JUNIPER NETWORKS | Engineering Simplicity

Junos OS

Junos Node Slicing User Guide

Published
2021-12-15

JUNOS

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 User Guide
Copyright © 2021 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 This Guide | ix

1 **Junos Node Slicing Overview**

Understanding Junos Node Slicing | 2

- Junos Node Slicing Overview | **2**
- Components of Junos Node Slicing | **4**
- Abstracted Fabric Interface | **8**
- Optimizing Fabric Path for Abstracted Fabric Interface | **13**
- Choosing Between External Server Model and In-Chassis Model | **14**
- Primary-role Behavior of BSYS and GNF | **14**
- Junos Node Slicing Administrator Roles | **16**
- Sub Line Card Overview | **16**
- Multiversion Software Interoperability Overview | **19**
- Next Gen Services on Junos node slicing | **20**
- Comparing Junos Node Slicing with Logical Systems | **20**
- Licensing for Junos Node Slicing | **21**

2 **Setting Up Junos Node Slicing**

Minimum Hardware and Software Requirements for Junos Node Slicing | 23

Preparing for Junos Node Slicing Setup | 29

- Connecting the Servers and the Router | **29**
- x86 Server CPU BIOS Settings | **32**
- x86 Server Linux GRUB Configuration | **34**
- Updating Intel X710 NIC Driver for x86 Servers | **36**
- Installing Additional Packages for JDM | **37**
- Completing the Connection Between the Servers and the Router | **38**

Setting Up Junos Node Slicing | 39

Configuring an MX Series Router to Operate in BSYS Mode (External Server Model) | **40**

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) | **44**

Configuring Non-Root Users in JDM (Junos Node Slicing) | **45**

Configuring JDM interfaces (External Server Model) | **47**

Configuring MX Series Router to Operate in In-Chassis Mode | **50**

Installing and Configuring JDM for In-Chassis Model | **51**

> Installing JDM RPM Package on MX Series Router (In-Chassis Model) | **52**
>
> Configuring JDM (In-Chassis Model) | **53**

