jdm-usr/gnf-config
```

2. Assign this image to a GNF by using the JDM CLI command as shown in the following example:

```
root@test-jdm-server0> request virtual-network-functions test-gnf add-image
/var/jdm-usr/gnf-images/junos-install-ns-mx-x86-64-17.4R1.10.tgz all-servers
```

```
Server0:
Added image: /vm-primary/test-gnf/test-gnf.img
```

```
Server1:
Added image: /vm-primary/test-gnf/test-gnf.img
```

3. Configure the VNF by applying the configuration statements as shown in the following example:

```
root@test-jdm-server0# set virtual-network-functions test-gnf id 1

root@test-jdm-server0# set virtual-network-functions test-gnf chassis-type mx2020

root@test-jdm-server0# set virtual-network-functions test-gnf resource-template
2core-16g
```

For in-chassis model, do not configure the platform type (**set virtual-network-functions test-gnf chassis-type mx2020**). It will be detected automatically.

To also specify a baseline or initial Junos OS configuration for a GNF, prepare the GNF configuration file (example: `/var/jdm-usr/gnf-config/test-gnf.conf`) on both the servers (server0 and server1) for external server model, and on both the Routing Engines (re0 and re1) for the in-chassis model, and specify the filename as the parameter in the **base-config** statement as shown below:

```
root@test-jdm-server0# set virtual-network-functions test-gnf base-config
/var/jdm-usr/gnf-config/test-gnf.conf
```

```
root@test-jdm-server0# commit synchronize
```



NOTE: Ensure that:

- You use the same GNF ID as the one specified earlier in BSYS.
- The baseline configuration filename (with the path) is the same on both the servers / Routing Engines.
- The syntax of the baseline file contents is in the Junos OS configuration format.
- The GNF name used here is the same as the one assigned to the Junos OS image for GNF in the step 2.

4. To verify that the VNF is created, run the following JDM CLI command:

```
root@test-jdm-server0> show virtual-network-functions test-gnf
```

5. Log in to the console of the VNF by issuing the following JDM CLI command:

```
root@test-jdm-server0> request virtual-network-functions test-gnf console
```

6. Configure the VNF the same way as you configure an MX Series Routing Engine.



NOTE:

- The CLI prompt for in-chassis model is `root@jdm#`.
- For sample configurations, see [“Sample Configuration for Junos Node Slicing” on page 63](#).
- In the case of the external server model, if you had previously brought down any physical x86 CB interfaces or the GNF management interface from Linux shell (by using the command `ifconfig interface-name down`), these will automatically be brought up when the GNF is started.

Chassis Configuration Hierarchy at BSYS and GNF

In Junos Node Slicing, the BSYS owns all the physical components of the router, including the line cards and fabric, while the GNFs maintain forwarding state on their respective line cards. In keeping with this split responsibility, Junos CLI configuration under the **chassis** hierarchy (if any), should be applied at the BSYS or at the GNF as follows:

- Physical-level parameters under the **chassis** configuration hierarchy should be applied at the BSYS. For example, the configuration for handling physical errors at an FPC is a physical-level parameter, and should therefore be applied at the BSYS.

At BSYS Junos CLI:

[edit]

```
user@router# set chassis fpc fpc slot error major threshold threshold value action alarm
```

- Logical or feature-level parameters under the **chassis** configuration hierarchy should be applied at the GNF associated with the FPC. For example, the configuration for max-queues per line card is a logical-level parameter, and should therefore be applied at the GNF.

At GNF Junos CLI:

[edit]

```
user@router# set chassis fpc fpc slot max-queues value
```

- As exceptions, the following two parameters under the **chassis** configuration hierarchy should be applied at both BSYS and GNF:

At both BSYS and GNF CLI:

[edit]

```
user@router# set chassis network-services network services mode
```

```
user@router# set chassis fpc fpc slot flexible-queueing-mode
```

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs

Creating an Abstracted Fabric (**af**) interface between two guest network functions (GNFs) involves configurations both at the base system (BSYS) and at the GNF. Abstracted Fabric interfaces are created on GNFs based on the BSYS configuration, which is then sent to those GNFs.



NOTE: Only one **af** interface can be configured between a pair of GNFs.

To configure **af** interfaces between a pair of GNFs:

- At the BSYS, apply the configuration as shown in the following example:

```
user@router# set chassis network-slices guest-network-functions gnf 2 af4 peer-gnf id 4
user@router# set chassis network-slices guest-network-functions gnf 2 af4 peer-gnf af2
user@router# set chassis network-slices guest-network-functions gnf 4 af2 peer-gnf id 2
user@router# set chassis network-slices guest-network-functions gnf 4 af2 peer-gnf af4
```

In this example, **af2** is the Abstracted Fabric interface instance 2 and **af4** is the Abstracted Fabric interface instance 4.



NOTE: The allowed **af** interface values range from **af0** through **af9**.

The GNF **af** interface will be visible and up. You can configure an **af** interface the way you configure any other interface.

2. At the GNF, apply the configuration as shown in the following example:

```
user@router-gnf-b# set interfaces af4 unit 0 family inet address 10.10.1/24
user@router-gnf-d# set interfaces af2 unit 0 family inet address 10.10.2/24
```



NOTE:

- If you want to apply MPLS family configurations on the **af** interfaces, you can apply the command `set interfaces af-name unit logical-unit-number family mpls` on both the GNFs between which the **af** interface is configured.
- For sample **af** configurations, see [“Sample Configuration for Junos Node Slicing” on page 63](#).

Class of Service on Abstracted Fabric Interfaces

Class of service (CoS) packet classification assigns an incoming packet to an output queue based on the packet's forwarding class. See [CoS Configuration Guide](#) for more details.

The following sections explain the forwarding class- to-queue mapping, and the behavior aggregate (BA) classifiers and rewrites supported on the Abstracted Fabric (**af**) interfaces.

- [Forwarding Class-to-Queue Mapping on page 60](#)
- [BA Classifiers and Rewrites on page 61](#)

Forwarding Class-to-Queue Mapping

An **af** interface is a simulated WAN interface with most capabilities of any other interface except that the traffic designated to a remote Packet Forwarding Engine will still have to go over the two fabric queues (Low/High priority ones).



NOTE: Presently, an **af** interface operates in 2-queue mode only. Hence, all queue-based features such as scheduling, policing, and shaping are not available on an **af** interface.

Packets on the **af** interface inherit the fabric queue that is determined by the fabric priority configured for the forwarding class to which that packet belongs. For example, see the following forwarding class to queue map configuration:

[edit]

user@router# show class-of-service forwarding-classes

```

class Economy queue-num 0 priority low; /* Low fabric priority */
class Stream queue-num 1;
class Business queue-num 2;
class Voice queue-num 3;
class NetControl queue-num 3;
class Business2 queue-num 4;
class Business3 queue-num 5;
class VoiceSig queue-num 6 priority high; /* High fabric priority */
class VoiceRTP queue-num 7;

```

As shown in the preceding example, when a packet gets classified to the forwarding class **VoiceSig**, the code in the forwarding path examines the fabric priority of that forwarding class and decides which fabric queue to choose for this packet. In this case, high-priority fabric queue is chosen.

BA Classifiers and Rewrites

The behavior aggregate (BA) classifier maps a class-of-service (CoS) value to a forwarding class and loss priority. The forwarding class and loss-priority combination determines the CoS treatment given to the packet in the router. The following BA classifiers and rewrites are supported:

- Inet-Precedence classifier and rewrite
- DSCP classifier and rewrite
- MPLS EXP classifier and rewrite

You can also apply rewrites for IP packets entering the MPLS tunnel and do a rewrite of both EXP and IPv4 type of service (ToS) bits. This approach will work as it does on other normal interfaces.

- DSCP v6 classifier and rewrite for IP v6 traffic



NOTE:

The following are not supported:

- IEEE 802.1 classification and rewrite
- IEEE 802.1AD (QinQ) classification and rewrite

See [CoS Configuration Guide](#) for details on CoS BA classifiers.

SNMP Trap Support: Configuring NMS Server (External Server Model)

The Juniper Device Manager (JDM) supports the following SNMP traps:

- LinkUp and linkDown traps for JDM interfaces.

Standard linkUp/linkDown SNMP traps are generated. A default community string **jdm** is used.

- LinkUp/linkDown traps for host interfaces.

Standard **linkUp/linkDown** SNMP traps are generated. A default community string **host** is used.

- JDM to JDM connectivity loss/regain traps.

JDM to JDM connectivity loss/regain traps are sent using generic syslog traps (**jnxSyslogTrap**) through the host management interface.

The JDM connectivity down trap **JDM_JDM_LINK_DOWN** is sent when the JDM is not able to communicate with the peer JDM on another server over **cb0** or **cb1** links. See the following example:

```
{ SNMPv2c C=host { V2Trap(296) R=1299287309
.1.3.6.1.2.1.1.3.0=42761992
.1.3.6.1.6.3.1.1.4.1.0=.1.3.6.1.4.1.2636.4.12.0.1
.1.3.6.1.4.1.2636.3.35.1.1.1.2.1="JDM_JDM_LINK_DOWN"
.1.3.6.1.4.1.2636.3.35.1.1.1.3.1=""
.1.3.6.1.4.1.2636.3.35.1.1.1.4.1=5
.1.3.6.1.4.1.2636.3.35.1.1.1.5.1=24
.1.3.6.1.4.1.2636.3.35.1.1.1.6.1=0
.1.3.6.1.4.1.2636.3.35.1.1.1.7.1="jdmmon"
.1.3.6.1.4.1.2636.3.35.1.1.1.8.1="JDM-HOST"
.1.3.6.1.4.1.2636.3.35.1.1.1.9.1="JDM to JDM Connection Lost"
.1.3.6.1.6.3.1.1.4.3.0.0="" } }
```

The JDM to JDM Connectivity up trap **JDM_JDM_LINK_UP** is sent when either the **cb0** or **cb1** link comes up, and JDMs on both the servers are able to communicate again. See the following example:

```
{ SNMPv2c C=host { V2Trap(292) R=998879760
.1.3.6.1.2.1.1.3.0=42762230
.1.3.6.1.6.3.1.1.4.1.0=.1.3.6.1.4.1.2636.4.12.0.1
.1.3.6.1.4.1.2636.3.35.1.1.1.2.1="JDM_JDM_LINK_UP"
.1.3.6.1.4.1.2636.3.35.1.1.1.3.1=""
.1.3.6.1.4.1.2636.3.35.1.1.1.4.1=5
.1.3.6.1.4.1.2636.3.35.1.1.1.5.1=24
.1.3.6.1.4.1.2636.3.35.1.1.1.6.1=0
.1.3.6.1.4.1.2636.3.35.1.1.1.7.1="jdmmon"
.1.3.6.1.4.1.2636.3.35.1.1.1.8.1="JDM-HOST"
.1.3.6.1.4.1.2636.3.35.1.1.1.9.1="JDM to JDM Connection Up"
.1.3.6.1.6.3.1.1.4.3.0.0="" } }
```

- VM(GNF) up/down—**libvirtGuestNotif** notifications.

For GNF start/shutdown events, the standard **libvirtGuestNotif** notifications are generated. For **libvirtMIB** notification details, see this [web page](#). Also, see the following example:

```

HOST [UDP: [127.0.0.1]:53568->[127.0.0.1]]: Trap ,
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (636682) 1:46:06.82,
SNMPv2-MIB::snmpTrapOID.0 = OID: LIBVIRT-MIB::libvirtGuestNotif,
LIBVIRT-MIB::libvirtGuestName.0 = STRING: "gnf1",
LIBVIRT-MIB::libvirtGuestUUID.1 = STRING: 7ad4bc2a-16db-d8c0-1f5a-6cb777e17cd8,
LIBVIRT-MIB::libvirtGuestState.2 = INTEGER: running(1),
LIBVIRT-MIB::libvirtGuestRowStatus.3 = INTEGER: active(1)

```

SNMP traps are sent to the target NMS server. To configure the target NMS server details in the JDM, see the following example:

[edit]

```

root@jdm# show snmp | display set
root@jdm# set snmp name name
root@jdm# set snmp description description
root@jdm# set snmp location location
root@jdm# set snmp contact user's email
root@jdm# set snmp trap-group tg-1 targets target ip address1
root@jdm# set snmp trap-group tg-1 targets target ip address2

```

Sample Configuration for Junos Node Slicing

This section provides sample configurations for Junos Node Slicing.

Sample JDM Configuration (External Server Model)

```

root@test-jdm-server0> show configuration
groups {
  server0 {
    system {
      host-name test-jdm-server0;
    }
    server {
      interfaces {
        cb0 p3p1;
        cb1 p3p2;
        jdm-management em2;
        vnf-management em3;
      }
    }
  }
  interfaces {
    jmgmt0 {
      unit 0 {
        family inet {
          address 10.216.105.112/21;
        }
      }
    }
  }
  routing-options {
    static {
      route {
        0.0.0.0/0 next-hop 10.216.111.254;
      }
    }
  }
}

```

```
    }
  }
}
server1 {
  system {
    host-name test-jdm-server1;
  }
  server {
    interfaces {
      cb0 p3p1;
      cb1 p3p2;
      jdm-management em2;
      vnf-management em3;
    }
  }
  interfaces {
    jmgmt0 {
      unit 0 {
        family inet {
          address 10.216.105.113/21;
        }
      }
    }
  }
  routing-options {
    static {
      route {
        0.0.0.0/0 next-hop 10.216.111.254;
      }
    }
  }
}
}
}
apply-groups [ server0 server1 ];
system {
  root-authentication {
    encrypted-password "..."; ## SECRET-DATA
  }
  services {
    ssh;
    netconf {
      ssh;
      rfc-compliant;
    }
  }
}
virtual-network-functions {
  test-gnf {
    id 1;
    chassis-type mx2020;
    resource-template 2core-16g;
    base-config /var/jdm-usr/gnf-config/test-gnf.conf;
  }
}
```


Sample JDM Configuration (In-Chassis Model)

```
root@test-jdm-server0> show configuration
groups {
  server0 {
    system {
      host-name test-jdm-server0;
    }
    interfaces {
      jmgmt0 {
        unit 0 {
          family inet {
            address 10.216.105.112/21;
          }
        }
      }
    }
    routing-options {
      static {
        route {
          0.0.0.0/0 next-hop 10.216.111.254;
        }
      }
    }
  }
  server1 {
    system {
      host-name test-jdm-server1;
    }
    interfaces {
      jmgmt0 {
        unit 0 {
          family inet {
            address 10.216.105.113/21;
          }
        }
      }
    }
    routing-options {
      static {
        route {
          0.0.0.0/0 next-hop 10.216.111.254;
        }
      }
    }
  }
}
apply-groups [ server0 server1 ];
system {
  root-authentication {
    encrypted-password "..."; ## SECRET-DATA
  }
  services {
    ssh;
  }
}
```

```

netconf {
  ssh;
  rfc-compliant;
}
}
virtual-network-functions {
  test-gnf {
    id 1;
    resource-template 2core-16g;
    base-config /var/jdm-usr/gnf-config/test-gnf.conf;
  }
}

```

Sample BSYS Configuration with Abstracted Fabric Interface

```

user@router> show configuration chassis
network-slices {
  guest-network-functions {
    gnf 1 {
      af2 {
        peer-gnf id 2 af1;
      }
      af4 {
        peer-gnf id 4 af1;
      }
      description gnf-a;
      fpcs [ 0 19];
    }
    gnf 2 {
      af1 {
        peer-gnf id 1 af2;
      }
      af4 {
        peer-gnf id 4 af2;
      }
      description gnf-b;
      fpcs [ 1 6 ];
    }
    gnf 4 {
      af1 {
        peer-gnf id 1 af4;
      }
      af2 {
        peer-gnf id 2 af4;
      }
      description gnf-d;
      fpcs [ 3 4 ];
    }
  }
}

```

Sample Abstracted Fabric Configuration at GNF with Class of Service

Assume that there is an Abstracted Fabric (**af**) interface between GNF1 and GNF2. The following sample configuration illustrates how to apply rewrites on the **af** interface at GNF1 and apply classifiers on the **af** interface on GNF2, in a scenario where traffic comes from GNF1 to GNF2:

GNF1 Configuration

```

interfaces {
  xe-4/0/0 {
    unit 0 {
      family inet {
        address 22.1.2.2/24;
      }
    }
  }
  af2 {
    unit 0 {
      family inet {
        address 32.1.2.1/24;
      }
    }
  }
}
class-of-service {
  classifiers {
    dscp testdscp {
      forwarding-class assured-forwarding {
        loss-priority low code-points [ 001001 000000 ];
      }
    }
  }
}
interfaces {
  xe-4/0/0 {
    unit 0 {
      classifiers {
        dscp testdscp;
      }
    }
  }
  classifiers {
    dscp testdscp;
  }
}
af1 {
  unit 0 {
    rewrite-rules {
      dscp testdscp; /*Rewrite rule applied on egress AF interface on GNF1.*/
    }
  }
}
rewrite-rules {
  dscp testdscp {

```

```

        forwarding-class assured-forwarding {
            loss-priority low code-point 001001;
        }
    }
}

```

GNF2 Configuration

```

interfaces {
    xe-3/0/0:0 {
        unit 0 {
            family inet {
                address 42.1.2.1/24;
            }
        }
    }
    af1 {
        unit 0 {
            family inet {
                address 32.1.2.2/24;
            }
        }
    }
}
class-of-service {
    classifiers {
        dscp testdscp {
            forwarding-class network-control {
                loss-priority low code-points 001001;
            }
        }
    }
    interfaces {
        af1 {
            unit 0 {
                classifiers {
                    dscp testdscp; /*Classifier applied on AF at ingress of GNF2*/
                }
            }
        }
    }
}

```

Sample Output for Abstracted Fabric Interface State at a GNF

user@router-gnf-b> show interfaces af9

```

Physical interface: af9, Enabled, Physical link is Up
Interface index: 209, SNMP ifIndex: 527
Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 370000mbps
Device flags   : Present Running
Interface flags: Internal: 0x4000
Link type      : Full-Duplex
Link flags     : None

```

```

Current address: 00:90:69:2b:00:4c, Hardware address: 00:90:69:2b:00:4c
Last flapped   : 2018-09-12 01:44:01 PDT (00:01:02 ago)
Input rate    : 0 bps (0 pps)
Output rate   : 0 bps (0 pps)
Bandwidth     : 370 Gbps
Peer GNF id   : 9
Peer GNF Forwarding element(FE) view :
FPC slot:FE num  FE Bandwidth(Gbps) Status    Transmit Packets    Transmit
Bytes
    6:0              130      Up              0
    0
    12:0             120      Up              0
    0
    12:1             120      Up              0
    0

Residual Transmit Statistics :
Packets :              0 Bytes :              0

Fabric Queue Statistics :
FPC slot:FE num    High priority(pkts)    Low priority(pkts)
    6:0              0                    0
    12:0              0                    0
    12:1              0                    0
FPC slot:FE num    High priority(bytes)    Low priority(bytes)
    6:0              0                    0
    12:0              0                    0
    12:1              0                    0

Residual Queue Statistics :
    High priority(pkts)    Low priority(pkts)
            0              0
    High priority(bytes)    Low priority(bytes)
            0              0

Logical interface af9.0 (Index 332) (SNMP ifIndex 528)
Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
Input packets : 0
Output packets: 13
Protocol inet, MTU: 1500

```

- Related Documentation**
- [Minimum Hardware and Software Requirements for Junos Node Slicing on page 29](#)
 - [Connecting the Servers and the Router on page 34](#)
 - [Components of Junos Node Slicing on page 18](#)

Setting Up YANG-Based Orchestration of GNFs

Starting from Junos OS Release 17.4R1, Junos Node Slicing supports YANG-based abstraction to orchestrate guest network functions (GNFs), using single touchpoint.

In the single touchpoint method, a software-defined networking (SDN) controller (for example, OpenDaylight or ODL), communicates with the base system (BSYS)—that is, the BSYS acts as the single touchpoint. The BSYS receives the RPC requests from the ODL controller, parses the RPC and then forwards the adequate RPC to the JDM (based

on scripts available at the BSYS). On getting the response from the JDM, the BSYS parses and forwards the response back to ODL.



NOTE: Junos Node Slicing also supports GNF life cycle management using the dual touchpoint method. In this method, ODL sends RPCs to, and receive responses from, JDM and BSYS separately. To enable dual touchpoint, you need to mount both BSYS and Juniper Device Manager (JDM) on ODL.

To enable single touchpoint, you need to perform the following steps.

- [Installing YANG Package on BSYS on page 70](#)
- [Sharing the SSH Identities Between BSYS and JDM on page 70](#)
- [Configuring the BSYS to Communicate with JDM on page 71](#)
- [Supported XML RPCs: YANG-Based Abstraction of GNFs on page 71](#)
- [Disabling the YANG-based Orchestration of GNFs on page 74](#)

Installing YANG Package on BSYS

In the node slicing setup, two servers (referred to as Server0 and Server1) are connected to an MX series router (or the BSYS). On the BSYS, you need to install the YANG package to enable communication between the BSYS and the JDM.

To install the YANG package on the BSYS, use the following command:

```
root@router> request system software add /var/tmp/junos-node-slicing-x86-32-17.4.tgz
```



NOTE: You need to install the same package on the backup Routing Engine as well.

After successful installation, you can find the YANG Package contents at the following location:

```
root@router:/opt/yang-pkg/junos-node-slicing
```

You can verify the package installation status on both the Routing Engines of the BSYS, using the following command:

```
root@router> show system yang package
```

Sharing the SSH Identities Between BSYS and JDM

To synchronize configurations between the BSYS and the JDM (on server0 and server1), the SSH identities need to be shared between the BSYS and the JDM. It is mandatory to share the authorization keys for passwordless authentication from BSYS to JDM.

**NOTE:**

- Ensure that server0 and server1 are up and running.
- Ensure that you replace the jdm-server0-ip and jdm-server1-ip with proper IP addresses.

Do the following operation on BSYS for JDM on server0 and server1:

1. Go to the BSYS shell.