Assigning MAC Addresses to GNF | **57**

Configuring Guest Network Functions | **58**

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | **61**

Optimizing Fabric Path for Abstracted Fabric Interface | **64**

SNMP Trap Support: Configuring NMS Server (External Server Model) | **65**

Chassis Configuration Hierarchy at BSYS and GNF | **68**

Configuring Sub Line Cards and Assigning Them to GNFs | **68**

Sample Configuration for Junos Node Slicing | **71**

Sample Configuration for Sub Line Cards | **80**

3 **Upgrading and Managing Junos Node Slicing**

Junos Node Slicing Upgrade | 84

Upgrading Junos Node Slicing | **84**

Downgrading JDM for External Server Model | **89**

Downgrading JDM for In-Chassis Model | **92**

Unified ISSU Support | **93**

Managing Multiversion Software Interoperability | **94**

Viewing Software Incompatibility Alarms | **96**

Viewing Incompatibilities Between Software Versions | **97**

Restarting External Servers | **97**

Updating Host OS on the External Servers | **99**

Applying Security Updates to Host OS | **99**

Applying Security Patches for Ubuntu Container | **101**

Managing Junos Node Slicing | 103

Deleting Guest Network Functions | **103**

Disabling Junos Node Slicing | **104**

Managing Sub Line Cards | **106**

4 **Configuration Statements for BSYS**

af-name | 114

control-plane-bandwidth-percent (Node Slicing) | 116

description (GNF) | 117

description (Abstracted Fabric) | 119

fpc-slice | 121

fpcs (Node Slicing) | 123

network-slices | 125

gnf | 127

collapsed-forward | 129

guest-network-functions | 131

peer-gnf | 133

power | 134

vlan-isolation | 136

vmhost resize vjunos compact | 138

5 Operational Commands for BSYS

request chassis fpc slc | 141

request vmhost jdm add (In-Chassis Model) | 142

request vmhost jdm delete (In-Chassis Model) | 144

request vmhost jdm start (In-Chassis Model) | 145

request vmhost jdm login (In-Chassis Model) | 146

request vmhost jdm stop (In-Chassis Model) | 148

show vmhost status (In-Chassis Model) | 149

show vmhost jdm status (In-Chassis Model) | 151

show chassis network-slices | 153

show chassis fpc pic-status (Node Slicing) | 158

show chassis fpc (Node Slicing) | 160

show chassis adc (Node Slicing) | 164

show chassis network-slices fpcs | 167

show system anomalies gnf-id | 169

6 Operational Commands for GNF

show chassis gnf | 174

show chassis gnf | 177

show chassis hardware (GNF) | 180

show chassis fpc (GNF) | 183

show chassis fpc pic-status (GNF) | 186

show chassis adc (GNF) | 188

show interfaces (Abstracted Fabric) | 190

show system anomalies (GNF) | 220

7 Configuration Statements for JDM

virtual-network-functions | 225

server | 227

interfaces (Junos Node Slicing) | 229

routing-options (Junos Node Slicing) | 230

system login (Junos Node Slicing) | 232

root-login (JDM) | 234

vnf-license-supplement | 236

8 Operational Commands for JDM

Generic Guidelines for Using JDM Server Commands | 239

clear log (JDM) | 239

monitor list (JDM) | 241

monitor start (JDM) | 242

monitor stop (JDM) | 245

request server authenticate-peer-server | 247

request virtual-network-functions | 249

show virtual-network-functions | 251

show version vnf | 261

show version (JDM) | 264

show system cpu (JDM) | 267

show system mac-addresses (JDM) | 270

show system max-smbios-version (JDM) | 273

show system memory (JDM) | 274

show system network (JDM) | 277

show system max-smbios-version (JDM) | 283

show system random-mac-prefix (JDM) | 284

restart (JDM) | 286

About This Guide

Use this guide to set up, configure and manage Junos Node Slicing. This guide contains procedures such as installing the required software packages, configuring the JDM and server interfaces, configuring the BSYS mode, creating GNFs, and configuring abstracted fabric interfaces. It also has the configuration statements and command summaries used for Junos Node Slicing.

1

CHAPTER

Junos Node Slicing Overview

Understanding Junos Node Slicing | 2

Understanding Junos Node Slicing

IN THIS SECTION

- Junos Node Slicing Overview | 2
- Components of Junos Node Slicing | 4
- Abstracted Fabric Interface | 8
- Optimizing Fabric Path for Abstracted Fabric Interface | 13
- Choosing Between External Server Model and In-Chassis Model | 14
- Primary-role Behavior of BSYS and GNF | 14
- Junos Node Slicing Administrator Roles | 16
- Sub Line Card Overview | 16
- Multiversion Software Interoperability Overview | 19
- Next Gen Services on Junos node slicing | 20
- Comparing Junos Node Slicing with Logical Systems | 20
- Licensing for Junos Node Slicing | 21

Junos Node Slicing Overview

IN THIS SECTION

- Benefits of Junos Node Slicing | 3

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 in hosting multiple services on a single physical infrastructure while avoiding the operational complexity involved. It also maintains operational, functional, and administrative separation of the functions hosted on the device.

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

IN THIS SECTION

- Base System (BSYS) | 6
- Guest Network Function (GNF) | 6
- Juniper Device Manager (JDM) | 7

Junos node slicing enables you to partition a single MX Series router to make it 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. Additionally, depending on the linecard type, you can even assign sets of PFEs within a linecard to different GNFs. See "Sub Line Card Overview" on page 16 for details.