```
root@router> start shell
```

2. Issue the following commands:

```
root@router:~ # ssh-keygen -t rsa
root@router:~ # ssh root@jdm-server0-ip mkdir -p .ssh
root@router:~ # cat .ssh/id_rsa.pub | ssh root@jdm-server0-ip 'cat >> .ssh/authorized_keys'
root@router:~ # ssh root@jdm-server0-ip "chmod 700 .ssh; chmod 640 .ssh/authorized_keys"

root@router:~ # ssh-keygen -t rsa
root@router:~ # ssh root@jdm-server1-ip mkdir -p .ssh
root@router:~ # cat .ssh/id_rsa.pub | ssh root@jdm-server1-ip 'cat >> .ssh/authorized_keys'
root@router:~ # ssh root@jdm-server1-ip "chmod 700 .ssh; chmod 640 .ssh/authorized_keys"
```

Configuring the BSYS to Communicate with JDM



NOTE: This is a one-time setup.

To complete the single touchpoint setup, you need to set the BSYS to communicate with the JDM. This enables the BSYS to service the RPC requests from ODL.

To enable the BSYS to communicate with the JDM, issue the following commands on the BSYS.

```
root@router# set jdm-configuration:jdm jdm-management server0 jdm-server0-ip-addr
root@router# set jdm-configuration:jdm jdm-management server1 jdm-server1-ip-addr
root@router# set system scripts commit file jdm-config-check.slax
root@router# set event-options policy JNS events UI_COMMIT_COMPLETED
root@router# set event-options policy JNS then event-script jdm-config-commit.slax
root@router# set event-options event-script file jdm-config-commit.slax
root@router# commit and-quit
```

Supported XML RPCs: YANG-Based Abstraction of GNFs

Table 8 on page 72 lists the key GNF management tasks, along with the XML RPCs that are used to perform those tasks.

**NOTE:**

- In the single touchpoint method, use the XML with the prefix `jdm-`, as seen in [Table 8 on page 72](#) (for example, `<rpc><jdm-get-route-information/></rpc>`).
- In the dual touchpoint method, use the XML without the `jdm-` prefix (for example, `<rpc><get-route-information/></rpc>`).

Table 8: Supported XML RPCs to Manage GNFs

Task	RPC
Add Junos OS package for a GNF.	<pre> <rpc> <jdm-request-virtual-network-functions> <vnf-name> GNF-NAME </vnf-name> <add-image> IMAGE-PATH </add-image> <all-servers/> </jdm-request-virtual-network-functions> </rpc> </pre>
Delete Junos OS package of a GNF.	<pre> <rpc> <jdm-request-virtual-network-functions> <vnf-name> GNF-NAME </vnf-name> <delete-image/> <server0/> </jdm-request-virtual-network-functions> </rpc> </pre>
Show information about a specific GNF.	<pre> <rpc> <jdm-get-virtual-network-functions> <vnf-name> GNF-NAME </vnf-name> <server1/> </jdm-get-virtual-network-functions> </rpc> </pre>
Stop a GNF.	<pre> <rpc> <jdm-request-virtual-network-functions> <vnf-name> GNF-NAME </vnf-name> <stop/> </jdm-request-virtual-network-functions> </rpc> </pre>

Table 8: Supported XML RPCs to Manage GNFs (continued)

Task	RPC
Start a GNF.	<pre> <rpc> <jdm-request-virtual-network-functions> <vnf-name> GNF-NAME </vnf-name> <start/> </jdm-request-virtual-network-functions> </rpc> </pre>
Restart a GNF.	<pre> <rpc> <jdm-request-virtual-network-functions> <vnf-name> GNF-NAME </vnf-name> <restart/> </jdm-request-virtual-network-functions> </rpc> </pre>
Get information about the JDM software.	<pre> <rpc> <jdm-get-software-information/> </rpc> </pre>
Get route information.	<pre> <rpc> <jdm-get-route-information/> </rpc> </pre>
Get information about JDM server connections.	<pre> <rpc> <jdm-get-server-connections/> </rpc> </pre>
Get the list of JDM bridges.	<pre> <rpc> <jdm-get-bridges-list/> </rpc> </pre>
Get information about JDM interfaces.	<pre> <rpc> <jdm-get-interface-information/> </rpc> </pre>
Get list of GNFs.	<pre> <rpc> <jdm-get-inventory-software-vnf-information/> </rpc> </pre>

Table 8: Supported XML RPCs to Manage GNFs (continued)

Get the GNF visibility details.

```
<rpc>

<jdm-get-visibility-vnf-information/>
</rpc>
```

Disabling the YANG-based Orchestration of GNFs

If you want to disable the YANG-based orchestration of GNFs, delete the YANG package. Also, in case of a software upgrade, you need to delete the existing package and install the latest one.

To delete the YANG package:

1. Issue the following commands on the BSYS:

```
root@router# delete jdm-configuration-jdm
root@router# delete system scripts commit file jdm-config-check.slax
root@router# delete event-options policy JNS events UI_COMMIT_COMPLETED
root@router# delete event-options policy JNS then event-script jdm-config-commit.slax
root@router# delete event-options event-script file jdm-config-commit.slax
root@router# commit and-quit
```

2. Issue the following command on both the master and backup Routing Engines:

```
root@router> request system software delete junos-node-slicing
```

Related Documentation

- [Junos Node Slicing Upgrade on page 75](#)
- [Configuring Abstracted Fabric Interfaces Between a Pair of GNFs on page 59](#)

CHAPTER 3

Upgrading and Managing Junos Node Slicing

- [Junos Node Slicing Upgrade on page 75](#)
- [Managing Junos Node Slicing on page 89](#)

Junos Node Slicing Upgrade

Junos Node Slicing upgrade involves upgrading Juniper Device Manager (JDM), guest network functions (GNFs), and the base system (BSYS).

- [Upgrading Junos Node Slicing on page 76](#)
- [Downgrading JDM for External Server Model on page 78](#)
- [Downgrading JDM for In-Chassis Model on page 81](#)
- [Unified ISSU Support on page 82](#)
- [Managing Multiversion Software Interoperability on page 82](#)
- [Restarting External Servers on page 85](#)
- [Updating Host OS on the External Servers on page 86](#)
- [Applying Security Updates to Host OS on page 86](#)
- [Applying Security Patches for Ubuntu Container on page 88](#)

Upgrading Junos Node Slicing

Junos Node Slicing comprises three types of software components:

- Juniper Device Manager (JDM) package
- Junos OS image for guest network function (GNFs)
- Junos OS package for base system (BSYS)

You can upgrade each of these components independently, as long as they are within the allowed range of software versions (see [“Multiversion Software Interoperability Overview” on page 26](#) for more details). You can also upgrade all of them together.



NOTE: Before starting the upgrade process, save the JDM, GNF VM, and BSYS configurations for reference.

- [Upgrading JDM for External Server Model on page 77](#)
- [Upgrading JDM for In-Chassis Model on page 77](#)
- [Upgrading GNF and BSYS on page 78](#)

Upgrading JDM for External Server Model

1. Upgrade the JDM by performing the following tasks on both the servers:
 - a. Copy the new JDM package (RPM or Ubuntu) to a directory on the host (for example, `/var/tmp`).
 - b. Stop the JDM by using the following command:

```
root@Linux server0# jdm stop
Stopping JDM
```

- c. Issue the upgrade command to upgrade the JDM package:

If you are upgrading the JDM RHEL package, use the following command:

```
root@Linux server0# rpm -U package_name.rpm --force
```

If you are upgrading the JDM Ubuntu package, use the following command:

```
root@Linux server0# dpkg -i deb package.deb
```



NOTE: A JDM upgrade does not affect any of the running GNFs.

See also:

- [Installing JDM RPM Package on x86 Servers Running RHEL \(External Server Model\) on page 42](#)
- [Installing JDM Ubuntu Package on x86 Servers Running Ubuntu 16.04 \(External Server Model\) on page 43](#)

Upgrading JDM for In-Chassis Model

1. Upgrade the JDM by performing the following tasks on the BSYS instance of both the routing engines:
 - a. Copy the new JDM RPM package to a directory (for example, `/var/tmp`).
 - b. Stop the JDM by running the following command:

```
root@router> request vmhost jdm stop
```

- c. Install the JDM RPM package for in-chassis Junos Node Slicing, by using the command shown in the following example:

```
root@router> request vmhost jdm add jns-jdm-vmhost-18.3-20180930.0.x86_64.rpm
```



NOTE: A JDM upgrade does not affect any of the running GNFs.



NOTE: In order to upgrade JDM for in-chassis model, you need not uninstall the existing JDM software. Uninstalling the existing JDM might impact the guest network functions (GNFs).

See Also • [Installing and Configuring JDM for In-Chassis Model on page 52](#)

Upgrading GNF and BSYS

The GNF and BSYS packages can be upgraded in the same way as you would upgrade Junos OS on a standalone MX Series router.

Ensure that all GNFs are online when you perform an upgrade. This is because both GNF and BSYS upgrade processes trigger multiversion checks (covered later in this guide), and all GNFs are required to be online during the multiversion check phase, failing which the upgrade will be aborted. In case a GNF remains shut down, you must deactivate its configuration from BSYS CLI, which will result in skipping multiversion checks for that particular GNF.



NOTE: A **force** option is also available, through which you can overwrite an existing GNF image with a new one by using the JDM CLI command `request virtual-network-functions vnf-name add-image new-image-name force`. This can be useful in a rare situation where the GNF image does not boot. You can also use the **force** option to perform a cleanup if, for example, you abruptly terminated an earlier `add-image` that was in progress, by pressing Ctrl-C (example: `request virtual-network-functions vnf-name delete-image image-name force`).

Downgrading JDM for External Server Model



NOTE: You cannot downgrade Juniper Device Manager (JDM) installed in a single-server based Junos Node Slicing setup.

Use the following steps to downgrade JDM:

1. Acquire mastership to the backup GNFs running on server1.

```
user@gnf> request chassis routing-engine master acquire no-confirm
Resolving mastership...
Complete. The local routing engine becomes the master.

user@gnf# commit synchronize

re1:
configuration check succeeds
re0:
commit complete
re1:
commit complete
```

2. On server0, stop all the GNFs and delete the **commit synchronize** configuration.

```
user@jdm> request virtual-network-functions test-gnf stop
test-gnf stopped

user@jdm# delete system commit synchronize
user@jdm# commit

server0:
configuration check succeeds
server1:
commit complete
server0:
commit complete
```

3. On server0, stop and uninstall JDM.

```
[user@server0 ~]# jdm stop
Stopping JDM

[user@server0 ~]# rpm -e jns-jdm

Detailed log of jdm setup saved in /var/log/jns-jdm-setup.log
Cleanup jdm from host...
Cleaning up jdm rootfs and bridges..
Domain jdm has been undefined

Done Cleanup jdm from host
```

4. On server0, install the target version of JDM.

```
[user@server0]# rpm -ivh jns-jdm-18.3-20181207.0.x86_64.rpm

Preparing... ##### [100%]
Detailed log of jdm setup saved in /var/log/jns-jdm-setup.log
```

```

Updating / installing...
 1:jns-jdm-18.3-20181207.0      ##### [100%]
Setup host for jdm...
Launch libvirtd in listening mode
Done Setup host for jdm
Installing /juniper/.tmp-jdm-install/juniper_ubuntu_rootfs.tgz...
Configure /juniper/lxc/jdm/jdm1/rootfs...
Configure /juniper/lxc/jdm/jdm1/rootfs DONE
Created symlink from /etc/systemd/system/multi-user.target.wants/jdm.service
to /usr/lib/systemd/system/jdm.service.
Done Setup jdm
Redirecting to /bin/systemctl restart rsyslog.service

```

5. Configure JDM with root authentication or interfaces, and routing-options.
6. On server0 JDM, add a GNF image version that is compatible with the JDM version.

```

user@jdm> request virtual-network-functions add-image
/var/tmp/junos-install-ns-mx-x86-64-18.3-R1.tgz gnf
Added Image

```

In case the GNF version is incompatible with the JDM version, the following error message is shown:

```

user@jdm> request virtual-network-functions test add-image
/var/jdm-usr/gnf-images/junos-install-ns-mx-x86-64-19.1-20181212_dev_common.0.tgz

SMBIOS version of GNF(v2) is incompatible with JDM(v1)

```

7. Wait till the GNF comes up on server0 JDM.
8. Perform a commit synchronize from the Master RE (which is the GNF running on server1).

```

user@gnf# commit synchronize

```

9. Assign mastership to the GNF which is running on server0 JDM.
10. On Server 1, repeat the steps 2 through 5.
11. Run the **request server authenticate-peer-server** command on both the servers.

```

user@jdm> request server authenticate-peer-server

```

12. Apply **show server connections all-servers** and ensure that no issues are seen.
13. Configure **set system commit synchronize** and then apply **commit** on server0 JDM.

```

user@jdm# set system commit synchronize

```



```
user@jdm# commit synchronize
```

14. Use the command **show virtual-network-functions all-servers** to see if the GNFs are coming up.

Downgrading JDM for In-Chassis Model



NOTE: You cannot downgrade Juniper Device Manager (JDM) installed in a single Routing Engine-based Junos Node Slicing setup.

Use the following steps to downgrade JDM:

1. Assign mastership to the backup GNFs running on Routing Engine 1 (re1).

```
user@gnf> request chassis routing-engine master switch no-confirm
```

2. On re0, stop all the GNFs and delete the **commit synchronize** configuration.

```
user@jdm> request virtual-network-functions stop server0 gnf
user@jdm# delete system commit synchronize
user@jdm# commit
```

3. On re0, uninstall JDM (on BSYS master).

```
user@bsys> request vmhost jdm delete
```

4. On re0, install the target version (example: 18.3R1) of JDM.

```
user@bsys> request vmhost jdm add /var/tmp/jns-jdm-vmhost-18.3-R1.3.x86_64.rpm
```

5. On re0, deploy the same version of GNF which is running on server1.

```
user@jdm> request virtual-network-functions add-image
/var/tmp/junos-install-ns-mx-x86-64-19.1-20181115.1.tgz gnf
```

In case the GNF version is incompatible with the JDM version, the following error message is shown:

```
user@jdm> request virtual-network-functions test add-image
/var/jdm-usr/gnf-images/junos-install-ns-mx-x86-64-19.1-20181212_dev_common.0.tgz

SMBIOS version of GNF(v2) is incompatible with JDM(v1)
```

You can use the following command to check the GNF version.

```
user@gnf1> show version
```

```
Hostname: gnf1
Model: mx960
Junos: 19.1-20181115.1
```

6. On re1, repeat the steps 1 through 5.

7. Run the **request server authenticate-peer-server** command on both the Routing Engines.

```
user@jdm> request server authenticate-peer-server
```

8. Perform a commit synchronize from the Master RE (which is the GNF running on server1).

```
user@gnf# commit synchronize
```

9. Configure **set system commit synchronize** and then apply **commit** on re0 JDM.

```
user@jdm# set system commit synchronize
user@jdm# commit synchronize
```

Now, JDM is up with Junos OS version 18.3R1.

Unified ISSU Support

Junos Node Slicing also supports unified in-service software upgrade (ISSU), enabling you to upgrade between two different Junos OS versions with no disruption on the control plane and with minimal disruption of traffic. You can perform unified ISSU on BSYS and GNFs separately. Also, you can run unified ISSU on each GNF independently—without affecting other GNFs. See also [Understanding the Unified ISSU Process](#).



NOTE: The multiversion software support restrictions (such as version deviation limits) are applicable to unified ISSU upgrade as well.

Managing Multiversion Software Interoperability

Junos Node Slicing supports multiversion software interoperability. However, if there are any incompatibilities between software versions, alert messages appear during the software upgrade process or when a GNF or a FRU comes online. When minor incompatibilities occur, you can choose to accept them and proceed. In case of a major incompatibility, you need to either abort the process or use the **force** option to accept the incompatibility and proceed.



NOTE: In case of vmhost software upgrade, the **force** option is not available. Therefore, if a GNF is offline or is incompatible with the software being installed, and is causing multiversion checks to abort, you need to deactivate that GNF during the software upgrade and then reactivate it once the upgrade is over.

The following are sample messages that appear if incompatibilities are detected during software upgrade:

Sample alert message indicating a minor incompatibility:

```

user@router> request system software add
/var/tmp/junos-install-mx-x86-64-17.4-20170703_dev_common.0.tgz
Starting Multiversion compatibility checks for package
/var/tmp/junos-install-mx-x86-64-17.4-20170703_dev_common.0.tgz
Starting compatibility checks...
-----
System Anomalies:
-----
Ano-ID  ACTION  MESSAGE
-----
      100    WARN  <sample system incompatibility 1>
Accept incompatibility? [yes,no] (no) yes

      103    WARN  <sample system incompatibility 2>
Accept incompatibility? [yes,no] (no) yes
-----
CFG Anomalies for: set snmp interface
-----
FRU-ID  Ano-ID  ACTION  MESSAGE
-----
NONE      102    WARN  <sample config incompatibility 1>
Accept incompatibility? [yes,no] (no) yes

NONE      105    WARN  <sample config incompatibility 2>
Accept incompatibility? [yes,no] (no) yes
-----
FRU Anomalies:
-----
FRU-ID  Ano-ID  ACTION  MESSAGE
-----
0xaa0b      100    WARN  <sample FRU incompatibility 1>
Accept incompatibility? [yes,no] (no) yes

0xbb0b      101    WARN  <sample FRU incompatibility 2>
Accept incompatibility? [yes,no] (no) yes

Compatibility Checks done... OK
NOTICE: Validating configuration against
junos-install-mx-x86-64-17.4-20170703_dev_common.0.tgz.
NOTICE: Use the 'no-validate' option to skip this if desired.
Verified junos-install-mx-x86-64-17.4-20170703_dev_common.0 signed by
PackageDevelopmentEc_2017 method ECDSA256+SHA256

```

Sample alert message indicating a major incompatibility:

```

user@router> request system software add
/var/tmp/junos-install-mx-x86-64-17.4-20170713_0718.tgz

```

```
Starting Multiversion compatibility checks for package
/var/tmp/junos-install-mx-x86-64-17.4I20170713_0718.tgz
Starting compatibility checks...
-----
System Anomalies:
-----
Ano-ID  ACTION  MESSAGE
-----
1677721600  ABORT  <sample system incompatibility 1>

error: Junos-Node-Slicing multi-version checks returned abort for package
/var/tmp/junos-install-mx-x86-64-17.4I20170713_0718.tgz
```

Sample output showing how to use the 'force' option to proceed with an upgrade:

```
user@router> request system software add
/var/tmp/junos-install-mx-x86-64-17.4I20170713_0718.tgz force

NOTICE: Validating configuration against
junos-install-mx-x86-64-17.4I20170713_0718.tgz.
NOTICE: Use the 'no-validate' option to skip this if desired.
Verified junos-install-mx-x86-64-17.4I20170713_0718 signed by
PackageDevelopmentEc_2017 method ECDSA256+SHA256
Verified manifest signed by PackageDevelopmentEc_2017 method ECDSA256+SHA256
Checking PIC combinations
Adding junos-x86-64-17.4I20170713_0718...
```

- [Viewing Software Incompatibility Alarms on page 84](#)
- [Viewing Incompatibilities Between Software Versions on page 84](#)

Viewing Software Incompatibility Alarms

After a software update of a GNF or BSYS, if software incompatibilities between the GNF and the BSYS exist, they will be raised as a chassis alarm. You can view the incompatibility alarm information by using the **show chassis alarms** command. You can further view the details of the incompatibilities by using the **show system anomalies** command. For more details, see [“Viewing Incompatibilities Between Software Versions” on page 84](#).

The alarms appear only on GNFs even if the upgrade is performed on the BSYS. The following types of alarm can occur:

- System Incompatibility with BSYS—This is a major alarm. It appears when any incompatibilities between BSYS and GNF software versions cause the GNF to go offline.
- Feature Incompatibility with BSYS—This is a minor alarm. It indicates a minor incompatibility between BSYS and GNF software versions. This does not cause the GNF to go offline.

Viewing Incompatibilities Between Software Versions

To view software incompatibilities from the BSYS, use the CLI as shown in the following example:

```
user@router> show system anomalies gnf-id 4 system
```

To view software incompatibilities from a GNF, use the CLI as shown in the following example:

```
user@router> show system anomalies system
```



NOTE:

- As shown in the CLI, remember to specify the GNF ID while viewing the incompatibilities from BSYS.
- The preceding examples show system-level incompatibilities. Use the `fru` or `config` options to view FRU or feature-level incompatibilities.

Restarting External Servers

Server maintenance activities such as hardware or host OS upgrade and fault isolation might require you to restart the external servers used in Junos Node Slicing. Use the following procedure to restart the servers:

1. Stop all the GNFs.

If you are restarting both the servers, choose the **all-servers** option while stopping each GNF as shown in the following example:

```
root@server1> request virtual-network-functions gnf_name stop all-servers
gnf_name stopped
```

If you are restarting a particular server, stop the GNFs on that server by specifying the server-id as shown in the following example:

```
root@server1> request virtual-network-functions gnf_name stop server0
gnf_name stopped
```

2. Verify that the GNFs have been stopped.

```
root@server1> show virtual-network-functions
```

ID	Name	State	Liveness
1	mgb-gnf-b	Shutdown	down



NOTE: If you want to view the status of GNFs on both the servers, choose the **all-servers** option. Example: `show virtual-network-functions all-servers`).

3. From the Linux host shell, stop the JDM by using the following command:

```
[root@HostLinux ~]# jdm stop
```

```
Stopping JDM
```

4. From the Linux host shell, verify that the JDM status shows as stopped.

```
[root@HostLinux ~]# jdm status
```

```
JDM is stopped
```

5. After rebooting, verify that the JDM status now shows as running.

```
[root@HostLinux ~]# jdm status
```

```
JDM (pid 2828) is running as server1
```

After a server reboot, the JDM and the configured GNFs will automatically start running.

If you are replacing the servers, ensure that the operating server pair continues to have similar or identical hardware configuration. If the server pair were to become temporarily dissimilar during the replacement (this could be the case when replacing the servers sequentially), it is recommended that you disable GRES and NSR for this period, and re-enable them only when both the servers are similar once again.

Updating Host OS on the External Servers

Before updating the host OS on an external server, you must first stop the GNFs and JDM on that server as described in [“Restarting External Servers” on page 85](#).

Following the host OS update, if you are using Intel X710 NICs, ensure that the version of the X710 NIC driver in use continues to be the latest version as described in [“Updating Intel X710 NIC Driver for x86 Servers” on page 38](#).

Applying Security Updates to Host OS

The host OS requires security updates from time to time. This section highlights the steps involved in applying Security Updates to the host OS using Red Hat (RHEL) OS.

Junos Node Slicing supports RHEL 7.3.

Before doing any updates to the host OS, ensure that Red Hat Subscription Manager is set to version 7.3 and that Red Hat Subscription Service includes Extended Update Support (EUS).

You can use the command **subscription-manager release --show** to confirm that the release is set to 7.3. If it is not, you can use the command **subscription-manager release --set=7.3** to set the release to 7.3.



NOTE: You must ensure that the Red Hat Subscription Manager is set to version 7.3. Otherwise, updates to the RHEL will attempt to upgrade to the latest minor release. For example, RHEL 7.3 could become RHEL 7.4 (or a later version) with a general yum update, or a yum security update can pull in a new kernel beyond RHEL 7.3.

Red Hat's extended update support allows for patches and security updates to be applied within the specified release. Allowed use of RHEL's Extended Update support is a function of the RHEL support contract and beyond the scope of this section. You can check to see if your RHEL subscription includes Extended Update Support (EUS), by using the command `subscription-manager repos --list | grep rhel-7-server-eus-rpms`. EUS support is not enabled by default. EUS can be enabled, by using the command `subscription-manager repos --enable rhel-7-server-eus-rpms`.

Steps to Apply Host OS Security Updates

Applying security updates to host OS will likely require you to reboot the external x86 servers. See the [“Updating Host OS on the External Servers” on page 86](#) topic.

It is also possible that a host OS security update will bring in a new kernel version. Updating the host OS kernel could also overwrite the Intel i40e driver to bring in a version of it that does not meet the i40e driver minimum version requirements. If so, you must update the i40e driver to meet the minimum requirements. For more details, see [“Updating Intel X710 NIC Driver for x86 Servers” on page 38](#).

Before rebooting the external x86 servers, you must stop all GNF VMs and JDM on that server. Since we have two external x86 servers, the host OS Security Updates can be done without disrupting GNF forwarding, by updating one server at a time. A GRES/NSR Master Routing Engine switch-over is required to move the Master Routing Engine away from the affected server.

We start with the default behavior of Routing Engine 1 (**re1**) as the Backup Routing Engine for each GNF where **re1** for each GNF is running on the external x86 server1.

1. Back up all configurations.
2. Gather view of host OS kernel and package versions on the external x86 servers before the host OS security update. Also confirm i40e driver and Intel X710 firmware meet minimum requirements (version: 2.4.10 and version: 18.5.17).

```
user@server# cat /etc/redhat-release
user@server# uname -r
user@server# uname -a
user@server# rpm -q kernel
user@server# ethtool -i p3p1
```

3. Ensure that RedHat Subscription Manager is set to RHEL 7.3 and the EUS Repository is enabled.

```
[user@server ~]# subscription-manager version
```

```
[user@server ~]# subscription-manager repos --list | grep rhel-7-server-eus-rpms
```

4. Ensure all GNFs are using Master RE on **server0**. The backup Routing Engine is **re1** on **server1**. First perform host OS security updates on the server that contains the backup Routing Engines.

```
user@router> show chassis routing-engine
```

Run this command on all the GNFs to confirm that all the GNFs have their master Routing Engine on server0.

5. Stop all GNF VMs in JDM cli via request **stop** on **server1** only. **server1** contains the backup Routing Engines for all the GNFs. Do not use the **all-servers** option. Example:

```
user@jdm> request virtual-network-functions gnf-a stop server 1
user@jdm> request virtual-network-functions gnf-b stop server 1
```

6. Stop JDM on the affected server from the host OS.

```
user@server# jdm status
user@server# jdm stop
```

7. Do the **yum** security update and reboot the server.

```
user@server# yum -y update -security
root@server# shutdown -r now
```

8. Reload or compile the i40e Driver. See the [Intel support page](#) for instructions on updating the driver.

At this point, the host OS security update to **server1** is done. Note that the GNF VMs start up on server reboot.

9. After the security updates are completed, the server rebooted and the GNFs are back up, repeat on the other server.

Applying Security Patches for Ubuntu Container

The Ubuntu container, which Juniper Device Manager (JDM) is based on, needs to have security patches applied from time to time.



NOTE: JDM must be able to reach the internet and must have **name-server** configured. Apply the following JDM CLI configuration statement to specify the **name-server**:

```
root@jdm# set system name-server address
```

Use the following steps to apply security updates to the Ubuntu container components of JDM:

1. From host OS, use the JDM console to enter JDM as root.

```
root@server# jdm console
```

Or, from the JDM CLI, enter JDM shell by using the following command:

```
root@jdm> start shell user root
```

2. From the JDM shell, use the command **apt-get update** to download information about new packages or the latest versions of the currently installed packages.

```
jdm-srv1:~# sudo apt-get update
```

3. From the JDM shell, use the command **apt-get upgrade**.

```
jdm-srv1:~# sudo apt-get upgrade
```

You are shown a list of upgrades, and prompted to continue. Answer Y for yes and press **Enter**.

4. From the JDM shell, use the command **apt-get dist-upgrade** to perform the upgrade.

```
jdm-srv1:~# sudo apt-get dist-upgrade
```

Answer Y when prompted to continue, and wait for the upgrades to finish.

5. From the host OS, restart the JDM.

```
user@server# sudo jdm restart
```

Related Documentation

- [Junos Node Slicing Overview on page 17](#)
- [Multiversion Software Interoperability Overview on page 26](#)

Managing Junos Node Slicing

- [Deleting Guest Network Functions on page 89](#)
- [Disabling Junos Node Slicing on page 90](#)

Deleting Guest Network Functions

This procedure involves shutting down a GNF and then deleting it. In JDM, GNF VMs are called VNFs. Use the following steps to delete a VNF:

1. Shut down a VNF by using the JDM CLI command **request virtual-network-functions *gnf-name* stop all-servers**. For example:

```
root@test-jdm-server0> request virtual-network-functions test-gnf stop all-servers
```

```
server0:
-----
test-gnf stopped

server1:
-----
test-gnf stopped
```

2. Delete the VNF configuration by applying the JDM CLI configuration statement **delete virtual-network-functions *gnf-name***. See the following example:

```
root@test-jdm-server0# delete virtual-network-functions test-gnf
root@test-jdm-server0# commit synchronize
```

3. Delete the VNF image repository by using the JDM CLI command **request virtual-network-functions *gnf-name* delete-image all-servers**. For example:

```
root@test-jdm-server0> request virtual-network-functions test-gnf delete-image
all-servers
```

```
server0:
-----
Deleted the image repository
server1:
-----
Deleted the image repository
```

**NOTE:**

- To delete a VNF completely, you must perform all the three steps.
- If you want to delete a VNF management interface, you must stop and delete the VNF first.

Disabling Junos Node Slicing

To disable Junos Node Slicing, you must uninstall the following packages:

- JDM package
- Junos OS image for GNFS



NOTE: Save the JDM configuration if you want to use it for reference.

Use the following steps to disable Junos Node Slicing (external server model):

1. Delete the GNFs first by performing all the steps described in the section [“Deleting Guest Network Functions” on page 89](#).

2. Stop the JDM on each server by running the following command at the host Linux shell:

```
root@Linux server0# jdm stop
```

Stopping jdm: Domain jdm destroyed

3. Uninstall the JDM on each server by running the following command at the host Linux shell.

For the servers running RHEL, run the following command:

```
root@Linux server0# rpm -e jns-jdm
```

For the servers running Ubuntu, run the following command:

```
root@Linux server0# dpkg --remove jns-jdm
```

4. To revert the MX Series router from BSYS mode to standalone mode, apply the following configuration statements on the MX Series router:

```
root@router# delete chassis network-slices guest-network-functions
root@router# commit
```

The router now operates in standalone mode.

To disable in-chassis Junos Node Slicing, you must:

- Shut down and delete all GNFs. Also, delete the Junos OS image associated with the GNF.
- Shut down JDM and then delete the JDM software package.
- Delete the in-chassis BSYS mode configuration (**set vmhost resize vjunos**).
- Reboot the Routing Engine.

Use the following steps to disable in-chassis Junos Node Slicing:

1. Delete the GNFs first by performing all the steps described in the section [“Deleting Guest Network Functions” on page 89](#).

2. Stop the JDM on each Routing Engine by running the following command:

```
root@router> request vmhost jdm stop
```

3. Uninstall the JDM on each Routing Engine by running the following command.

```
root@router> request vmhost jdm delete
```

4. To revert the MX Series router from BSYS mode to standalone mode, apply the following configuration statements on the MX Series router:

```
root@router# delete vmhost resize vjunos
root@router# commit
```

5. Reboot VM host.

```
user@router> request vm host reboot (re0|re1)
```

The router now operates in standalone mode.



NOTE: All files in the `/var/` location, including the log files (`/var/log`) and core files (`/var/crash`), are deleted when you reboot VM host after deleting the `vmhost resize vjunos compact` configuration. You must save any files currently in `/var/log` or `/var/crash` before deleting the `vmhost resize vjunos compact` configuration if you want to use them for reference.

**Related
Documentation**

- [Junos Node Slicing Overview on page 17](#)
- [Components of Junos Node Slicing on page 18](#)
- [Sample Configuration for Junos Node Slicing on page 63](#)

CHAPTER 4

Configuration Statements for BSYS

- `af-name` on page 94
- `control-plane-bandwidth-percent` (Node Slicing) on page 95
- `description` (GNF) on page 96
- `description` (Abstracted Fabric) on page 97
- `fpcs` (Node Slicing) on page 98
- `network-slices` on page 99
- `gnf` on page 100
- `guest-network-functions` on page 101
- `peer-gnf` on page 102
- `vmhost resize vjunos compact` on page 102

af-name

Syntax

```
af-name {
  peer-gnf {
    id peer-gnf-id;
    remote-af-name;
  }
  description af-description;
}
```

Hierarchy Level [edit chassis network-slices guest-network-functions gnf id]

Release Information Statement introduced in Junos OS Release 17.4R1.

Description Configure Abstracted Fabric (**af**) interface between a pair of guest network functions (GNFs). Abstracted Fabric interface is a pseudo interface that represents a first class Ethernet interface behavior. An **af** interface is created on a GNF to communicate with the peer GNF when the two GNFs are connected to each other through the CLI.



NOTE: Only one **af** interface can be configured between a pair of GNFs.

Options **af-name**—Name of the **af** interface being created on the GNF.

Range: af0 through af9

id peer-gnf-id—Name of the GNF peer connected using the **af** interface.

Range: 1 through 10

remote-af-name—Name of the **af** interface on the peer GNF.

description af-description—A description for the **af** interface.

Required Privilege Level system

Related Documentation

- [Configuring Abstracted Fabric Interfaces Between a Pair of GNFs on page 59](#)

control-plane-bandwidth-percent (Node Slicing)

Syntax	<code>control-plane-bandwidth-percent <i>percent</i>;</code>
Hierarchy Level	<code>[edit chassis network-slices guest-network-functions gnf]</code>
Release Information	Statement introduced in Junos OS Release 17.2.
Description	Allocate a percentage of the bandwidth that exists on the control plane on the router to the specified guest network function (GNF). Allocating bandwidth prevents potential overutilization by one GNF over another.
Options	<i>percent</i> —Percentage of control plane bandwidth.
Required Privilege Level	view-level—To view this statement in the configuration. control-level—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• network-slices on page 99• guest-network-functions on page 101• gnf on page 100• description on page 96• fpcs on page 98


description (GNF)

Syntax	<code>description <i>description</i>;</code>
Hierarchy Level	<code>[edit chassis network-slices guest-network-functions gnf]</code>
Release Information	Statement introduced in Junos OS Release 17.2.
Description	Provide a description string for the specified guest network function (GNF).
Options	<i>description</i> —A description string for the specified guest network function (GNF).
Required Privilege Level	view-level—To view this statement in the configuration. control-level—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• network-slices on page 99• guest-network-functions on page 101• gnf on page 100• control-plane-bandwidth-percent on page 95• fpcs on page 98

description (Abstracted Fabric)

Syntax	<code>description <i>description</i>;</code>
Hierarchy Level	<code>[edit chassis network-slices guest-network-functions gnf af-name]</code>
Release Information	Statement introduced in Junos OS Release 17.4R1.
Description	Provide a description string for the specified Abstracted Fabric (af) interface.
Options	description —A description string for the specified Abstracted Fabric (af) interface.
Required Privilege Level	view-level—To view this statement in the configuration. control-level—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Configuring Abstracted Fabric Interfaces Between a Pair of GNFs on page 59• gnf on page 100• control-plane-bandwidth-percent on page 95• fpcs on page 98

fpcs (Node Slicing)

Syntax	<code>fpcs fpcs;</code>
Hierarchy Level	<code>[edit chassis network-slices guest-network-functions gnf]</code>
Release Information	Statement introduced in Junos OS Release 17.2.
Description	Assign FPCs to a guest network function.
<div> NOTE: A given FPC cannot be assigned to more than one GNF.</div>	
Options	fpcs —The FPC to be assigned to a guest network function. Range: 0–63
Required Privilege Level	view-level—To view this statement in the configuration. control-level—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• network-slices on page 99• guest-network-functions on page 101• gnf on page 100• control-plane-bandwidth-percent on page 95• description on page 96

network-slices

Syntax	<pre> network-slices { guest-network-functions{ gnf <i>id</i> { control-plane-bandwidth-percent <i>percent</i>; description <i>description</i>; fpcs <i>fpcs</i>; af-name } } } </pre>
Hierarchy Level	[edit chassis]
Release Information	Statement introduced in Junos OS Release 17.2.
Description	<p>Configure Junos Node Slicing.</p> <p>Junos Node Slicing enables a single MX Series router to be partitioned to appear as multiple, independent routers. Each partition has its own Junos control plane, which runs as a virtual machine (VM), and a dedicated set of line cards. Each partition is called a guest network function (GNF).</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>view-level—To view this statement in the configuration.</p> <p>control-level—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring an MX Series Router to Operate in BSYS Mode (External Server Model) on page 41 • Configuring Guest Network Functions on page 56

gnf

Syntax	<pre>gnf id { control-plane-bandwidth-percent <i>percent</i>; description <i>description</i>; fpcs <i>fpcs</i>; af-name }</pre>
Hierarchy Level	[edit chassis network-slices guest-network-functions]
Release Information	Statement introduced in Junos OS Release 17.2.
Description	<p>Define a GNF by assigning an ID to it.</p> <p>Junos Node Slicing enables a single MX Series router to be partitioned to appear as multiple, independent routers. Each partition has its own Junos control plane, which runs as a virtual machine (VM), and a dedicated set of line cards. Each partition is called a guest network function (GNF).</p>
Options	<p>id—GNF ID</p> <p>Range: 1–10</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>view-level—To view this statement in the configuration.</p> <p>control-level—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none">• Configuring Guest Network Functions on page 56• network-slices on page 99• guest-network-functions on page 101

guest-network-functions

Syntax

```

guest-network-functions {
  gnf id {
    control-plane-bandwidth-percent percent;
    description description;
    fpcs fpcs;
    af-name
  }
}

```

Hierarchy Level [edit chassis network-slices]

Release Information Statement introduced in Junos OS Release 17.2.

Description Configure a guest network function for Junos Node Slicing.

Junos Node Slicing enables a single MX Series router to be partitioned to appear as multiple, independent routers. Each partition has its own Junos control plane, which runs as a virtual machine (VM), and a dedicated set of line cards. Each partition is called a guest network function (GNF).

The remaining statements are explained separately.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- [Configuring Guest Network Functions on page 56](#)
- [network-slices on page 99](#)

peer-gnf

Syntax	<pre>peer-gnf { id <i>peer-gnf-id</i>; remote-af-name; }</pre>
Hierarchy Level	[edit chassis network-slices guest-network-functions gnf id <i>af-name</i>]
Release Information	Statement introduced in Junos OS Release 17.4R1.
Description	Configure the details of the GNF peer connected using the Abstracted Fabric (af) interface.
Options	<p>id<i>peer-gnf-id</i>—Name of the GNF peer connected using the Abstracted Fabric (af) interface.</p> <p>Range: 1 through 10</p> <p>remote-af-name—Name of the af interface on the peer GNF.</p>
Required Privilege Level	system
Related Documentation	<ul style="list-style-type: none">• Configuring Abstracted Fabric Interfaces Between a Pair of GNFs on page 59

vmhost resize vjunos compact

Syntax	<pre>vmhost resize vjunos compact</pre>
Hierarchy Level	[edit]
Release Information	Statement introduced in Junos OS Release 19.1R1.
Description	Reduce the routing engine resource footprint on MX Series router to accommodate GNF VMs within the Routing Engine in in-chassis Junos Node Slicing.
Required Privilege Level	<p>view-level—To view this statement in the configuration.</p> <p>control-level—To add this statement to the configuration.</p>

CHAPTER 5

Operational Commands for BSYS

- request vmhost jdm add (In-Chassis Model)
- request vmhost jdm delete (In-Chassis Model)
- request vmhost jdm start (In-Chassis Model)
- request vmhost jdm login (In-Chassis Model)
- request vmhost jdm stop (In-Chassis Model)
- show vmhost status (In-Chassis Model)
- show vmhost jdm status (In-Chassis Model)
- show chassis network-slices
- show chassis fpc pic-status (Node Slicing)
- show chassis fpc (Node Slicing)
- show chassis adc (Node Slicing)
- show chassis network-slices fpcs
- show system anomalies gnf-id (BSYS)

request vmhost jdm add (In-Chassis Model)

Syntax `request vmhost jdm add package-name`

Release Information Command introduced in Junos OS Release 19.1R1.

Description Adds the Juniper Device Manager (JDM) software package. This command is applicable to in-chassis Junos Node Slicing.

Options `add package-name`—Add the Juniper Device Manager (JDM) software package. You need to type the package name after the command `add`.

Required Privilege Level maintenance

Related Documentation

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [show virtual-network-functions on page 172](#)
- [request server authenticate-peer-server on page 170](#)

request vmhost jdm delete (In-Chassis Model)

Syntax request vmhost jdm delete

Release Information Command introduced in Junos OS Release 19.1R1.

Description Uninstalls the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos Node Slicing.

Required Privilege Level maintenance

Related Documentation

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [show virtual-network-functions on page 172](#)
- [request server authenticate-peer-server on page 170](#)

request vmhost jdm start (In-Chassis Model)

Syntax	request vmhost jdm start
Release Information	Command introduced in Junos OS Release 19.1R1.
Description	Starts the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos Node Slicing.
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none">• Generic Guidelines for Using JDM Server Commands on page 163• show virtual-network-functions on page 172• request server authenticate-peer-server on page 170

Sample Output

request vmhost jdm start