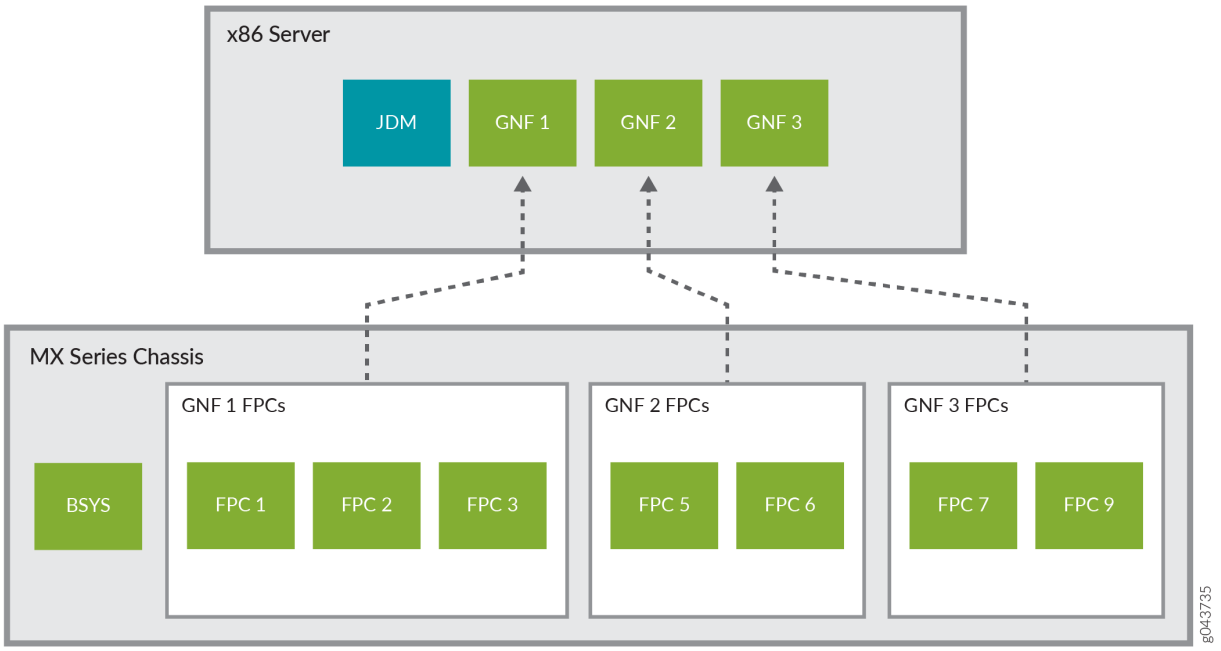
Junos node slicing supports two models:

- 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 5 shows three GNFs with their dedicated line cards running on an external server.

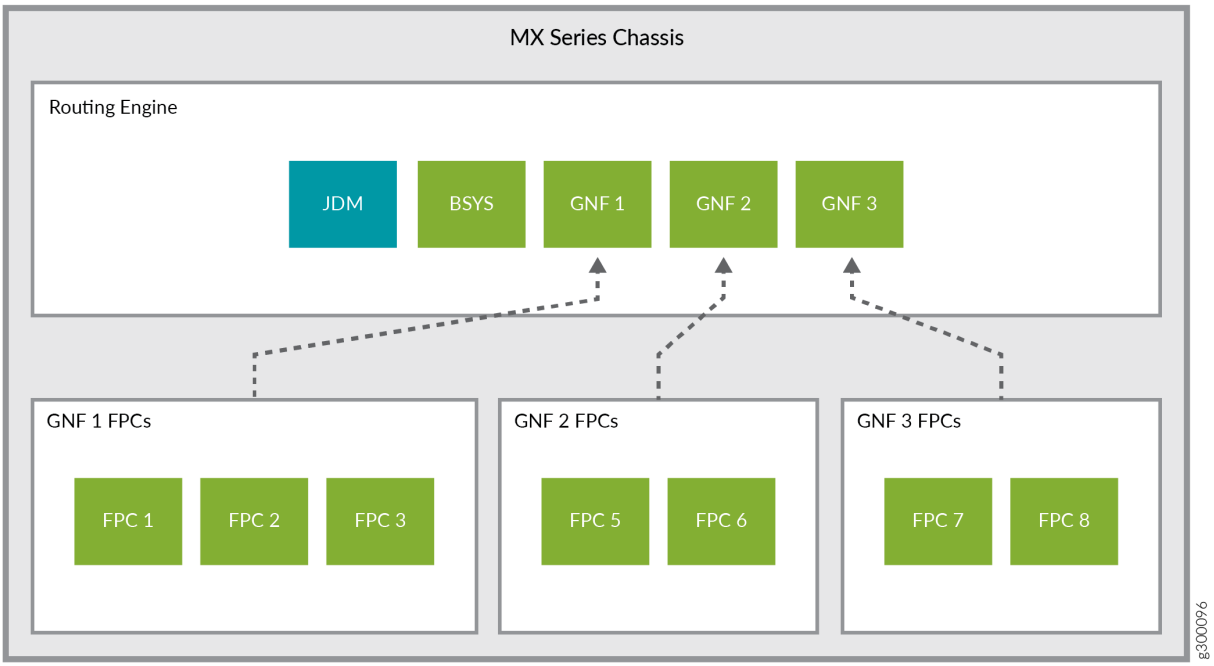
Figure 1: GNFs on External Server



See "Connecting the Servers and the Router" on page 29 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 6.