```
user@router> request vmhost jdm start
Starting JDM
Starting jdm: Domain jdm defined from
/vm/vm/iapps/jdm//install/juniper/lxc/jdm/current/config/jdm.xml
Domain jdm started
```

request vmhost jdm login (In-Chassis Model)

Syntax	request vmhost jdm login
Release Information	Command introduced in Junos OS Release 19.1R1.
Description	Logs in to the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos Node Slicing.
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • show virtual-network-functions on page 172 • request server authenticate-peer-server on page 170

Sample Output

request vmhost jdm login

```
root@router> request vmhost jdm login
```

```
*****
* The Juniper Device Manager (JDM) must only be used for orchestrating the *
* Virtual Machines for Junos Node Slicing *
* *
* Host Linux Distro: Wind River Linux *
* JDM Version: jns-jdm-vmhost-19.1-20181003.dev.common.0.x86_64 *
* Free Disk Space on JDM's root-fs ("/"): 125081(MiB) *
*****
Last login: Thu Oct 4 15:26:30 2018 from 192.168.1.1
```

request vmhost jdm stop (In-Chassis Model)

Syntax	<code>request vmhost jdm stop</code>
Release Information	Command introduced in Junos OS Release 19.1R1.
Description	Stops the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos Node Slicing.
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none">• Generic Guidelines for Using JDM Server Commands on page 163• show virtual-network-functions on page 172• request server authenticate-peer-server on page 170

show vmhost status (In-Chassis Model)

Syntax	<pre>show vmhost status (re0 re1) <routing-engine (backup both local master other)> <invoke-on (all-routing-engines other-routing-engine)></pre>
Release Information	Command introduced in Junos OS Release 19.1R1.
Description	<p>Displays the VM host status. This command is applicable to in-chassis Junos Node Slicing. When the router is operating in the in-chassis Junos Node Slicing mode, the vJunos Resource Status is shown as Compact. When the router is not operating in the in-chassis Junos Node Slicing mode, the vJunos Resource Status is shown as Normal.</p>
Options	<p>re0—Displays Routing Engine 0 (re0) VM host status.</p> <p>re1—Displays Routing Engine 1 (re1) VM host status.</p> <p>routing-engine—Displays the VM host status of a specific routing engine. You have the following sub-options:</p> <ul style="list-style-type: none"> • backup • master • both • local • other <p>invoke-on—Displays the VM host status of the remote Routing Engines or all Routing Engines. You have the following options:</p> <ul style="list-style-type: none"> • all-routing-engines • other-routing-engine
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • show virtual-network-functions on page 172 • request server authenticate-peer-server on page 170

Sample Output

`show vmhost jdm status re0`

```
user@router> show vmhost status re0
```

```
bsys-re0:
```

```
-----
```

```
Compute cluster: rainier-re-cc
```

```
  Compute Node: rainier-re-cn, Online
```

```
vJunos Resource Status: Compact
```

show vmhost jdm status (In-Chassis Model)

Syntax `show vmhost jdm status`

Release Information Command introduced in Junos OS Release 19.1R1.

Description Displays the Juniper Device Manager (JDM) status. This command is applicable to in-chassis Junos Node Slicing.

Required Privilege Level View

Related Documentation

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [show virtual-network-functions on page 172](#)
- [request server authenticate-peer-server on page 170](#)

Sample Output

`show vmhost jdm status`

```
user@router> show vmhost jdm status
```

```
JDM Information
```

```
-----
Package   : jns-jdm-vmhost-19.1-B2.x86_64
Status    : Running
PID       : 3088
Free Space : 62967 (MiB)
```

show chassis network-slices

Syntax	<code>show chassis network-slices</code> <code><gnf <i>gnf-id</i>></code>
Release Information	Command introduced in Junos OS Release 17.2R1.
Description	Display Junos Node Slicing information for the guest network functions (GNFs) configured on the base system (BSYS). The gnf <i>gnf-id</i> option displays the information about a particular GNF.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show chassis fpc pic-status (Node Slicing) on page 115 • show chassis fpc (Node Slicing) on page 117 • show chassis adc (Node Slicing) on page 119 • show chassis network-slices fpcs on page 121
List of Sample Output	show chassis network-slices on page 113 show chassis network-slices gnf on page 113 show chassis network-slices gnf <gnf-id> on page 114
Output Fields	Table 9 on page 112 lists the output fields for the show chassis network-slices command. Output fields are listed in the approximate order in which they appear.

Table 9: show chassis network-slices Output Fields

Field Name	Field Description
GNF	GNF identifier for each partition.
Description	Description of the guest network function.
State	Status of the GNF. <ul style="list-style-type: none"> • Online—GNF online and running. • Offline—GNF is offline.
Uptime	Duration for which the GNFs have been up and running.
GNF ID	Shows the GNF ID.
GNF description	Shows the description of GNF.
FPCs assigned	Shows the FPC slot numbers assigned to the GNF.

Table 9: show chassis network-slices Output Fields (continued)

Field Name	Field Description
FPCs online	Shows the FPCs that are online.
BSYS	Shows the name of the BSYS.
BSYS sw version	Shows the Junos software version used in the BSYS.
GNF sw version	Shows the Junos software version used in the GNF.
BSYS master RE	Master Routing Engine slot.
GNF uptime	Duration for which the GNF has been up and running.
GNF Routing Engine Status:	Show the details of the Routing Engines in the slot 0 and 1. It includes the Current State, (master or backup), Routing Engine model, and GNF host name.

Sample Output

show chassis network-slices

```
user@router> show chassis network-slices
```

```
guest-network-functions:
```

GNF	Description	State	Uptime
1	gnf-a	Online	12 hours, 46 minutes, 11 seconds
2	gnf-b	Online	12 hours, 13 minutes, 57 seconds
3	gnf-c	Online	12 hours, 3 minutes, 55 seconds
4	gnf-d	Online	12 hours, 8 minutes, 20 seconds
5	gnf-e	Online	12 hours, 2 minutes, 46 seconds
6	gnf-f	Online	11 hours, 56 minutes, 29 seconds

show chassis network-slices gnf

```
user@router> show chassis network-slices gnf
```

```
GNF ID          1
GNF description NA
GNF state       Online
FPCs assigned   7 8
FPCs online     7 8
BSYS            router(mx960)
BSYS sw version 18.2-20180321_0948_bsys
GNF sw version  18.2-20180314_gnf
Chassis         mx960
BSYS master RE  0
GNF uptime      4 days, 23 hours, 55 minutes, 1 second
GNF Routing Engine Status:
Slot 0:
  Current state  Master
  Model         RE-GNF-2400x4
  GNF host name  gnf-host0
```

```

Slot 1:
  Current state  Backup
  Model         RE-GNF-2400x4
  GNF host name  gnf-host1
GNF ID         2
GNF description NA
GNF state      Online
FPCs assigned  NA
FPCs online    NA
BSYS           router(mx960)
BSYS sw version 18.2-20180321_0948_bsys
GNF sw version  18.2-20180216_gnf
Chassis        mx960
BSYS master RE  0
GNF uptime      4 days, 23 hours, 53 minutes, 54 seconds
GNF Routing Engine Status:
Slot 0:
  Current state  Master
  Model         RE-GNF-2400x4
  GNF host name  gnf-host2
Slot 1:
  Current state  Backup
  Model         RE-GNF-2400x4
  GNF host name  gnf-host3

```

show chassis network-slices gnf <gnf-id>

```

user@router> show chassis network-slices gnf 2

GNF ID         2
GNF description NA
GNF state      Online
FPCs assigned  NA
FPCs online    NA
BSYS           router(mx960)
BSYS sw version 18.2-20180321_0948_bsys
GNF sw version  18.2-20180216_gnf
Chassis        mx960
BSYS master RE  0
GNF uptime      4 days, 23 hours, 53 minutes, 54 seconds
GNF Routing Engine Status:
Slot 0:
  Current state  Master
  Model         RE-GNF-2400x4
  GNF host name  gnf-host2
Slot 1:
  Current state  Backup
  Model         RE-GNF-2400x4
  GNF host name  gnf-host3

```

show chassis fpc pic-status (Node Slicing)

Syntax	show chassis fpc pic-status
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 12.3 for MX2020 Universal Routing Platforms. Command introduced in Junos OS Release 12.3 for MX2010 Universal Routing Platforms.
Description	Display the status of the physical interface cards (PICs) of each Flexible PIC Concentrator (FPC) assigned to different guest network functions (GNFs).
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show chassis network-slices on page 112 • show chassis fpc (Node Slicing) on page 117 • show chassis adc (Node Slicing) on page 119 • show chassis network-slices fpcs on page 121

Sample Output

```
user@router> show chassis fpc pic-status
```

```

Slot 0  Online      MPC5E 3D 24XGE+6XLGE      GNF 3
      PIC 0  Online      12X10GE SFPP OTN
      PIC 1  Offline     12X10GE SFPP OTN
      PIC 2  Offline     3X40GE QSFPP
      PIC 3  Online      3X40GE QSFPP
Slot 1  Online      MPC9E 3D      GNF 2
      PIC 1  Online      MRATE-12xQSFPP-XGE-XLGE-CGE
Slot 2  Online      MPC5E 3D Q 2CGE+4XGE      GNF 3
      PIC 0  Online      2X10GE SFPP OTN
      PIC 1  Online      1X100GE CFP2 OTN
      PIC 2  Online      2X10GE SFPP OTN
      PIC 3  Online      1X100GE CFP2 OTN
Slot 3  Online      MPCE Type 2 3D EQ      GNF 6
Slot 4  Online      MPC6E 3D      GNF 6
      PIC 0  Online      24X10GE SFPP
      PIC 1  Online      2X100GE CFP2 OTN
Slot 5  Online      MPC9E 3D      GNF 4
      PIC 0  Online      MRATE-12xQSFPP-XGE-XLGE-CGE
Slot 6  Online      MPC7E 3D MRATE-12xQSFPP-XGE-XLGE-CGE      GNF 1
      PIC 0  Online      MRATE-6xQSFPP-XGE-XLGE-CGE
      PIC 1  Online      MRATE-6xQSFPP-XGE-XLGE-CGE
Slot 7  Online      MPC5E 3D 2CGE+4XGE      GNF 5
      PIC 0  Online      2X10GE SFPP OTN
      PIC 1  Online      1X100GE CFP2 OTN
      PIC 2  Online      2X10GE SFPP OTN
      PIC 3  Online      1X100GE CFP2 OTN

```

Slot 8	Online	MPC6E 3D	GNF 5
PIC 0	Online	24X10GE SFPP OTN	
Slot 9	Online	MPC6E 3D	GNF 5
PIC 0	Online	24X10GE SFPP	
PIC 1	Online	4X100GE CXP	

show chassis fpc (Node Slicing)

Syntax `show chassis fpc`

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.3 for MX2020 Universal Routing Platforms.
Command introduced in Junos OS Release 12.3 for MX2010 Universal Routing Platforms.

Description Display information about Flexible PIC Concentrators (fpcs) assigned to different guest network functions (GNFs).

Required Privilege Level view

Related Documentation

- [show chassis network-slices on page 112](#)
- [show chassis fpc pic-status \(Node Slicing\) on page 115](#)
- [show chassis adc \(Node Slicing\) on page 119](#)
- [show chassis network-slices fpcs on page 121](#)

Output Fields [Table 10 on page 117](#) lists the output fields for the `show chassis fpc` command. Output fields are listed in the approximate order in which they appear.

Table 10: show chassis fpc Output Fields

Field Name	Field Description
Slot or Slot State	Slot number and state. The state can be one of the following conditions: <ul style="list-style-type: none"> • Dead—Held in reset because of errors. • Diag—Slot is being ignored while the FPC is running diagnostics. • Dormant—Held in reset. • Empty—No FPC is present. • Online—FPC is online and running. • Present—FPC is detected by the chassis daemon but either is not supported by the current version of Junos OS or is inserted in the wrong slot. The output also states either Hardware Not Supported or Hardware Not In Right Slot. The FPC is coming up but not yet online. • Probed—Probe is complete; awaiting restart of the Packet Forwarding Engine. • Probe-wait—Waiting to be probed.
Temp (C) or Temperature	Temperature of the air passing by the FPC, in degrees Celsius or in both Celsius and Fahrenheit.
Total CPU Utilization (%)	Total percentage of CPU being used by the FPC's processor.

Table 10: show chassis fpc Output Fields (continued)

Field Name	Field Description
Interrupt CPU Utilization (%)	Of the total CPU being used by the FPC's processor, the percentage being used for interrupts.
1 min CPU Utilization (%)	Information about the Routing Engine's CPU utilization in the past 1 minute.
5 min CPU Utilization (%)	Information about the Routing Engine's CPU utilization in the past 5 minutes.
15 min CPU Utilization (%)	Information about the Routing Engine's CPU utilization in the past 15 minutes.
Memory DRAM (MB)	Total DRAM, in megabytes, available to the FPC's processor.
Heap Utilization (%)	Percentage of heap space (dynamic memory) being used by the FPC's processor. If this number exceeds 80 percent, there may be a software problem (memory leak). NOTE: On MX Series routers and EX Series switches in a broadband edge environment, heap utilization levels higher than 70 percent can affect unified ISSU, router stability, or scaling capability.
Buffer Utilization (%)	Percentage of buffer space being used by the FPC's processor for buffering internal messages.
GNF	GNF identifier associated with each FPC. (pic-status output only) GNF identifier associated with each PIC.

Sample Output

```
user@router> show chassis fpc
```

Slot	State	Temp (C)	CPU Utilization (%)		CPU Utilization(%)			Memory Utilization (%)		
			Total	Interrupt	1min	5min	15min	DRAM(MB)	Heap	Buffer
0	Online	45	12	0	12	12	12	3584	6	25
1	Online	57	22	0	20	20	20	3136	16	22
2	Online	50	19	0	17	17	16	3584	6	25
3	Online	28	10	0	11	11	11	2048	10	20
4	Online	42	20	0	20	19	19	3584	8	25
5	Online	58	22	0	21	20	20	3136	16	22
6	Online	49	17	0	15	16	16	3136	13	20
7	Online	44	11	0	10	10	10	3584	6	25
8	Online	40	19	0	18	18	18	3584	8	25
9	Online	44	19	0	20	20	20	3584	8	25

show chassis adc (Node Slicing)

Syntax	show chassis adc
Release Information	Command introduced in Junos OS Release 12.3 for MX2020 Universal Routing Platforms. Command introduced in Junos OS Release 12.3 for MX2010 Universal Routing Platforms. Command introduced in Junos OS Release 17.2 for MX2008 Universal Routing Platforms.
Description	Display chassis information about the adapter cards (ADCs).
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show chassis network-slices on page 112 • show chassis fpc pic-status (Node Slicing) on page 115 • show chassis fpc (Node Slicing) on page 117 • show chassis network-slices fpcs on page 121
List of Sample Output	show chassis adc (Node Slicing) on page 119
Output Fields	Table 11 on page 119 lists the output fields for the show chassis adc command. Output fields are listed in the approximate order in which they appear.

Table 11: show chassis adc Output Fields

Field Name	Field Description
Slot	Slot number.
State	Status of the adapter card. <ul style="list-style-type: none"> • Online—The adapter card is online and running. • Offline—Adapter card is powered down.
Uptime	How long the Routing Engine has been connected to the adapter card and, therefore, how long the adapter card has been up and running.
GNF (Node slicing)	GNF identifier for each ADC.

Sample Output

show chassis adc (Node Slicing)

```
user@router> show chassis adc
```

Slot	State	Uptime	GNF
0	Online	12 hours, 57 minutes, 46 seconds	3
1	Empty	--- Native line card ---	2
2	Online	12 hours, 57 minutes, 18 seconds	3
3	Online	11 minutes, 23 seconds	6
4	Empty	--- Native line card ---	6
5	Empty	--- Native line card ---	4
6	Online	13 hours, 38 minutes, 58 seconds	1
7	Online	13 hours, 3 minutes, 40 seconds	5
8	Empty	--- Native line card ---	5
9	Empty	--- Native line card ---	5

show chassis network-slices fpcs

Syntax	show chassis network-slices fpcs
Release Information	Command introduced in Junos OS Release 17.2R1.
Description	Display information about the FPCs associated with different guest network functions (GNFs).
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• show chassis network-slices on page 112• show chassis fpc pic-status (Node Slicing) on page 115• show chassis fpc (Node Slicing) on page 117• show chassis adc (Node Slicing) on page 119
Output Fields	Table 12 on page 121 lists the output fields for the show chassis network-slices fpcs command. Output fields are listed in the approximate order in which they appear.

Table 12: show chassis network-slices fpcs Output Fields

Field Name	Field Description
GNF	GNF ID.
FPCs	Slot numbers of FPCs associated with the GNF ID.

Sample Output

```
user@router> show chassis network-slices fpcs
guest-network-functions:
GNF    FPCs
1      6
2      1
3      0 2
4      5
5      7 8 9
6      3 4
```

show system anomalies gnf-id (BSYS)

Syntax	show system anomalies gnf-id <i>id</i> (all-anomalies config fru system)
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display incompatibilities between the software version running on the base system (BSYS) and the software running on a specific guest network function (GNF).
Options	<p>gnf-id <i>id</i>—Specify the GNF ID for which you want to view the software incompatibilities.</p> <p>all-anomalies—Display the multiversion software incompatibilities from all categories—system, configuration and FRU.</p> <p>config—Display feature-level incompatibilities between software versions.</p> <p>fru—Display the FRU-level incompatibilities between software versions. This can be an incompatibility pertaining to the support for a specific FRU.</p> <p>system—Display the system-level incompatibilities between software versions. These include IPC incompatibility, CLI or SNMP incompatibility.</p>
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • show system anomalies (GNF) on page 153 • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170
Output Fields	Table 13 on page 122 lists the output fields for the show system anomalies gnf-id command. Output fields are listed in the approximate order in which they appear.

Table 13: show system anomalies gnf-id Output Fields

Field Name	Field Description
Anomaly Type	Shows the software incompatibility type. The following are the possible values: <ul style="list-style-type: none"> • SYS—Indicates system-specific incompatibilities. • FRU—Indicates FRU-specific incompatibilities. • CONFIG—Indicates feature-specific incompatibilities.
Anomaly ID	Shows the incompatibility ID.

Table 13: show system anomalies gnf-id Output Fields (continued)

Field Name	Field Description
Default Action	Shows the default actions associated with incompatibilities. The following are the possible values: <ul style="list-style-type: none"> WARN—Indicates the presence of a minor incompatibility. It causes a warning. ABORT—Indicates the presence of a major incompatibility. It causes an abort action.
Class	Indicates of the incompatibility is present in GNF, or BSYS, or both.
Message	Incompatibility description.
FRU ID	Field-replaceable unit (FRU) ID. Applicable in the case of FRU-specific incompatibilities.
Firmware	Firmware of the FRU.

Sample Output

```
user@router> show system anomalies gnf-id 1
```

Anomaly Type	Anomaly ID FRU ID	Default Action Firmware	Class	Message
SYS incompatibility 1	100	WARN	GNF Present	system
SYS incompatibility 2	103	WARN	BOTH Present	system
SYS incompatibility 3	200	WARN	BSYS Present	system

CHAPTER 6

Operational Commands for GNF

- `show chassis gnf`
- `show chassis gnf`
- `show chassis hardware (GNF)`
- `show chassis fpc (GNF)`
- `show chassis fpc pic-status (GNF)`
- `show chassis adc (GNF)`
- `show interfaces (Abstracted Fabric)`
- `show system anomalies (GNF)`

show chassis gnf

Syntax `show chassis gnf`

Release Information Command introduced in Junos OS Release 18.2R1.

Description Display information about the guest network function (GNF) you logged in.

Required Privilege Level view

Related Documentation

- [show chassis hardware \(GNF\) on page 130](#)
- [show chassis fpc pic-status \(GNF\) on page 135](#)
- [show chassis adc \(GNF\) on page 136](#)
- [show chassis network-slices on page 112](#)

List of Sample Output [show chassis gnf on page 127](#)

Output Fields [Table 14 on page 126](#) lists the output fields for the **show chassis gnf** command. Output fields are listed in the approximate order in which they appear.

Table 14: show chassis gnf Output Fields

Field Name	Field Description
GNF ID	Shows the GNF ID.
GNF State	Status of the GNF. <ul style="list-style-type: none"> • Online—GNF online and running. • Offline—GNF is offline.
GNF description	Description of the guest network function.
FPCs assigned	The FPC slot numbers assigned to the GNF.
FPCs online	The FPCs that are online.
BSYS	Name of the BSYS.
BSYS sw version	The Junos software version used in the BSYS.
GNF sw version	The Junos software version used in the GNF.
BSYS master RE	Master Routing Engine slot.
GNF uptime	Duration for which the GNF has been up and running.

Table 14: show chassis gnf Output Fields (continued)

Field Name	Field Description
GNF Routing Engine Status:	The details of the Routing Engines in the slot 0 and 1. The details include the Current State, (master or backup), Routing Engine model, and GNF host name.

Sample Output

show chassis gnf

```
user@router> show chassis gnf
```

```
GNF ID          1
GNF description NA
GNF state       Online
FPCs assigned   8 9
FPCs online     8 9
BSYS            router(mx960)
BSYS sw version 18.2-20180321_0948_bsys
GNF sw version  18.2-20180314_1035_gnf
Chassis         mx960
BSYS master RE  0
GNF uptime      54 minutes, 37 seconds
GNF Routing Engine Status:
Slot 0:
  Current state  Master
  Model         RE-GNF-2100x4
  GNF host name  gnf-host0
Slot 1:
  Current state  Backup
  Model         RE-GNF-2100x4
  GNF host name  gnf-host1
```

show chassis gnf

Syntax `show chassis gnf`

Release Information Command introduced in Junos OS Release 18.2R1.

Description Display information about the guest network function (GNF) you logged in.

Required Privilege Level view

Related Documentation

- [show chassis hardware \(GNF\) on page 130](#)
- [show chassis fpc pic-status \(GNF\) on page 135](#)
- [show chassis adc \(GNF\) on page 136](#)
- [show chassis network-slices on page 112](#)

List of Sample Output [show chassis gnf on page 129](#)

Output Fields [Table 14 on page 126](#) lists the output fields for the **show chassis gnf** command. Output fields are listed in the approximate order in which they appear.

Table 15: show chassis gnf Output Fields

Field Name	Field Description
GNF ID	Shows the GNF ID.
GNF State	Status of the GNF. <ul style="list-style-type: none"> • Online—GNF online and running. • Offline—GNF is offline.
GNF description	Description of the guest network function.
FPCs assigned	The FPC slot numbers assigned to the GNF.
FPCs online	The FPCs that are online.
BSYS	Name of the BSYS.
BSYS sw version	The Junos software version used in the BSYS.
GNF sw version	The Junos software version used in the GNF.
BSYS master RE	Master Routing Engine slot.
GNF uptime	Duration for which the GNF has been up and running.

Table 15: show chassis gnf Output Fields (continued)

Field Name	Field Description
GNF Routing Engine Status:	The details of the Routing Engines in the slot 0 and 1. The details include the Current State, (master or backup), Routing Engine model, and GNF host name.

Sample Output

show chassis gnf

```
user@router> show chassis gnf
```

```
GNF ID          1
GNF description NA
GNF state       Online
FPCs assigned   8 9
FPCs online     8 9
BSYS            router(mx960)
BSYS sw version 18.2-20180321_0948_bsys
GNF sw version  18.2-20180314_1035_gnf
Chassis         mx960
BSYS master RE  0
GNF uptime      54 minutes, 37 seconds
GNF Routing Engine Status:
Slot 0:
  Current state  Master
  Model         RE-GNF-2100x4
  GNF host name  gnf-host0
Slot 1:
  Current state  Backup
  Model         RE-GNF-2100x4
  GNF host name  gnf-host1
```

show chassis hardware (GNF)

Syntax	show chassis hardware
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 12.3 for MX2010 and MX2020 Universal Routing Platforms.
Description	Display a list of all hardware components of the chassis, including the hardware version level and serial number, the GNF Routing Engine details, and the FPCs assigned to the GNF.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show chassis fpc (GNF) on page 133 • show chassis fpc pic-status (GNF) on page 135 • show chassis adc (GNF) on page 136
Output Fields	Table 16 on page 130 lists the output fields for the show chassis hardware command. Output fields are listed in the approximate order in which they appear.

Table 16: show chassis hardware Output Fields

Field Name	Field Description
Item	Chassis component: <ul style="list-style-type: none"> • Information about the backplane, Routing Engine, Power Entry Modules (PEMs), and fan trays. Also displays information about Flexible PIC Concentrators (FPCs) and associated Physical Interface Cards (PICs), Modular Port Concentrators (MPCs) and associated Modular Interface Cards (MICs), or Dense Port Concentrators (DPCs).
Version	Revision level of the chassis component.
Part number	Part number of the chassis component.
Serial number	Serial number of the chassis component. The serial number of the backplane is also the serial number of the router chassis. Use this serial number when you need to contact Juniper Networks Customer Support about the router or switch chassis.
Description	Brief description of the hardware item.

Sample Output

```
user@router> show chassis hardware
```

```
bsys-re0:
```

```
-----
```

```
Hardware inventory:
```

Item	Version	Part number	Serial number	Description
Chassis			JN11C9CDDAFK	MX2010
Midplane	REV 35	750-044636	ABAB9184	Lower Backplane
Midplane 1	REV 02	711-044557	ABAB9048	Upper Backplane
PMP	REV 04	711-032426	ACAJ2622	Power Midplane
FPM Board	REV 09	760-044634	ABCF2618	Front Panel Display
PSM 0	REV 01	740-050037	1EDB3130084	DC 52V Power Supply
Module				
PSM 1	REV 01	740-050037	1EDB313001Z	DC 52V Power Supply
Module				
PSM 2	REV 01	740-050037	1EDB321018D	DC 52V Power Supply
Module				
PSM 3	REV 01	740-050037	1EDB32101AZ	DC 52V Power Supply
Module				
PSM 4	REV 01	740-050037	1EDB32202C2	DC 52V Power Supply
Module				
PSM 5	REV 01	740-050037	1EDB32100TC	DC 52V Power Supply
Module				
PSM 6	REV 01	740-050037	1EDB3210166	DC 52V Power Supply
Module				
PSM 7	REV 01	740-050037	1EDB3210165	DC 52V Power Supply
Module				
PSM 8	REV 01	740-050037	1EDB3210163	DC 52V Power Supply
Module				
PDM 0	REV 03	740-045234	1EGA3170177	DC Power Dist Module
Routing Engine 0	REV 08	750-055814	CAFV5537	RE-S-2X00x8
CB 0	REV 08	750-055087	CAFN3426	MX2K Enhanced SCB
Xcvr 0	REV 01	740-031980	ALM0HC7	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	123363A00418	SFP+-10G-SR
CB 1	REV 08	750-055087	CAFN3423	MX2K Enhanced SCB
SPMB 0	REV 05	711-041855	CAEZ5998	PMB Board
SPMB 1	REV 05	711-041855	CAEZ5993	PMB Board
SFB 0	REV 06	711-044466	ABCD6742	Switch Fabric Board
SFB 1	REV 06	711-044466	ABCG5627	Switch Fabric Board
SFB 2	REV 06	711-044466	ABCG5659	Switch Fabric Board
SFB 3	REV 06	711-044466	ABCG5653	Switch Fabric Board
SFB 4	REV 06	711-044466	ABCG5611	Switch Fabric Board
SFB 5	REV 06	711-044466	ABCG5635	Switch Fabric Board
SFB 6	REV 06	711-044466	ABCG5638	Switch Fabric Board
SFB 7	REV 06	711-044466	ABCG3650	Switch Fabric Board
FPC 8	REV 68	750-044130	ABCY5967	MPC6E 3D
CPU	REV 12	711-045719	ABCY9696	RMPC PMB
Fan Tray 0	REV 06	760-046960	ACAY0428	172mm FanTray - 6 Fans
Fan Tray 1	REV 06	760-046960	ACAY0800	172mm FanTray - 6 Fans
Fan Tray 2	REV 06	760-046960	ACAY0797	172mm FanTray - 6 Fans
Fan Tray 3	REV 06	760-046960	ACAY1047	172mm FanTray - 6 Fans

```
gnf2-re0:
```

```
-----
```

Chassis	GN59081553B0	MX2010-GNF	<<<
Routing Engine 0		RE-GNF-1700x4	

show chassis fpc (GNF)

Syntax	show chassis fpc
Release Information	<p>Command introduced before Junos OS Release 7.4.</p> <p>Command introduced in Junos OS Release 12.3 for MX2020 Universal Routing Platforms.</p> <p>Command introduced in Junos OS Release 12.3 for MX2010 Universal Routing Platforms.</p>
Description	Display information about the Flexible PIC Concentrators (fpcs) assigned to the guest network function (GNF).
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show chassis hardware (GNF) on page 130 • show chassis fpc pic-status (GNF) on page 135 • show chassis adc (GNF) on page 136
Output Fields	Table 17 on page 133 lists the output fields for the show chassis fpc command. Output fields are listed in the approximate order in which they appear.

Table 17: show chassis fpc Output Fields

Field Name	Field Description
Slot or Slot State	<p>Slot number and state. The state can be one of the following conditions:</p> <ul style="list-style-type: none"> • Dead—Held in reset because of errors. • Diag—Slot is being ignored while the FPC is running diagnostics. • Dormant—Held in reset. • Empty—No FPC is present. • Online—FPC is online and running. • Present—FPC is detected by the chassis daemon but either is not supported by the current version of Junos OS or is inserted in the wrong slot. The output also states either Hardware Not Supported or Hardware Not In Right Slot. The FPC is coming up but not yet online. • Probed—Probe is complete; awaiting restart of the Packet Forwarding Engine. • Probe-wait—Waiting to be probed.
Temp (C) or Temperature	Temperature of the air passing by the FPC, in degrees Celsius or in both Celsius and Fahrenheit.
Total CPU Utilization (%)	Total percentage of CPU being used by the FPC's processor.
Interrupt CPU Utilization (%)	Of the total CPU being used by the FPC's processor, the percentage being used for interrupts.

Table 17: show chassis fpc Output Fields (continued)

Field Name	Field Description
1 min CPU utilization (%)	Information about the Routing Engine's CPU utilization in the past 1 minute.
5 min CPU utilization (%)	Information about the Routing Engine's CPU utilization in the past 5 minutes.
15 min CPU utilization (%)	Information about the Routing Engine's CPU utilization in the past 15 minutes.
Memory DRAM (MB)	Total DRAM, in megabytes, available to the FPC's processor.
Heap Utilization (%)	<p>Percentage of heap space (dynamic memory) being used by the FPC's processor. If this number exceeds 80 percent, there might be a software problem (memory leak).</p> <p>NOTE: On MX Series routers and EX Series switches in a broadband edge environment, heap utilization levels higher than 70 percent can affect unified ISSU, router stability, or scaling capability.</p>
Buffer Utilization (%)	Percentage of buffer space being used by the FPC's processor for buffering internal messages.
GNF	<p>GNF identifier associated with each FPC.</p> <p>(pic-status output only) GNF identifier associated with each PIC.</p>

Sample Output

```
user@router>show chassis fpc
```

Slot	State	Temp (C)	CPU Utilization(%) Total	CPU Utilization(%) Interrupt	CPU Utilization(%) 1min	CPU Utilization(%) 5min	CPU Utilization(%) 15min	Memory Utilization(%) DRAM(MB)	Memory Utilization(%) Heap	Memory Utilization(%) Buffer	GNF
4	Online	42	20	0	19	19	19	3584	8	25	2
6	Online	46	12	0	11	11	11	3136	8	19	2

show chassis fpc pic-status (GNF)

Syntax	show chassis fpc pic-status
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 12.3 for MX2020 Universal Routing Platforms. Command introduced in Junos OS Release 12.3 for MX2010 Universal Routing Platforms.
Description	Display the status of the physical interface cards (PICs) of each Flexible PIC Concentrator (FPC) assigned to the guest network function (GNF).
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show chassis hardware (GNF) on page 130 • show chassis fpc (GNF) on page 133 • show chassis adc (GNF) on page 136

Sample Output

```
user@router> show chassis fpc pic-status
```

```

Slot 4  Online      MPC6E 3D                GNF 2
      PIC 0  Online      24X10GE SFPP
      PIC 1  Online      4X100GE CXP
Slot 6  Online      MPC7E 3D MRATE-12xQSFPP-XGE-XLGE-CGE  GNF 2
      PIC 0  Online      MRATE-6xQSFPP-XGE-XLGE-CGE
      PIC 1  Online      MRATE-6xQSFPP-XGE-XLGE-CGE

```

show chassis adc (GNF)

Syntax	show chassis adc
Release Information	Command introduced in Junos OS Release 12.3 for MX2020 Universal Routing Platforms. Command introduced in Junos OS Release 12.3 for MX2010 Universal Routing Platforms. Command introduced in Junos OS Release 17.2 for MX2008 Universal Routing Platforms.
Description	Display chassis information about the adapter cards (ADCs) assigned to the guest network function (GNF).
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • show chassis hardware (GNF) on page 130 • show chassis fpc (GNF) on page 133 • show chassis fpc pic-status (GNF) on page 135
Output Fields	Table 18 on page 136 lists the output fields for the show chassis adc command. The output fields are listed in the approximate order in which they appear.

Table 18: show chassis adc Output Fields

Field Name	Field Description
Slot	Slot number.
State	Status of the adapter card. <ul style="list-style-type: none"> • Online—The adapter card is online and running. • Offline—Adapter card is powered down.
Uptime	How long the Routing Engine has been connected to the adapter card and, therefore, how long the adapter card has been up and running.
GNF (Node slicing)	GNF identifier for each ADC.

Sample Output

```
user@router> show chassis adc
```

Slot	State	Uptime	GNF
4	Empty	--- Native line card ---	2
6	Online	50 minutes, 2 seconds	2

show interfaces (Abstracted Fabric)

Syntax `show interfaces af-interface-name`
`<brief | detail | extensive | terse>`
`<descriptions>`
`<media>`
`<snmp-index snmp-index>`
`<statistics>`

Release Information Command introduced before Junos OS Release 17.4R1.

Description Display status information for the specified Abstracted Fabric (af) interface.



NOTE: Starting in Junos OS Release 18.3R1, the `show interfaces af-interface-name` output provides transmit statistics of each Packet Forwarding Engine peer list on a given af interface, in addition to the physical interface statistics.

Options `brief | detail | extensive | terse`—(Optional) Display the specified level of output.

`descriptions`—(Optional) Display interface description strings.

`media`—(Optional) Display media-specific information about network interfaces.

`snmp-index snmp-index`—(Optional) Display information for the specified SNMP index of the interface.

`statistics`—(Optional) Display static interface statistics.

Required Privilege Level view

Related Documentation

- [Configuring Abstracted Fabric Interfaces Between a Pair of GNFs on page 59](#)

List of Sample Output

- [show interfaces af-interface-name on page 142](#)
- [show interfaces af-interface-name detail on page 143](#)
- [show interfaces af-interface-name extensive on page 145](#)
- [show interfaces af-interface-name statistics on page 147](#)
- [show interfaces af-interface-name on page 148](#)
- [show interfaces af-interface-name brief on page 149](#)
- [show interfaces af-interface-name extensive on page 149](#)
- [show interfaces af-interface-name detail on page 151](#)

Output Fields Table 19 on page 139 describes the output fields for the **show interfaces** (Abstracted Fabric) command. Output fields are listed in the approximate order in which they appear.

Table 19: show interfaces (Abstracted Fabric) Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name and status of the physical interface.	All levels
Interface index	Index number of the physical interface, which reflects its initialization sequence.	detail extensive none
SNMP ifIndex	SNMP index number for the physical interface.	detail extensive none
Generation	Unique number for use by Juniper Networks technical support only.	detail extensive
Type	Interface type.	
Link-level type	Encapsulation being used on the physical interface.	All levels
MTU	Maximum transmission unit size on the physical interface.	All levels
Speed	Speed at which the interface is running.	All levels
Device flags	Information about the physical device. Possible values are described in the “Device Flags” section under <i>Common Output Fields Description</i> .	All levels
Interface flags	Information about the interface. Possible values are described in the “Interface Flags” section under <i>Common Output Fields Description</i> .	All levels
Link type	Link type. For example, Full-Duplex.	All levels
Damping	Damping information.	detail extensive
Alternate link address	Address of the alternative link.	detail extensive
Hold-times	Current interface hold-time up and hold-time down, in milliseconds (ms).	detail extensive
Current address	Configured MAC address.	detail extensive none
Hardware address	Hardware MAC address.	detail extensive none
Last flapped	Date, time, and how long ago the interface went from down to up. The format is Last flapped: year-month-day hour:minute:second:timezone (hour:minute:second ago) . For example, Last flapped: 2002-04-26 10:52:40 PDT (04:33:20 ago) .	detail extensive none
Statistics last cleared	Time when the statistics for the interface were last set to zero.	detail extensive

Table 19: show interfaces (Abstracted Fabric) Output Fields (continued)

Field Name	Field Description	Level of Output
Traffic statistics	Number and rate of bytes and packets received and transmitted on the physical interface. <ul style="list-style-type: none"> • Input bytes—Number of bytes received on the interface. • Output bytes—Number of bytes transmitted on the interface. • Input packets—Number of packets received on the interface. • Output packets—Number of packets transmitted on the interface. 	detail extensive
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the interface if IPv6 statistics tracking is enabled.	extensive
Input errors	Input errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: <ul style="list-style-type: none"> • Errors—Sum of the incoming frame aborts and FCS errors. • Drops—Number of packets dropped by the input queue. • Framing errors—Number of packets received with an invalid frame checksum (FCS). • Runts—Number of frames received that are smaller than the runt threshold. • Policed discards—Number of frames that the incoming packet match code discarded because they were not recognized or not of interest. Usually, this field reports protocols that Junos OS does not handle. • Resource errors—Sum of transmit drops. 	extensive
Output errors	Output errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: <ul style="list-style-type: none"> • Carrier transitions—Number of times the interface has gone from down to up. • Errors—Sum of the outgoing frame aborts and FCS errors. • Drops—Number of packets dropped by the output queue. <p>NOTE:</p> <ul style="list-style-type: none"> • MTU errors—Number of packets whose size exceeded the MTU of the interface. • Resource errors—Sum of transmit drops. 	extensive
Bandwidth	Shows the interface bandwidth.	detail extensive none
Peer GNF id	The GNF peer connected using the Abstracted Fabric interface.	detail extensive none
Peer GNF Forwarding element(FE) view	Shows forwarding element (FE) number and the FPC slot, FE bandwidth, and FE status (up/down).	detail extensive none
Transmit Packets	Shows the packets transmitted.	
Transmit Bytes	Shows the bytes transmitted.	
Residual Transmit Statistics	Displays the historical transmit statistics for the peer af interface from the FPC events (FPC offline/online/removal/power off on the local or remote side).	

Table 19: show interfaces (Abstracted Fabric) Output Fields (continued)

Field Name	Field Description	Level of Output
Fabric Queue Statistics	Displays the statistics of the transmit traffic on the fabric queues (high and low queues) for each peer PFE on the af interface.	
Residual Queue Statistics	Displays the historical fabric queue statistics for the peer af interface from the FPC events (FPC offline/online/removal/power off on the local or remote side).	
Logical Interface		
Logical interface	Name of the logical interface.	All levels
Index	Index number of the logical interface, which reflects its initialization sequence.	detail extensive none
SNMP ifIndex	SNMP interface index number for the logical interface.	detail extensive none
Generation	Unique number for use by Juniper Networks technical support only.	detail extensive
Flags	Information about the logical interface. Possible values are described in the “Logical Interface Flags” section under <i>Common Output Fields Description</i> .	All levels
Encapsulation	Encapsulation on the logical interface.	All levels
Protocol	Protocol family. Possible values are described in the “Protocol Field” section under <i>Common Output Fields Description</i> .	detail extensive none
MTU	Maximum transmission unit size on the logical interface.	detail extensive none
Traffic statistics	<p>Number and rate of bytes and packets received and transmitted on the specified interface set.</p> <ul style="list-style-type: none"> Input bytes, Output bytes—Number of bytes received and transmitted on the interface set. Input packets, Output packets—Number of packets received and transmitted on the interface set. 	detail extensive
Transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the logical interface if IPv6 statistics tracking is enabled.	extensive
Local statistics	Number and rate of bytes and packets destined to the router.	extensive
Generation	Unique number for use by Juniper Networks technical support only.	detail extensive
Route Table	Route table in which the logical interface address is located. For example, 0 refers to the routing table inet.0.	detail extensive none
Flags	Information about protocol family flags. Possible values are described in the “Family Flags” section under <i>Common Output Fields Description</i> .	detail extensive
Addresses, Flags	Information about the address flags. Possible values are described in the “Addresses Flags” section under <i>Common Output Fields Description</i> .	detail extensive none

Table 19: show interfaces (Abstracted Fabric) Output Fields (continued)

Field Name	Field Description	Level of Output
protocol-family	Protocol family configured on the logical interface. If the protocol is inet , the IP address of the interface is also displayed.	brief
Flags	Information about the address flag. Possible values are described in the "Addresses Flags" section under <i>Common Output Fields Description</i> .	detail extensive none
Destination	IP address of the remote side of the connection.	detail extensive none
Local	IP address of the logical interface.	detail extensive none
Broadcast	Broadcast address of the logical interface.	detail extensive none
Generation	Unique number for use by Juniper Networks technical support only.	detail extensive

Sample Output

show interfaces af-interface-name

This output is applicable to Junos OS 18.3R1 and later releases.