Figure 2: In-chassis Junos Node Slicing



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 58). 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 23.

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 | 2

Primary-role Behavior of BSYS and GNF | 14

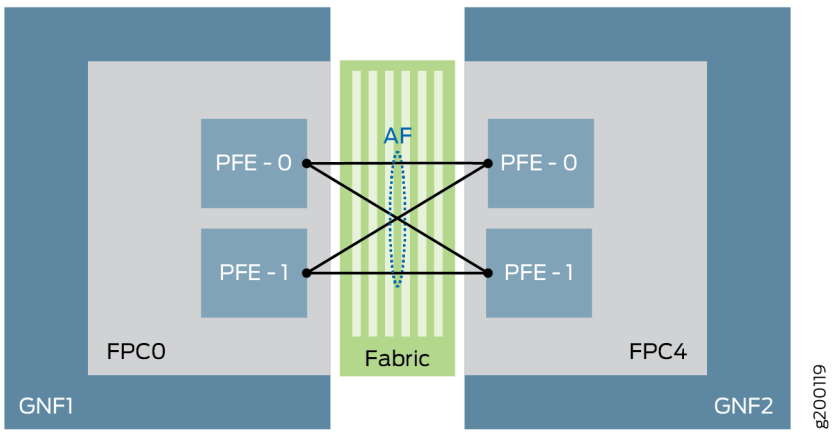
Abstracted Fabric Interface

IN THIS SECTION

- Understanding Abstracted Fabric Interface Bandwidth | 9
- Features Supported on Abstracted Fabric Interfaces | 9
- Abstracted Fabric Interface Restrictions | 12

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 8.

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 4x240 Gbps = 960 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 1 on page 11.

Features Supported on Abstracted Fabric Interfaces

Abstracted fabric interfaces support the following features:

- Unified in-service software upgrade (ISSU).

  NOTE: Abstracted fabric interfaces do not support ISSU on MPC11E line cards.

- Hyper mode configuration at the BSYS level (starting in Junos OS Release 19.3R2). This feature is supported on MPC6E, MPC8E, MPC9E, and MPC11E line cards.

  NOTE:

  - You cannot have different hyper mode configurations for individual GNFs as they inherit the configuration from the BSYS.
  - The MX2020 and MX2010 routers with SFB3 come up in hyper mode by default. If you require hyper mode to be disabled at any GNF, you must configure it at the BSYS, and it will apply to all GNFs of that chassis.

- 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

  NOTE: The MPC11E line card does not support abstracted fabric interface fabric-stats sensor.

- 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. This support is not available for the MPC11E line card.
- With the af interface configuration, GNFs support af-capable MPCs. Table 1 on page 11 lists the af-capable MPCs, the number of PFEs supported per MPC, and the bandwidth supported per MPC.

Table 1: 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
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)

Table 1: Supported Abstracted Fabric-capable MPCs *(Continued)*

MPC	Initial Release	Number of PFEs	Total Bandwidth
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)
MX2K-MPC11E	19.3R2	8	4T (500G*8)

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.
- 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 | 61

Optimizing Fabric Path for Abstracted Fabric Interface

You can optimize the traffic flowing over the abstracted fabric (af) interfaces between two guest network functions (GNFs), by configuring a fabric path optimization mode. This feature reduces fabric bandwidth consumption by preventing any additional fabric hop (switching of traffic flows from one Packet Forwarding Engine to another) before the packets eventually reach the destination Packet Forwarding Engine. Fabric path optimization, supported on MX2008, MX2010, and MX2020 with MPC9E and MX2K-MPC11E, prevents only a single additional traffic hop that results from abstracted fabric interface load balancing.

You can configure one of the following fabric path optimization modes:

- `monitor`—If you configure this mode, the peer GNF monitors the traffic flow and sends information to the source GNF about the Packet Forwarding Engine to which the traffic is being forwarded currently and the desired Packet Forwarding Engine that could provide an optimized traffic path. In this mode, the source GNF does not forward the traffic towards the desired Packet Forwarding Engine.
- `optimize`—If you configure this mode, the peer GNF monitors the traffic flow and sends information to the source GNF about the Packet Forwarding Engine to which the traffic is being forwarded currently and the desired Packet Forwarding Engine that could provide an optimized traffic path. The source GNF then forwards the traffic towards the desired Packet Forwarding Engine.

To configure a fabric path optimization mode, use the following CLI commands at BSYS.

```
user@router# set chassis network-slices guest-network-functions gnf id af-name collapsed-forward (monitor
| optimize)
user@router# commit
```

After configuring fabric path optimization, you can use the command `show interfaces af-interface-name` in GNF to view the number of packets that are currently flowing on the optimal / non-optimal path.

SEE ALSO

collapsed-forward | 129

show interfaces (Abstracted Fabric) | 190

Choosing Between External Server Model and In-Chassis Model

The external server model allows you to configure more instances of GNFs with higher scale, since you can choose a server of sufficient capacity to match GNF requirements. With the in-chassis model, the number of GNFs that can be configured is a function of the scale requirements of the constituent GNFs and the overall capacity of the Routing Engine.

The external server and in-chassis models of Junos node slicing are mutually exclusive. An MX Series router can be configured to operate in only one of these models at one time.

Primary-role Behavior of BSYS and GNF

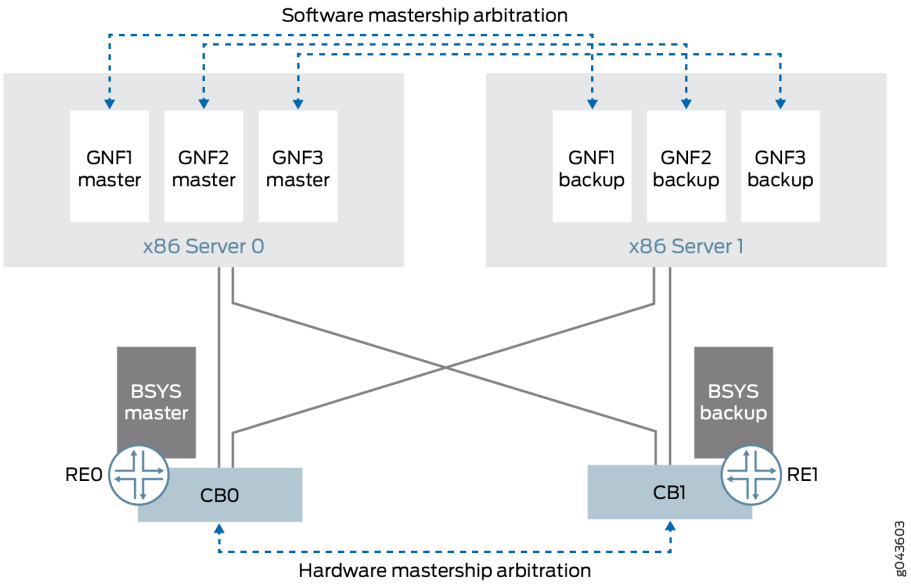
IN THIS SECTION

- BSYS Primary Role | **15**
- GNF Primary Role | **15**

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

Figure 4 on page 15 shows the primary-role behavior of GNF and BSYS with Routing Engine redundancy.

Figure 4: Primary-role Behavior of GNF and BSYS (External Server Model)



BSYS Primary Role

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

GNF Primary Role

The GNF VM primary-role arbitration behavior is similar to that of MX Series Routing Engines. Each GNF runs as a primary-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 primary role is independent of the BSYS primary role and that of other GNFs. The GNF primary role arbitration is done through Junos OS. Under connectivity failure conditions, GNF primary role is handled conservatively.

The GNF primary-role 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 primary role when the primary 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.

Sub Line Card Overview

IN THIS SECTION

- Line Card Resources for SLCs | **17**
- MPC11E Line Card Resources for SLCs | **18**

In Junos node slicing, each GNF comprises a set of line cards (FPCs). By default, the finest granularity provided by a GNF is at the line card level, because each GNF is assigned whole line cards (that is, the complete set of Packet Forwarding Engines in each line card). With the sub line card (SLC) feature, you can define even finer granularity of partitioning, by assigning subsets of Packet Forwarding Engines in a single line card to different GNFs.