```
user@router> show interfaces af9
```

```
Physical interface: af9, Enabled, Physical link is Up
  Interface index: 209, SNMP ifIndex: 527
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 370000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 00:90:69:2b:00:4c, Hardware address: 00:90:69:2b:00:4c
  Last flapped   : 2018-09-12 01:44:01 PDT (00:01:02 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Bandwidth      : 370 Gbps
  Peer GNF id    : 9
  Peer GNF Forwarding element(FE) view :
  FPC slot:FE num  FE Bandwidth(Gbps) Status    Transmit Packets    Transmit
  Bytes
      6:0              130          Up              0
      0
      12:0             120          Up              0
      0
      12:1             120          Up              0
      0

  Residual Transmit Statistics :
  Packets :              0 Bytes :              0

  Fabric Queue Statistics :
  FPC slot:FE num  High priority(pkts)    Low priority(pkts)
      6:0              0              0
      12:0             0              0
      12:1             0              0
```

```

FPC slot:FE num    High priority(bytes)    Low priority(bytes)
6:0                0                      0
12:0               0                      0
12:1               0                      0
Residual Queue Statistics :
  High priority(pkts)    Low priority(pkts)
                        0                      0
  High priority(bytes)    Low priority(bytes)
                        0                      0

Logical interface af9.0 (Index 332) (SNMP ifIndex 528)
Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
Input packets : 0
Output packets: 13
Protocol inet, MTU: 1500

```

show interfaces af-interface-name detail

This output is applicable to Junos OS 18.3R1 and later releases.

```
user@router> show interfaces af2 detail
```

```

Physical interface: af2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 674, Generation: 349
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Clocking: Unspecified,
Speed: 130000Mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Physical info   : Unspecified
  Hold-times     : Up 0 ms, Down 0 ms
  Damping        : half-life: 0 sec, max-suppress: 0 sec, reuse: 0, suppress: 0,
state: unsuppressed
  Current address: 00:90:69:fd:85:a4, Hardware address: 00:90:69:fd:85:a4
  Alternate link address: Unspecified
  Last flapped   : 2018-11-01 20:44:26 PDT (6d 02:57 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes :      23381827      280 bps
    Output bytes :    652664393      416 bps
    Input packets:      341618        0 pps
    Output packets:    5986312        0 pps
  IPv6 transit statistics:
    Input bytes :      0
    Output bytes :      0
    Input packets:      0
    Output packets:      0
  Bandwidth      : 130 Gbps
  Peer GNF id    : 2
  Peer GNF Forwarding element(FE) view :
FPC slot:FE num FE Bandwidth(Gbps) Status Transmit Packets Transmit
Bytes
      2:0                130      Up      5656951
622264610

  Residual Transmit Statistics :
  Packets :      0 Bytes :      0

  Fabric Queue Statistics :

```

```

FPC slot:FE num    High priority(pkts)    Low priority(pkts)
2:0                5656951                0
FPC slot:FE num    High priority(bytes)    Low priority(bytes)
2:0                622264610              0
Residual Queue Statistics :
  High priority(pkts)    Low priority(pkts)
                        0                0
  High priority(bytes)    Low priority(bytes)
                        0                0

Logical interface af2.0 (Index 334) (SNMP ifIndex 647) (Generation 234)
Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ] Encapsulation: ENET2
Traffic statistics:
  Input bytes :          23382038
  Output bytes :        650688227
  Input packets:         341621
  Output packets:        5986312
Local statistics:
  Input bytes :          23381827
  Output bytes :        28423617
  Input packets:         341618
  Output packets:        329361
Transit statistics:
  Input bytes :           211                0 bps
  Output bytes :        622264610            0 bps
  Input packets:           3                0 pps
  Output packets:        5656951            0 pps
Protocol inet, MTU: 1500
Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold
cnt: 0, NH drop cnt: 0
Generation: 314, Route table: 0
  Flags: Sendbroadcast-pkt-to-re
  Addresses, Flags: Is-Preferred Is-Primary
    Destination: 10.10.10/24, Local: 10.10.10.1, Broadcast: 10.10.10.255,
Generation: 224
  Protocol mpls, MTU: 1488, Maximum labels: 3, Generation: 315, Route table: 0
    Flags: Is-Primary
  Protocol multiservice, MTU: Unlimited, Generation: 316, Route table: 0
    Flags: Is-Primary
    Policer: Input: __default_arp_policer__

Logical interface af2.1 (Index 336) (SNMP ifIndex 649) (Generation 235)
Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.2 ] Encapsulation: ENET2
Traffic statistics:
  Input bytes :          0
  Output bytes :          0
  Input packets:         0
  Output packets:         0
Local statistics:
  Input bytes :          0
  Output bytes :          0
  Input packets:         0
  Output packets:         0
Transit statistics:
  Input bytes :          0                0 bps
  Output bytes :          0                0 bps
  Input packets:         0                0 pps
  Output packets:         0                0 pps
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
Generation: 317, Route table: 0
Flags: Sendbcst-pkt-to-re
Protocol mpls, MTU: 1488, Maximum labels: 3, Generation: 318, Route table: 0

Protocol multiservice, MTU: Unlimited, Generation: 319, Route table: 0
  Policer: Input: __default_arp_policer__

Logical interface af2.32767 (Index 337) (SNMP ifIndex 675) (Generation 236)
Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2
Traffic statistics:
  Input bytes :                0
  Output bytes :                0
  Input packets:               0
  Output packets:              0
Local statistics:
  Input bytes :                0
  Output bytes :                0
  Input packets:               0
  Output packets:              0
Transit statistics:
  Input bytes :                0                0 bps
  Output bytes :               0                0 bps
  Input packets:               0                0 pps
  Output packets:              0                0 pps
Protocol multiservice, MTU: Unlimited, Generation: 320, Route table: 0
Flags: None
Policer: Input: __default_arp_policer__

```

show interfaces af-interface-name extensive

This output is applicable to Junos OS 18.3R1 and later releases.

```
user@router> show interfaces af2 extensive
```

```

Physical interface: af2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 674, Generation: 349
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Clocking: Unspecified,
  Speed: 130000Mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Physical info  : Unspecified
  Hold-times    : Up 0 ms, Down 0 ms
  Damping       : half-life: 0 sec, max-suppress: 0 sec, reuse: 0, suppress: 0,
  state: unsuppressed
  Current address: 00:90:69:fd:85:a4, Hardware address: 00:90:69:fd:85:a4
  Alternate link address: Unspecified
  Last flapped   : 2018-11-01 20:44:26 PDT (6d 02:57 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes :          23382976                0 bps
    Output bytes :         652665950                0 bps
    Input packets:           341635                0 pps
    Output packets:        5986329                0 pps
  IPv6 transit statistics:
    Input bytes :                0
    Output bytes :                0

```

```

Input packets:          0
Output packets:         0
Input errors:
  Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards:
0, Resource errors: 0
Output errors:
  Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
0
Bandwidth      : 130 Gbps
Peer GNF id    : 2
Peer GNF Forwarding element(FE) view :
FPC slot:FE num  FE Bandwidth(Gbps) Status      Transmit Packets      Transmit
Bytes
      2:0                130          Up          5656951
622264610

Residual Transmit Statistics :
Packets :          0 Bytes :          0

Fabric Queue Statistics :
FPC slot:FE num  High priority(pkts)      Low priority(pkts)
      2:0                5656951          0
FPC slot:FE num  High priority(bytes)      Low priority(bytes)
      2:0                622264610        0
Residual Queue Statistics :
  High priority(pkts)      Low priority(pkts)
      0                    0
  High priority(bytes)      Low priority(bytes)
      0                    0

Logical interface af2.0 (Index 334) (SNMP ifIndex 647) (Generation 234)
Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ] Encapsulation: ENET2
Traffic statistics:
Input bytes :          23383187
Output bytes :          650689682
Input packets:          341638
Output packets:          5986329
Local statistics:
Input bytes :          23382976
Output bytes :          28425072
Input packets:          341635
Output packets:          329378
Transit statistics:
Input bytes :          211          0 bps
Output bytes :          622264610      0 bps
Input packets:          3          0 pps
Output packets:          5656951      0 pps
Protocol inet, MTU: 1500
Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold
cnt: 0, NH drop cnt: 0
Generation: 314, Route table: 0
Flags: Sendbroadcast-pkt-to-re
Addresses, Flags: Is-Preferred Is-Primary
Destination: 10.10.10/24, Local: 10.10.10.1, Broadcast: 10.10.10.255,
Generation: 224
Protocol mpls, MTU: 1488, Maximum labels: 3, Generation: 315, Route table: 0

Flags: Is-Primary
Protocol multiservice, MTU: Unlimited, Generation: 316, Route table: 0
Flags: Is-Primary

```

```

    Policer: Input: __default_arp_policer__

Logical interface af2.1 (Index 336) (SNMP ifIndex 649) (Generation 235)
Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.2 ] Encapsulation: ENET2
Traffic statistics:
  Input bytes : 0
  Output bytes : 0
  Input packets: 0
  Output packets: 0
Local statistics:
  Input bytes : 0
  Output bytes : 0
  Input packets: 0
  Output packets: 0
Transit statistics:
  Input bytes : 0 0 bps
  Output bytes : 0 0 bps
  Input packets: 0 0 pps
  Output packets: 0 0 pps
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
Generation: 317, Route table: 0
Flags: Sendbcst-pkt-to-re
Protocol mpls, MTU: 1488, Maximum labels: 3, Generation: 318, Route table: 0

Protocol multiservice, MTU: Unlimited, Generation: 319, Route table: 0
Policer: Input: __default_arp_policer__

Logical interface af2.32767 (Index 337) (SNMP ifIndex 675) (Generation 236)
Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2
Traffic statistics:
  Input bytes : 0
  Output bytes : 0
  Input packets: 0
  Output packets: 0
Local statistics:
  Input bytes : 0
  Output bytes : 0
  Input packets: 0
  Output packets: 0
Transit statistics:
  Input bytes : 0 0 bps
  Output bytes : 0 0 bps
  Input packets: 0 0 pps
  Output packets: 0 0 pps
Protocol multiservice, MTU: Unlimited, Generation: 320, Route table: 0
Flags: None
Policer: Input: __default_arp_policer__

```

show interfaces af-interface-name statistics

This output is applicable to Junos OS 18.3R1 and later releases.

```
user@router> show interfaces af4 statistics
```

```

Physical interface: af4, Enabled, Physical link is Up
Interface index: 165, SNMP ifIndex: 958
Type: Ethernet, Link-level type: Flexible-Ethernet, MTU: 1522, Speed: 240000mbps

```

```

Device flags      : Present Running
Interface flags: Internal: 0x4000
Link type        : Full-Duplex
Current address: 00:90:69:c7:31:62, Hardware address: 00:90:69:c7:31:62
Last flapped    : 2018-08-07 21:47:10 PDT (00:58:48 ago)
Statistics last cleared: Never
Input rate      : 1523826080 bps (2976222 pps)
Output rate     : 112168 bps (232 pps)
Input errors: 0, Output errors: 0
Bandwidth       : 240 Gbps
Peer GNF id     : 4
Peer GNF Forwarding element(FE) view :
FPC slot:FE num  FE Bandwidth(Gbps) Status    Transmit Packets    Transmit
Bytes
      12:0                120      Up              2802
168120
      12:1                120      Up              2848
170880

Residual Transmit Statistics :
Packets :                0 Bytes :                0

Fabric Queue Statistics :
FPC slot:FE num  High priority(pkts)    Low priority(pkts)
      12:0                2802                0
      12:1                2848                0
FPC slot:FE num  High priority(bytes)    Low priority(bytes)
      12:0                168120            0
      12:1                170880            0

Residual Queue Statistics :
      High priority(pkts)    Low priority(pkts)
              0                0
      High priority(bytes)    Low priority(bytes)
              0                0

Logical interface af4.1 (Index 335) (SNMP ifIndex 7354)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 ] Encapsulation: ENET2
  Input packets : 71572652
  Output packets: 5740
  Protocol inet, MTU: 1500
  Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold
  cnt: 0, NH drop cnt: 0
  Flags: Sendbroadcast-pkt-to-re
  Addresses, Flags: Is-Preferred Is-Primary
    Destination: 20.0.0/24, Local: 20.0.0.1, Broadcast: 20.0.0.255
  Protocol mpls, MTU: 1488, Maximum labels: 3
  Protocol multiservice, MTU: Unlimited

Logical interface af4.32767 (Index 336) (SNMP ifIndex 7355)
  Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ] Encapsulation: ENET2
  Input packets : 0
  Output packets: 0
  Protocol multiservice, MTU: Unlimited
  Flags: None

```

show interfaces af-interface-name

This output is applicable to Junos OS versions prior to 18.3R1.

```
user@router> show interfaces af4
```

```
Physical interface: af4, Enabled, Physical link is Up
  Interface index: 139, SNMP ifIndex: 514
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Speed: 480000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Current address: 2c:6b:f5:55:eb:f6, Hardware address: 2c:6b:f5:55:eb:f6
  Last flapped   : 2017-10-18 19:40:00 EDT (02:47:02 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Bandwidth      : 480 Gbps
  Peer GNF id    : 4
  Peer GNF Forwarding element(FE) view :
      FPC slot:FE Num      FE Bandwidth(Gbps)      Status
      6:0                  240                    Up
      6:1                  240                    Up

Logical interface af4.1 (Index 328) (SNMP ifIndex 593)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ] Encapsulation: ENET2
  Input packets : 9
  Output packets: 9
  Protocol inet, MTU: 1500
  Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1,
  Curr new hold cnt: 0, NH drop cnt: 0
  Flags: Sendbroadcast-pkt-to-re
  Addresses, Flags: Is-Preferred Is-Primary
    Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255
  Protocol multiservice, MTU: Unlimited
```

show interfaces af-interface-name brief

This output is applicable to Junos OS versions prior to 18.3R1.

```
user@router> show interfaces af4 brief
```

```
Physical interface: af4, Enabled, Physical link is Up
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Clocking: Unspecified,
  Speed: 480000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000

Logical interface af4.1
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ] Encapsulation: ENET2
  inet 1.1.1.1/24
  multiservice
```

show interfaces af-interface-name extensive

This output is applicable to Junos OS versions prior to 18.3R1.

```
user@router> show interfaces af4 extensive
```

```
Physical interface: af4, Enabled, Physical link is Up
  Interface index: 139, SNMP ifIndex: 514, Generation: 142
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Clocking: Unspecified,
```

```

Speed: 480000mbps
Device flags : Present Running
Interface flags: Internal: 0x4000
Link type : Full-Duplex
Physical info : Unspecified
Hold-times : Up 0 ms, Down 0 ms
Damping : half-life: 0 sec, max-suppress: 0 sec, reuse: 0, suppress: 0,
state: unsuppressed
Current address: 2c:6b:f5:55:eb:f6, Hardware address: 2c:6b:f5:55:eb:f6
Alternate link address: Unspecified
Last flapped : 2017-10-18 19:40:00 EDT (02:50:04 ago)
Statistics last cleared: Never
Traffic statistics:
Input bytes : 4048 0 bps
Output bytes : 144092 59440 bps
Input packets: 88 0 pps
Output packets: 186 4 pps
IPv6 transit statistics:
Input bytes : 0
Output bytes : 0
Input packets: 0
Output packets: 0
Input errors:
Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Giants: 0, Policed discards:
0, Resource errors: 0
Output errors:
Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
0
Bandwidth : 480 Gbps
Peer GNF id : 4
Peer GNF Forwarding element(FE) view :
      FPC slot:FE Num      FE Bandwidth(Gbps)      Status
      6:0                240                Up
      6:1                240                Up

Logical interface af4.1 (Index 328) (SNMP ifIndex 593) (Generation 137)
Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ] Encapsulation: ENET2
Traffic statistics:
Input bytes : 414
Output bytes : 139906
Input packets: 9
Output packets: 107
Local statistics:
Input bytes : 414
Output bytes : 598
Input packets: 9
Output packets: 13
Transit statistics:
Input bytes : 0 0 bps
Output bytes : 139308 59240 bps
Input packets: 0 0 pps
Output packets: 94 4 pps
Protocol inet, MTU: 1500
Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold
cnt: 0, NH drop cnt: 0
Generation: 162, Route table: 0
Flags: Sendbroadcast-pkt-to-re
Output Filters: f-basic-sr-tcm-ca
Addresses, Flags: Is-Preferred Is-Primary
Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255, Generation:

```

148

```
Protocol multiservice, MTU: Unlimited, Generation: 163, Route table: 0
  Policer: Input: __default_arp_policer__
```

show interfaces af-interface-name detail

This output is applicable to Junos OS versions prior to 18.3R1.