Such user-defined subsets of Packet Forwarding Engines in a line card are referred to as sub line cards (SLCs). Operationally, SLCs function like independent line cards.

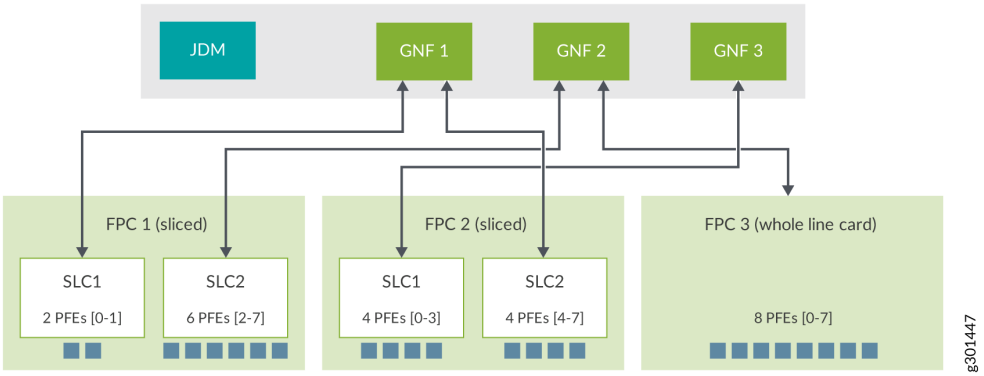
When you slice a line card, every SLC of that line card must be assigned to a different GNF.

You can assign SLCs from multiple line cards to the same GNF.

In a Junos node slicing setup with the SLC feature, a GNF can comprise a set of whole line cards as well as a set of slices (SLCs) of line cards, providing a higher level of flexibility.

When a line card is sliced, two types of software instances run on that line card - a single base line card (BLC) instance and multiple SLC instances (as many as the number of slices of that line card). Currently, the SLC capability is available only on the MPC11E, which supports two SLCs. The BLC instance is responsible for managing hardware common to all SLCs of that line card, while each SLC instance is responsible for managing the set of Packet Forwarding Engines exclusively assigned to it. The BLC instance runs the Junos software of the BSYS, while each SLC instance runs the Junos software of its associated GNF.

Figure 5: SLCs assigned to GNFs in an external server-based Junos node slicing setup



SLCs support abstracted fabric interface and collapsed forwarding (see Optimizing Fabric Path for Abstracted Fabric Interface). You can use the `show interface` *`af-interface-name`* command to view the load balance statistics of the remote FPC slice-specific Packet Forwarding Engines. See show interfaces (Abstracted Fabric) for details.

The SLC capability is available only on the MPC11E (model number: MX2K-MPC11E).

Line Card Resources for SLCs

An SLC or a slice of a line card defines the set of Packet Forwarding Engines (of that line card) that must operate together. Packet Forwarding Engines in a line card are identified by numeric IDs. If a line card has ‘n’ Packet Forwarding Engines, the individual Packet Forwarding Engines are numbered 0 to (n-1). In addition, CPU cores and DRAM on the control board of the line card must also be divided and allocated

to the slice. To define an SLC, then, is to define the following line card resources to be dedicated to that SLC:

- A Packet Forwarding Engine range
- The number of CPU cores on the control board of the line card
- The size of DRAM (in GB) on the control board of the line card

NOTE: A certain amount of the DRAM is automatically reserved for the BLC instance on that line card, and the remainder is available for SLC instances.

Every SLC is identified by a numeric ID, assigned by the user.

When a line card is sliced, the resource partitions for every slice on that line card must be completely defined.

MPC11E Line Card Resources for SLCs

An MPC11E line card has:

- 8 Packet Forwarding Engines
- 8 CPU cores on the control board
- 32 GB of DRAM on the control board

5 GB of DRAM is automatically reserved for BLC use, 1 GB of DRAM is allocated to the line card host, and the remaining 26 GB is available for SLC slices.

An MPC11E is capable of supporting two SLCs.

The Table 2 on page 19 defines two types of resource allocation profiles supported by an MPC11E for the two SLCs, referred to here as SLC1 and SLC2.