```
user@router> show interfaces af4 detail
```

```
Physical interface: af4, Enabled, Physical link is Up
  Interface index: 139, SNMP ifIndex: 514, Generation: 142
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Clocking: Unspecified,
  Speed: 480000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Physical info  : Unspecified
  Hold-times     : Up 0 ms, Down 0 ms
  Damping        : half-life: 0 sec, max-suppress: 0 sec, reuse: 0, suppress: 0,
  state: unsuppressed
  Current address: 2c:6b:f5:55:eb:f6, Hardware address: 2c:6b:f5:55:eb:f6
  Alternate link address: Unspecified
  Last flapped   : 2017-10-18 19:40:00 EDT (02:55:48 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes   :          77518216910          1944810944 bps
    Output bytes  :          3059154          68832 bps
    Input packets :          52450649          164494 pps
    Output packets:           4672          20 pps
  IPv6 transit statistics:
    Input bytes   :          0
    Output bytes  :          0
    Input packets :          0
    Output packets:          0
  Bandwidth      : 480 Gbps
  Peer GNF id    : 4
  Peer GNF Forwarding element(FE) view :
      FPC slot:FE Num      FE Bandwidth(Gbps)      Status
      6:0                  240                      Up
      6:1                  240                      Up

Logical interface af4.1 (Index 328) (SNMP ifIndex 593) (Generation 137)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ] Encapsulation: ENET2
  Traffic statistics:
    Input bytes   :          77518213184
    Output bytes  :          3054342
    Input packets :          52450568
    Output packets:          4591
  Local statistics:
    Input bytes   :          460
    Output bytes  :          4600
    Input packets :          10
    Output packets:          100
  Transit statistics:
    Input bytes   :          77518212724          1944810944 bps
    Output bytes  :          3049742          68632 bps
    Input packets :          52450558          164494 pps
```

```
Output packets:          4491          20 pps
Protocol inet, MTU: 1500
Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold
cnt: 0, NH drop cnt: 0
Generation: 162, Route table: 0
Flags: Sendbcst-pkt-to-re
Output Filters: f-basic-sr-tcm-ca
Addresses, Flags: Is-Preferred Is-Primary
Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255, Generation:
148
Protocol multiservice, MTU: Unlimited, Generation: 163, Route table: 0
Policer: Input: __default_arp_policer__
```


show system anomalies (GNF)

Syntax	show system anomalies (all-anomalies config fru system)
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display the incompatibilities between the software version running on the guest network function (GNF) and the version running on the base system (BSYS).
Options	<p>all-anomalies—Display the multiversion software incompatibilities from all categories—system, configuration, and FRU.</p> <p>config—Display the feature incompatibilities between software versions.</p> <p>fru—Display the FRU-level incompatibilities between software versions. This can be an incompatibility pertaining to the support for a specific FRU.</p> <p>system—Display the system-level incompatibilities between software versions. These include interprocess communication (IPC) message, CLI, or SNMP incompatibility.</p>
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • show system anomalies gnf-id (BSYS) on page 122 • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170
Output Fields	Table 20 on page 153 lists the output fields for the show system anomalies command. Output fields are listed in the approximate order in which they appear.

Table 20: show system anomalies gnf-id Output Fields

Field Name	Field Description
Anomaly Type	Shows the software incompatibility type. The following are the possible values: <ul style="list-style-type: none"> • SYS—Indicates system-specific incompatibilities. • FRU—Indicates FRU-specific incompatibilities. • CONFIG—Indicates feature-specific incompatibilities.
Anomaly ID	Shows the incompatibility ID.

Table 20: show system anomalies gnfx-id Output Fields (continued)

Field Name	Field Description
Default Action	Shows the default actions associated with incompatibilities. The following are the possible values: <ul style="list-style-type: none"> WARN—Indicates the presence of a minor incompatibility. It causes a warning. ABORT—Indicates the presence of a major incompatibility. It causes an abort action.
Class	Indicates of the incompatibility is present in GNF, or BSYS, or both.
Message	Incompatibility description.
FRU ID	Field-replaceable unit (FRU) ID. Applicable in the case of FRU-specific incompatibilities.
Firmware	Firmware of the FRU.

Sample Output

```
user@router> show system anomalies
```

Anomaly Type	Anomaly ID FRU ID	Default Action Firmware	Class	Message
SYS incompatibility 1	100	WARN	GNF Present	system
SYS incompatibility 2	103	WARN	BOTH Present	system
SYS incompatibility 3	200	WARN	BSYS Present	system

CHAPTER 7

Configuration Statements for JDM

- [virtual-network-functions](#) on page 156
- [server](#) on page 157
- [interfaces \(Junos Node Slicing\)](#) on page 158
- [routing-options \(Junos Node Slicing\)](#) on page 159
- [system login \(Junos Node Slicing\)](#) on page 160
- [root-login \(JDM\)](#) on page 161

virtual-network-functions

Syntax

```
virtual-network-functions vnf-name {
  base-config base-config ;
  chassis-type chassis-type;
  id id;
  resource-template resource-template;
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 17.2R1.

Description Associate a GNF ID, base configuration, chassis type and resource template with the VNF.

The GNFs that are configured and committed will appear as auto-complete options in operational commands.

Options

vnf-name—Name of the VNF.

base-config base-config—Choose a base configuration for the VNF.

chassis-type chassis-type—Applicable only to Junos Node Slicing set up using external server model. Choose the type of the router chassis (for example, MX960) used as the base system (BSYS) in the node slicing setup.

id id—Assign a GNF ID.

Range: 1–10

Range: 1-4 (for in-chassis model)

resource-template resource-template—Assign a resource template to the VNF. The server resource template defines the number of dedicated CPU cores and the size of DRAM to be assigned to a VNF.

Required Privilege Level root

Related Documentation

- [server on page 157](#)

server

Syntax

```
server {
  interfaces {
    cb0 cb0-interface;
    cb1 cb1-interface;
    jdm-management jdm-management-interface;
    vnf-management gnf-management-interface;
  }
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 17.2R1.

Description Applicable only to Junos Node Slicing set up using external server model.

Configure the server interfaces for the JDM and GNFs. These include a JDM management interface, a GNF management interface, and two server interfaces that are connected to the MX Series router.

Options

cb0 *cb0-interface*—The server interface that is connected to the control board 0 of the MX Series router.

cb1 *cb1-interface*—The server interface that is connected to the control board 1 of the MX Series router.

jdm-management *jdm-management-interface*—The server interface to be used as the JDM management interface.

vnf-management *gnf-management-interface*—The server interface to be used as GNF management interface.

Required Privilege Level root

Related Documentation

- [virtual-network-functions on page 156](#)

interfaces (Junos Node Slicing)

Syntax

```
interfaces interface-name{  
  unit unit {  
    family (inet|inet6) {  
      address address;  
    }  
  }  
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 17.2.

Description Configure an IP address for the JDM management interface.

Options *interface-name*—Name of the interface.

unit unit—Interface unit number. This is a logical unit number. The only supported value is 0.

family (inet|inet6)—Protocol family.

- *inet*—Indicates IPv4.
- *inet6*—Indicates IPv6.

address address—IP address of the interface.

Required Privilege Level root

Related Documentation

- [server on page 157](#)
- [virtual-network-functions on page 156](#)

routing-options (Junos Node Slicing)

Syntax

```
routing-options {  
  static {  
    route route {  
      next-hop next-hop;  
    }  
  }  
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 17.2.

Description Configure static routes for JDM.

Options

- route *route***—Static route destination.
- next-hop *next-hop***—Specify the address of the next hop to destination.

Required Privilege Level root

Related Documentation

- [server on page 157](#)
- [virtual-network-functions on page 156](#)

system login (Junos Node Slicing)

Syntax	<pre>system login { user <i>username</i> ; class <i>class-name</i>; }</pre>
Hierarchy Level	[root]
Release Information	Statement introduced in Junos OS Release 18.3 R1.
Description	Create a non-root user in JDM for Junos Node Slicing. The non-root user accounts function only inside JDM, not on the host server.
Options	<p>user <i>username</i>—Username of the account.</p> <p>class <i>class-name</i>—Predefined login classes that JDM supports for non-root users.</p> <ul style="list-style-type: none">• super-user• operator• read-only• unauthorized
Required Privilege Level	
Related Documentation	<ul style="list-style-type: none">• Configuring Non-Root Users in JDM (Junos Node Slicing) on page 47• Configuring JDM on the x86 Servers (External Server Model) on page 45

root-login (JDM)

Syntax	<code>root-login (allow deny);</code>
Hierarchy Level	<code>[edit system services ssh]</code>
Release Information	Statement introduced in Junos OS Release 18.3.
Description	Control user access to the JDM through SSH.
Default	<code>root-login allow</code> is the default.
Options	<p>allow—Allow users to log in to the JDM as root through SSH.</p> <p>deny—Disable users from logging in to the JDM as root through SSH. This configuration option is applicable only to the JDM management interface (jmgmt0). Setting this configuration option does not block the internal JDM to JDM communication, which uses root account with password-less authentication method.</p>
Required Privilege Level	<p><code>admin</code>—To view this statement in the configuration.</p> <p><code>admin-control</code>—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Configuring JDM on the x86 Servers (External Server Model) on page 45

CHAPTER 8

Operational Commands for JDM

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [clear log \(JDM\)](#)
- [monitor list \(JDM\)](#)
- [monitor start \(JDM\)](#)
- [monitor stop \(JDM\)](#)
- [request server authenticate-peer-server](#)
- [request virtual-network-functions](#)
- [show virtual-network-functions](#)
- [show version vnf](#)
- [show version \(JDM\)](#)
- [show system cpu \(JDM\)](#)
- [show system storage \(JDM\)](#)
- [show system memory \(JDM\)](#)
- [show system network \(JDM\)](#)
- [restart \(JDM\)](#)

Generic Guidelines for Using JDM Server Commands

The following are general guidelines on how to use the JDM server commands:

- Append **all-servers** to an operational command to take action on both the servers. Example: **request virtual-network-functions gnfl restart all-servers**.
- Append **server0** or **server1** to an operational command to take action on server0 or server1. Example: **request virtual-network-functions gnfl restart server0**.

By default, the operational commands work only on the local JDM.

- Use the **commit synchronize** command to ensure that the configuration committed on one server is synchronized with the other server. The synchronization is bidirectional. A JDM configuration change at either of the servers is synchronized with the other server. When a virtual machine (VM) is instantiated, the GNF-re0 VM instance starts on server0 and the GNF-re1 VM instance starts on server1.



NOTE: If you do not use the `commit synchronize` command, you must configure and manage the VMs on both the servers manually.

**Related
Documentation**

- [show virtual-network-functions on page 172](#)
- [request virtual-network-functions on page 171](#)
- [request server authenticate-peer-server on page 170](#)

clear log (JDM)

Syntax	<code>clear log <i>file-name</i></code>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Clear the system log or trace files.
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none">• monitor start (JDM) on page 167
List of Sample Output	clear log on page 165
Output Fields	This command produces no output.

Sample Output

clear log

```
user@jdm> clear log syslog
```

monitor list (JDM)

Syntax `monitor list`

Release Information Command introduced in Junos OS Release 17.4R1.

Description Show status of monitored files.

Required Privilege Level View

Related Documentation

- [monitor start \(JDM\) on page 167](#)

List of Sample Output [monitor list on page 166](#)

Sample Output

`monitor list`

```
user@jdm> monitor list
```

monitor start (JDM)

Syntax	monitor start <i>file-name</i>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Start displaying the system log or trace file and additional entries being added to those files.
Additional Information	Log files are generated by the routing protocol process or by system logging.
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> monitor stop (JDM) on page 169
List of Sample Output	monitor start on page 167
Output Fields	Table 21 on page 167 describes the output fields for the monitor start command. Output fields are listed in the approximate order in which they appear.

Table 21: monitor start Output Fields

Field Name	Field Description
filename	Name of the file from which entries are being displayed.
Date and time	Timestamp for the log entry.

Sample Output

monitor start

```

user@jdm> monitor start syslog

*** syslog ***
Oct 19 19:44:21 jdm mgd[4069]: UI_CMDLINE_READ_LINE: User 'root', command 'monit
    or start syslog '
Oct 19 19:44:29 jdm mgd[3268]: UI_CMDLINE_READ_LINE: User 'root', command 'delet
    e groups server0 server interfaces cb1 '
Oct 19 19:44:29 jdm mgd[3268]: UI_CFG_AUDIT_OTHER: User 'root' delete: [groups s
    erver0 server interfaces cb1]
Oct 19 19:44:35 jdm mgd[3268]: UI_CMDLINE_READ_LINE: User 'root', command 'delet
    e groups server1 server interfaces cb1 '

```

```

Oct 19 19:44:35 jdm mgd[3268]: UI_CFG_AUDIT_OTHER: User 'root' delete: [groups s
    erver1 server interfaces cb1]
Oct 19 19:44:36 jdm mgd[3268]: UI_CMDLINE_READ_LINE: User 'root', command 'commi
    t '
Oct 19 19:44:36 jdm mgd[3268]: UI_COMMIT: User 'root' requested 'commit' operati
    on (comment: none)
Oct 19 19:44:36 jdm mgd[3268]: UI_COMMIT_PROGRESS: Commit operation in progress:
    Obtaining lock for commit
Oct 19 19:44:36 jdm mgd[3268]: UI_COMMIT_PROGRESS: Commit operation in progress:
    updating commit revision
Oct 19 19:44:36 jdm mgd[3268]: UI_COMMIT_PROGRESS: Commit operation in progress:
    obtaining db lock on  server1
Oct 19 19:44:36 jdm mgd[3268]: UI_COMMIT_PROGRESS: Commit operation in progress:
    UI extensions feature is not configured
Oct 19 19:44:36 jdm mgd[3268]: UI_COMMIT_PROGRESS: Commit operation in progress:
    Started running translation script

```


monitor stop (JDM)

Syntax	<code>monitor stop <i>file-name</i></code>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Stop displaying the system log or trace file.
Additional Information	Log files are generated by the routing protocol process or by system logging.
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none">• monitor start (JDM) on page 167
Output Fields	This command produces no output.

request server authenticate-peer-server

Syntax request server authenticate-peer-server

Release Information Command introduced in Junos OS Release 17.4R1.

Description Copy the **ssh** public key to the peer JDM. This command is equivalent to **ssh-copy-id user@jdm-server<0/1>**.



NOTE: If the JDM fails to establish SSH connection with its peer on either of the two CB links, you need to run the JDM CLI command **request server authenticate-peer-server**. You can use the JDM CLI command **show server connections** to view the status of the SSH connection between the JDM peers. Note that the command **request server authenticate-peer-server** will prompt for user confirmation twice - once per CB link.

Required Privilege Level View

Related Documentation

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [show virtual-network-functions on page 172](#)
- [request virtual-network-functions on page 171](#)

Sample Output

```
user@jdm> request server authenticate-peer-server
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out anythat are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
user@jdm-server1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'user@jdm-server1'"
and check to make sure that only the key(s) you wanted were added.
```

request virtual-network-functions

Syntax `request virtual-network-functions vnf-name (add-image | all-servers | delete-image | force | restart | server | start | stop)`

Release Information Command introduced in Junos OS Release 17.2R1.
The **force** option is available from Junos OS Release 17.4R1.

Description Start, stop or restart the VNFs. Also, you can add or remove the base image.



NOTE: You can issue these commands either on both the servers (`server0` and `server1`) or on one specific server.

Options ***vnf-name***—Name of the VNF.

add-image—Add the base image to the VNF repository.

all-servers—Issue the command on both the servers.

delete-image—Delete the base image from the VNF repository.

force—Overwrite the image that was added earlier (example: `request virtual-network-functions vnf-name add-image new-image-name force`). The **force** option is also used with **delete-image** command to perform a cleanup in case, for example, you abruptly stopped an earlier **add-image** process by pressing Ctrl-C.

restart—Restart the VNF specified.

server—Issue the command on a specific server. Applicable value is **0** or **1**.

start—Start the VNF.

stop—Stop the VNF.

Required Privilege Level View

Related Documentation

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [show virtual-network-functions on page 172](#)
- [request server authenticate-peer-server on page 170](#)

show virtual-network-functions

Syntax `show virtual-network-functions (all-servers | server | slot)
<vnf-name>
<vnf-name detail>
<vnf-name vnf-xml-profile>`

Release Information Command introduced in Junos OS Release 17.2R1.

Description Display the list of guest network functions (GNFs) along with their IDs, status and availability.

Options **all-servers**—Display the details of the GNFs on both the servers.
server—Display the details of the GNFs on one specific server. Applicable value is **0** or **1**.
vnf-name—Display additional details of a particular GNF. You can use the **detail** option to view the detailed output. For example, **show virtual-network-functions gnf1 detail**.
vnf-xml-profile—Displays the xml profile of a GNF.

Required Privilege Level View

Related Documentation

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [request virtual-network-functions on page 171](#)
- [request server authenticate-peer-server on page 170](#)

Output Fields [Table 22 on page 172](#) lists the output fields for the **show virtual-network-functions** command.

Table 22: show virtual network functions Output Fields

Field Name	Field Description
ID	The ID associated with the VNF.
Name	Name of the VNF.
State	Status of the VNF. <ul style="list-style-type: none"> • Running—The VNF is online and running. • Shut off—The VNF is in the shut down state.

Table 22: *show virtual network functions Output Fields (continued)*

Field Name	Field Description
Liveness	Indicates the availability of VNF. <ul style="list-style-type: none"> • Up • Down
Name	Name of the VNF.
IP Address	IP address of the VNF.
Status	Status of a particular VNF. <ul style="list-style-type: none"> • Running—The VNF is online and running. • Shut off—The VNF is in the shut down state.
Cores	Number of cores in the VNF.
Memory	The VNF memory.
Resource Template	The resource template associated with a VNF.
Qemu Process id	Qemu process ID.
VNF CPU Utilization and Allocation Information	Shows the GNF CPU utilization details. See also: show system cpu (JDM) .
VNF Memory Information	Displays the following memory information about the GNFs: <ul style="list-style-type: none"> • Name—GNF name. • Resident—The memory used by the GNFs. • Actual—Actual memory.
VNF Storage Information	Displays the following guest network function (GNF) storage information: <ul style="list-style-type: none"> • Directories—Names of the directories. • Size—Total storage size. • Used—Storage used.
VNF Interfaces Statistics	Shows the GNF interface statistics information. See also: show system network (JDM) .
VNF Network Information	Shows the list of Physical Interfaces, Virtual Interfaces and MAC addresses.

Sample Output

show virtual-network-functions

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

ID	Name	State	Liveness
1	bittern-gnf-a	Running	Up
2	bittern-gnf-b	Running	Up
3	bittern-gnf-c	Running	Up
4	bittern-gnf-d	Running	Up
5	bittern-gnf-e	Running	Up

Sample Output

show virtual-network-functions (for a specific VNF)

```
user@jdm> show virtual-network-functions gnfl
```

Virtual Machine Information

```
-----
Name:          gnfl
IP Address:    192.168.2.1
Status:        Running
Liveness:      up
Cores:         4
Memory:        32GB
Resource Template: 4core-32g
Qemu Process id: 10891
```

Sample Output

show virtual-network-functions <vnf-name> detail

```
user@jdm> show virtual-network-functions gnfl detail
```

VNF Information

```
-----
ID          1
Name:       gnfl
Status:     Running
Liveness:   up
IP Address: 192.168.2.1
Cores:      2
Memory:     16GB
Resource Template: 2core-16g
Qemu Process id: 20478
SMBIOS version: v1
```

VNF Uptime: 206:09.58

VNF CPU Utilization and Allocation Information

GNF	CPU-Id(s)	Usage	Qemu Pid
gnfl	10,11	6.1%	20478

VNF Memory Information

Name	Actual	Resident
gnf1	16.0G	15.5G

VNF Storage Information

Directory	Size	Used
/vm-primary/gnf1	50.2G	4.2G

VNF Interfaces Statistics

Interface	bytes	Trxd	Packets	Trxd	Bytes	Rcvd	packets	Error	Drop	Trxd
macvtap0	261601960	3545272	0	0	19077011	335687	0	0		
macvtap1	172763580	2786344	0	0	99369778	689729	0	0		
vnet1	250	0	0	0	24686	527	0	0	11620	
macvtap2	861020752	10813152	0	0	428385	7405	0	0		

VNF Network Information

Virtual Interface	Physical Interface	MAC
macvtap0	ens3f0	00:85:49:54:cd:30
macvtap1	ens3f1	00:85:49:54:cd:31
vnet1	bridge_jdm_vm	00:85:49:54:cd:32
macvtap2	enp3s0f2	00:85:49:54:cd:33

show virtual-network-functions <vnf-name> vnf-xml-profile

```
user@jdm> show virtual-network-functions gnf1 vnf-xml-profile
```

```
<domain type='kvm' id='1'>
  <name>gnf1</name>
  <uuid>a1c62c35-910f-4354-991c-7ad26c1b32e7</uuid>
  <memory unit='KiB'>33554432</memory>
  <currentMemory unit='KiB'>33554432</currentMemory>
  <vcpu placement='static'>4</vcpu>
  <cputune>
    <vcpupin vcpu='0' cpuset='4'>/>
    <vcpupin vcpu='1' cpuset='5'>/>
    <vcpupin vcpu='2' cpuset='6'>/>
    <vcpupin vcpu='3' cpuset='7'>/>
    <emulatorpin cpuset='2-3'>/>
  </cputune>
</domain>
```

```

<resource>
  <partition>/machine</partition>
</resource>
<sysinfo type='smbios'>
  <system>
    <entry name='manufacturer'>JUNIPER</entry>
    <entry name='product'>RE-GNF</entry>
    <entry
name='version'>v2-mx480-gnf1-re0-vtnet0-vtnet1-vtnet3-vtnet2-external-0x009069136000:0x0800</entry>

  </system>
</sysinfo>
<os>
  <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
  <boot dev='hd'>/>
  <smbios mode='sysinfo'>/>
</os>
<features>
  <acpi>/>
  <apic>/>
</features>
<clock offset='utc'>/>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>restart</on_crash>
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='raw' cache='directsync' io='native'>/>
    <source file='/vm-primary/gnf1/gnf1.img'>/>
    <backingStore>/>
    <target dev='hda' bus='virtio'>/>
    <alias name='virtio-disk0'>/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'>/>

  </disk>
  <disk type='file' device='disk'>
    <driver name='qemu' type='raw' cache='directsync' io='native'>/>
    <source file='/vm-primary/gnf1/var-config.img'>/>
    <backingStore>/>
    <target dev='vdb' bus='virtio'>/>
    <alias name='virtio-disk1'>/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'>/>

  </disk>
  <disk type='file' device='disk'>
    <driver name='qemu' type='raw' cache='directsync' discard='unmap'>/>
    <source file='/vm-primary/gnf1/swap-disk.img'>/>
    <backingStore>/>
    <target dev='hdc' bus='virtio'>/>
    <alias name='virtio-disk2'>/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0'>/>

  </disk>
  <disk type='file' device='disk'>
    <driver name='qemu' type='raw' cache='directsync' discard='unmap'>/>
    <source file='/vm-primary/gnf1/aux-disk.img'>/>
    <backingStore>/>
    <target dev='hdb' bus='ide'>/>
    <alias name='ide0-0-1'>/>

```



```

    <address type='drive' controller='0' bus='0' target='0' unit='1'>
  </disk>
  <disk type='file' device='disk'>
    <driver name='qemu' type='raw' cache='directsync'>
    <source file='/vm-primary/gnf1/usb-disk.img'>
    <backingStore/>
    <target dev='sda' bus='usb'>
    <alias name='usb-disk0'>
    <address type='usb' bus='0' port='1'>
  </disk>
  <controller type='usb' index='0'>
    <alias name='usb'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'>

  </controller>
  <controller type='pci' index='0' model='pci-root'>
    <alias name='pci.0'>
  </controller>
  <controller type='ide' index='0'>
    <alias name='ide'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'>

  </controller>
  <interface type='direct' trustGuestRxFilters='yes'>
    <mac address='52:54:00:09:ab:83'>
    <source dev='ens2f0' mode='vepa'>
    <target dev='macvtap0'>
    <model type='virtio'>
    <alias name='net0'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'>

  </interface>
  <interface type='direct' trustGuestRxFilters='yes'>
    <mac address='52:54:00:45:e3:ba'>
    <source dev='ens2f1' mode='vepa'>
    <target dev='macvtap1'>
    <model type='virtio'>
    <alias name='net1'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'>

  </interface>
  <interface type='bridge'>
    <mac address='52:54:00:ee:73:93'>
    <source bridge='bridge_jdm_vm'>
    <target dev='vnet1'>
    <model type='virtio'>
    <alias name='net2'>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'>

  </interface>
  <interface type='direct'>
    <mac address='00:90:69:13:7f:00'>
    <source dev='eno2' mode='bridge'>
    <bandwidth>
      <inbound average='125000' peak='125000' burst='256'>
      <outbound average='125000' peak='125000' burst='256'>
    </bandwidth>
    <target dev='macvtap2'>
    <model type='virtio'>
    <driver name='qemu'>

```

```
<alias name='net3' />
<address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />

</interface>
<serial type='pty'>
  <source path='/dev/pts/6' />
  <target port='0' />
  <alias name='serial0' />
</serial>
<console type='pty' tty='/dev/pts/6'>
  <source path='/dev/pts/6' />
  <target type='serial' port='0' />
  <alias name='serial0' />
</console>
<input type='mouse' bus='ps2'>
  <alias name='input0' />
</input>
<input type='keyboard' bus='ps2'>
  <alias name='input1' />
</input>
<memballoon model='virtio'>
  <alias name='balloon0' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0' />

</memballoon>
</devices>
<seclabel type='none' model='none' />
<seclabel type='dynamic' model='dac' relabel='yes'>
  <label>+107:+107</label>
  <imagelabel>+107:+107</imagelabel>
</seclabel>
</domain>
```

show version vnf

Syntax	show version vnf <i>vnf-name</i>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display the hostname and version information about the specified guest network function (GNF).
Options	vnf-name —Name of the GNF for which you want to view the version details.
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170

Sample Output

Depending on the platform running Junos OS, you might see different installed sub-packages.

show version vnf

```
user@jdm> show version vnf gnf2
```

```

Hostname: gnf2
Model: mx960
Junos: 17.4X48-D10.3
JUNOS OS Kernel 64-bit [20170913.201739_fbsd-builder_stable_11]
JUNOS OS libs [20170913.201739_fbsd-builder_stable_11]
JUNOS OS runtime [20170913.201739_fbsd-builder_stable_11]
JUNOS OS time zone information [20170913.201739_fbsd-builder_stable_11]
JUNOS network stack and utilities [20170926.111120_builder_junos_174_x48_d10]
JUNOS modules [20170926.111120_builder_junos_174_x48_d10]
JUNOS mx modules [20170926.111120_builder_junos_174_x48_d10]
JUNOS libs [20170926.111120_builder_junos_174_x48_d10]
JUNOS OS libs compat32 [20170913.201739_fbsd-builder_stable_11]
JUNOS OS 32-bit compatibility [20170913.201739_fbsd-builder_stable_11]
JUNOS libs compat32 [20170926.111120_builder_junos_174_x48_d10]
JUNOS runtime [20170926.111120_builder_junos_174_x48_d10]
Junos vmguest package [20170926.111120_builder_junos_174_x48_d10]
JUNOS py extensions [20170926.111120_builder_junos_174_x48_d10]
JUNOS py base [20170926.111120_builder_junos_174_x48_d10]
JUNOS OS vmguest [20170913.201739_fbsd-builder_stable_11]
JUNOS OS crypto [20170913.201739_fbsd-builder_stable_11]
JUNOS mx libs compat32 [20170926.111120_builder_junos_174_x48_d10]
JUNOS mx runtime [20170926.111120_builder_junos_174_x48_d10]
JUNOS common platform support [20170926.111120_builder_junos_174_x48_d10]

```

```
JUNOS mx libs [20170926.111120_builder_junos_174_x48_d10]
JUNOS mtx Data Plane Crypto Support [20170926.111120_builder_junos_174_x48_d10]
JUNOS daemons [20170926.111120_builder_junos_174_x48_d10]
JUNOS mx daemons [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services URL Filter package [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services TLB Service PIC package [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services SSL [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services SOFTWIRE [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Stateful Firewall [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services RPM [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services PTSP Container package [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services PCEF package [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services NAT [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Mobile Subscriber Service Container package
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Services MobileNext Software package
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Logging Report Framework package
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Services LL-PDF Container package [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Jflow Container package [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Deep Packet Inspection package
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Services IPSec [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services IDS [20170926.111120_builder_junos_174_x48_d10]
JUNOS IDP Services [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services HTTP Content Management package
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Crypto [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Captive Portal and Content Delivery Container package
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Services COS [20170926.111120_builder_junos_174_x48_d10]
JUNOS AppId Services [20170926.111120_builder_junos_174_x48_d10]
JUNOS Services Application Level Gateways
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Services ACL Container package [20170926.111120_builder_junos_174_x48_d10]
JUNOS SDN Software Suite [20170926.111120_builder_junos_174_x48_d10]
JUNOS Extension Toolkit [20170926.111120_builder_junos_174_x48_d10]
JUNOS Packet Forwarding Engine Support (wrlinux)
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Packet Forwarding Engine Support (MX/EX92XX Common)
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Packet Forwarding Engine Support (M/T Common)
[20170926.111120_builder_junos_174_x48_d10]
JUNOS Packet Forwarding Engine Support (MX Common)
[20170926.111120_builder_junos_174_x48_d10]
JUNOS jfirmware [20170926.111120_builder_junos_174_x48_d10]
JUNOS Online Documentation [20170926.111120_builder_junos_174_x48_d10]
```

show version (JDM)

Syntax	show version (all-servers server vnf all-vnfs detail brief)
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display the version information about the Juniper Device Manager (JDM).
Options	<p>all-servers—Display the version details of the JDM instances on both the servers.</p> <p>server—Display the version details of the JDM instance on one specific server. Range: 0 through 1</p> <p>vnf—Display the version details for a particular guest network function (GNF). You need to mention the GNF name in the command. Example: show version vnf gnf2.</p> <p>all-vnfs—Display the version details for all the GNFs.</p> <p>(detail brief)—Display the specified level of output.</p>
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170

Sample Output

show version

```
user@jdm> show version
```

```

Hostname: mgb-dvaita-ixr1-jdm
Model: junos_node_slicing
Server slot : 1
JDM package version : 17.4-R1.7
Host Software [Red Hat Enterprise Linux]
JDM container Software [Ubuntu 14.04.1 LTS]
JDM daemon jdmd [Version: 17.4R1.7-secure]
JDM daemon jinventoryd [Version: 17.4R1.7-secure]
JDM daemon jdmmon [Version: 17.4R1.7-secure]
Host daemon jlinkmon [Version: 17.4R1.7-secure]
```

```
user@jdm> show version brief
```

```

Hostname: mgb-dvaita-ixr1-jdm
Model: junos_node_slicing
Server slot : 1
```

```
JDM package version : 17.4-R1.7
Host Software [Red Hat Enterprise Linux]
JDM container Software [Ubuntu 14.04.1 LTS]
JDM daemon jdmd [Version: 17.4R1.7-secure]
JDM daemon jinventoryd [Version: 17.4R1.7-secure]
JDM daemon jdmmon [Version: 17.4R1.7-secure]
Host daemon jlinkmon [Version: 17.4R1.7-secure]
```

```
user@jdm> show version detail
```

```
Hostname: mgb-dvaita-ixr1-jdm
Model: junos_node_slicing
Server slot : 1
JDM package version : 17.4-R1.7
Host Software [Red Hat Enterprise Linux]
JDM container Software [Ubuntu 14.04.1 LTS]
JDM daemon jdmd [Version: 17.4R1.7-secure]
JDM daemon jinventoryd [Version: 17.4R1.7-secure]
JDM daemon jdmmon [Version: 17.4R1.7-secure]
Host daemon jlinkmon [Version: 17.4R1.7-secure]
KERNEL 3.10.0-514.el7.x86_64
MGD release 17.4R1.7 built by builder on 2017-11-17 11:29:41 UTC
CLI release 17.4R1.7 built by builder on 2017-11-17 10:53:44 UTC
base-actions-dd release 17.4R1.7 built by builder on 2017-11-17 10:06:17 UTC
jdmd_common-actions-dd release 17.4R1.7 built by builder on 2017-11-17 10:06:09
UTC
jdmd_nv_jdm-actions-dd release 17.4R1.7 built by builder on 2017-11-17 10:06:09
UTC
```

show system cpu (JDM)

Syntax	<code>show system cpu</code>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display the CPUs and their usage status.
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170
Output Fields	Table 23 on page 183 describes the output fields for the show system cpu command. Output fields are listed in the approximate order in which they appear.

Table 23: show system cpu Output Fields

Field Name	Field Description
VNF	Name of the GNF.
State	GNF status. For example, "Running".
CPU-Id(s)	List of CPU IDs.
Usage	CPU usage percentage.
Qemu Pid	Qemu process ID.

Sample Output

show system cpu

```

user@jdm> show system cpu

VNF CPU Utilization and Allocation Information
-----
VNF      State      CPU-Id(s)      Usage      Qemu Pid
-----

```

```

test      -----
          4,5,6,7,8,9,10,11      5.0%  32392

          Running

Free CPUs      : 12,13,14,15
Host Isolcpu(s): 2-15
Emulator Pins  : 2-3

```


show system storage (JDM)

Syntax	<code>show system storage</code>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display the JDM storage details such as storage size, used space, and available space.
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170
Output Fields	Table 24 on page 185 describes the output fields for the show system storage command. Output fields are listed in the approximate order in which they appear.

Table 24: show system storage Output Fields

Field Name	Field Description
Host Storage Information	Displays the following host storage information: <ul style="list-style-type: none"> • Device—Host storage file system. • Size—Total storage size. • Used—Storage used. • Available—Storage available for use. • Use—Percentage of the storage being used. • Mount Point—Directory in which the host storage file system is mounted.
JDM Storage Information	Displays the following JDM storage information: <ul style="list-style-type: none"> • Directories—Names of the directories. • Used—Storage used.
VNF Storage Information	Displays the following guest network function (GNF) storage information: <ul style="list-style-type: none"> • Directories—Names of the directories. • Size—Total storage size. • Used—Storage used.

Sample Output

show system storage

```
user@jdm> show system storage
```

Host Storage Information

Device	Size	Used	Available	Use	Mount Point
/dev/mapper/rhel_mgb--dvaita--ixr0	493G	18G	451G	4%	/
/dev/sda2	976M	137M	772M	16%	/boot
/dev/sda1	200M	9.5M	191M	5%	/boot/efi
/dev/mapper/rhel_mgb--dvaita--ixr0	128G	40G	83G	33%	/var
/dev/mapper/rhel_mgb--dvaita--ixr0	99G	5.1G	89G	6%	/home

JDM Storage Information

Directories	Used
/vm-primary/	8.0G
/var/third-party/	61M
/var/jdm-usr/	1.7G
/juniper	1.1G

VNF Storage Information

Directories	Size	Used
/vm-primary/test	50.3G	4.5G
/vm-primary/test123	16.8G	3.5G

show system memory (JDM)

Syntax	<code>show system memory</code>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display the memory usage information about the host server, Juniper Device Manager (JDM), and guest network functions (GNF).
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170
Output Fields	Table 25 on page 187 describes the output fields for the show system memory command. Output fields are listed in the approximate order in which they appear.

Table 25: show system memory Output Fields

Field Name	Field Description
Memory Usage Information	Displays the following memory usage information about host server and JDM: <ul style="list-style-type: none"> • Total—Total memory. • Used—Used memory. • Free—Available memory.
VNF Memory Information	Displays the following memory information about the GNFs: <ul style="list-style-type: none"> • Name—GNF name. • Resident—The memory used by the GNFs. • Actual—Actual memory.

Sample Output

show system memory

```
user@jdm> show system memory
```

```
Memory Usage Information
-----
      Total  Used   Free
-----
Host:  251G   14G   219G
JDM :   2.0G   2.0G    0
```

VNF Memory Information

Name	Actual	Resident
gnf1	16.0G	13.6G
gnf2	16.0G	11.6G
gnf3	16.0G	12.6G
gnf4	16.0G	12.0G
gnf5	16.0G	12.7G

show system network (JDM)

Syntax	<code>show system network</code>
Release Information	Command introduced in Junos OS Release 17.4R1.
Description	Display the statistics information for physical interface, JDM interface, and interfaces per guest network function (GNF).
Required Privilege Level	View
Related Documentation	<ul style="list-style-type: none"> • Generic Guidelines for Using JDM Server Commands on page 163 • request virtual-network-functions on page 171 • request server authenticate-peer-server on page 170
Output Fields	Table 26 on page 189 describes the output fields for the <code>show system network</code> command. Output fields are listed in the approximate order in which they appear.

Table 26: show system network Output Fields

Field Name	Field Description
Physical Interfaces	
Name	Name of the physical interface.
Index	Interface index.
MTU	The maximum transmission unit.
Hardware-address	The physical interface hardware address.
Rcvd Bytes	The number of bytes received.
Rcvd Packets	The number of packets received.
Rcvd Error	The packets received with error.
Rcvd Drop	The packets dropped during reception.
Trxd Bytes	The number of bytes transmitted.
Trxd Packets	The number of packets transmitted.
Trxd Error	The packets transmitted with error.

Table 26: show system network Output Fields (continued)

Field Name	Field Description
Trxd Drop	The packets dropped during transmission.
Flags	Interface flags.
Per VNF Interface Statistics	
Name	List of the interfaces for each GNF.
Source	Interface source.
MAC Address	The MAC address of the interface.
Rcvd Bytes	The number of bytes received.
Rcvd Packets	The number of packets received.
Rcvd Error	The packets received with error.
Rcvd Drop	The packets dropped during reception.
Trxd Bytes	The number of bytes transmitted.
Trxd Packets	The number of packets transmitted.
Trxd Error	The packets transmitted with error.
Trxd Drop	The packets dropped during transmission.
JDM Interface Statistics	
Name	Name of the JDM interface.
Index	Interface index.
MTU	The maximum transmission unit.
Hardware-address	The JDM interface hardware address.
Rcvd Bytes	The number of bytes received.
Rcvd Packets	The number of packets received.
Rcvd Error	The packets received with error.
Rcvd Drop	The packets dropped during reception.
Trxd Bytes	The number of bytes transmitted.

Table 26: show system network Output Fields (continued)

Field Name	Field Description
Trxd Packets	The number of packets transmitted.
Trxd Error	The packets transmitted with error.
Trxd Drop	The packets dropped during transmission.
Flags	Interface flags.
VNF MAC Address Pool	
Start MAC Address	MAC address start value.
Range	Available MAC address range.

Sample Output

show system network

```
user@jdm> show system network
```

Physical Interfaces

Name	Index	MTU	Hardware-address	Rcvd Bytes	Rcvd Packets	Rcvd Error
Rcvd Drop	Trxd	Bytes	Trxd Packets	Trxd Error	Trxd Drop	Flags
-----	-----	-----	-----	-----	-----	-----
enp3s0f1	4	1500	00:25:90:b5:75:51	8787662837	51975964	0
538926	40009223		407379	0	0	BMPRU
ens3f1	7	1500	3c:fd:fe:08:87:02	1019880532	16723722	0
11243028	19265494115		31971968	0	0	BMPRU
ens3f0	3	1500	3c:fd:fe:08:87:00	5951717054	81330473	0
11226877	139135292735		124708008	0	0	BMPRU
enp3s0f2	5	1500	00:25:90:b5:75:52	3343179197	40806691	0
461955	3449064446		12191724	0	0	BMRU

Per VNF Interface Statistics

Interface	Source	MAC Address	Rcvd Bytes	Rcvd
packets Rcvd Error Rcvd Drop Trxd bytes	Trxd Packets	Trxd Error Trxd Drop		
-----	-----	-----	-----	-----
VNF name: test				
macvtap0	ens3f0	00:f1:60:3d:20:23	91526	1697
0	1254382	16782	0	0
macvtap1	ens3f1	00:f1:60:3d:20:24	561558	3727
0	803396	12958	0	0
vnet1	bridge_jdm_vm	00:f1:60:3d:20:25	2888	32
0	2282	25	0	0
macvtap2	enp3s0f2	00:f1:60:3d:20:26	152563	1080
0	4472700	55664	0	0

JDM Interface Statistics

Name	Index	MTU	Hardware-address	Rcvd Bytes	Rcvd Packets	Rcvd Error	Rcvd
Drop	Trxd Bytes		Trxd Packets	Trxd Error	Trxd Drop	Flags	

bme1	1433	1500	52:54:00:21:20:2e	502730	4506	0	0
	477328		2619	0	0	BMRU	
jmgmt0	1439	1500	00:f1:60:3d:20:22	4991675	66429	0	2862
	100548		891	0	0	BMRU	
bme2	1435	1500	52:54:00:88:b5:dd	2930	33	0	0
	3466		39	0	0	ABMRU	
cb0.4002	2	1500	00:f1:60:3d:20:20	12204921	209269	0	0
	3688591023		195579	0	0	ABMRU	
cb1.4002	3	1500	00:f1:60:3d:20:21	161850	3026	0	0
	204784		3029	0	0	ABMRU	

VNF MAC Address Pool

Start MAC Address: 00:f1:60:3d:20:20

Range: 96

.....

restart (JDM)

Syntax restart (gracefully | immediately | soft)
 <jdmd>
 <jdmmon>
 <jinventoryd>
 <jlinkmon>

Release Information Command introduced in Junos OS Release 17.4R1.

Description Restart daemons belonging to Juniper Device Manager (JDM).



CAUTION: Never restart a software process unless instructed to do so by a customer support engineer.

Options **gracefully**—(Optional) Restart a specified module or component by sending the equivalent of a UNIX SIGTERM signal.

immediately—(Optional) Immediately restart a module or component by sending the equivalent of a UNIX SIGKILL signal.

soft—(Optional) Reread and reactivate the configuration without completely restarting a module or component. This option is the equivalent of a UNIX SIGHUP signal.

jdmd—Restart the JDM service process.

jdmmon—Restart the JDM link monitoring daemon.

jinventoryd—Restart the JDM inventory management daemon.

jlinkmon—Restart the JDM link monitor daemon, which runs on the Linux host.



NOTE: The options **gracefully**, **immediately**, and **soft** are not available for restarting the Juniper link monitor daemon.

Related Documentation

- [Generic Guidelines for Using JDM Server Commands on page 163](#)
- [request virtual-network-functions on page 171](#)
- [request server authenticate-peer-server on page 170](#)

List of Sample Output [restart jdmd gracefully on page 194](#)

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

restart jdmd gracefully

```
user@jdm> restart jdmd gracefully
```

```
Juniper Device Manager service process started, pid 12058
```