In the symmetric profile, the Packet Forwarding Engines and other line card resources are distributed evenly between the slices. In the asymmetric profile, only the specified line card resource combinations shown in Table 2 on page 19 are supported.

NOTE: You can configure the following SLC profiles, based on how the Packet Forwarding Engines [0-7] are split between the two SLCs:

- Packet Forwarding Engines 0-3 for one SLC, and 4-7 for the other SLC (symmetric profile)

- Packet Forwarding Engines 0-1 for one SLC, and 2-7 for the other SLC (asymmetric profile)
- Packet Forwarding Engines 0-5 for one SLC and 6-7 for the other SLC (asymmetric profile)

In the asymmetric profile, you can assign either 9 GB or 17 GB of DRAM to an SLC. Since all the line card resources must be fully assigned, and the total DRAM available for SLCs is 26 GB, assigning 9 GB of DRAM to an SLC requires that the remaining 17 GB must be assigned to the other SLC.

Table 2: SLC Profiles Supported by MPC11E

	Symmetric Profile		Asymmetric Profile	
Resource Type	SLC1	SLC2	SLC1	SLC2
Packet Forwarding Engine	4	4	2	6
DRAM	13 GB	13 GB	17 GB/9 GB	9 GB/17 GB
CPU	4	4	4	4

See also: Configuring Sub Line Cards and Assigning Them to GNFs and Managing Sub Line Cards.

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.

Next Gen Services on Junos node slicing

Junos node slicing supports MX-SPC3 Services Card, a security services card that provides additional processing power to run the Next Gen Services on the MX platforms. You can enable Next Gen Services at guest network function (GNF), by using the CLI `request system enable unified-services` at GNF. To support an MX-SPC3, a GNF must have a line card associated with it.

In a Junos node slicing setup, you can use both MX-SPC3 and MS-MPC on the same chassis but on different GNF Routing Engines. If you have enabled Next Gen Services at GNF, by using `request system enable unified-services`, the MX-SPC3 comes online. If you have not enabled Next Gen Services, the MS-MPC comes online.

The software installation or upgrade of an MX-SPC3 card happens when you install or upgrade the associated GNF Routing Engine.

> **NOTE**: The MX-SPC3 does not support abstracted fabric interfaces. Therefore, a GNF that has an MX-SPC3 card linked to it must also have a line card associated with it.

Comparing Junos Node Slicing with Logical Systems

Junos node slicing is a layer below logical systems in Junos. Both technologies have some overlapping capabilities but differ in other aspects. With Junos node slicing, complete line cards, and therefore, physical interfaces, are assigned to a GNF, while with logical systems, a single physical interface itself can be shared across different logical systems, since multiple logical interfaces defined over a physical interface can all be assigned to separate logical systems. This means, logical systems allow finer granularity of sharing than Junos node slicing. But all logical systems share a single Junos kernel, thus necessarily running the same Junos version, besides having to share the Routing Engine and line card physical resources such as CPU, memory and storage. With Junos node slicing, each GNF gets its own equivalent of a pair of Routing Engines, as also line cards dedicated to that GNF, so the GNFs do not share most physical resources – they only share the chassis and switch fabric. GNFs, unlike logical systems, can be independently upgraded and administered like a MX standalone router.

Junos node slicing is a technology that complements, and even augments logical systems, since a GNF can itself have multiple logical systems within it. Where physical isolation, guaranteed resources and complete administrative isolation is paramount, Junos node slicing would be a better match. And where fine granularity of sharing, down to the logical interface level, is paramount, a logical system would be the better match.

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 | 84

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | 61

2

CHAPTER

Setting Up Junos Node Slicing

Minimum Hardware and Software Requirements for Junos Node Slicing | 23

Preparing for Junos Node Slicing Setup | 29

Setting Up Junos Node Slicing | 39

Minimum Hardware and Software Requirements for Junos Node Slicing

IN THIS SECTION

- MX Series Router | 23
- x86 Servers (External Server Model) | 24

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-1800X4-32G, RE-S-1800X4-16G, 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)
  - REMX2008-X8-128G (used in MX2008 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.

NOTE: We recommend that you:

- use hardware RAID 1 configuration for storage resiliency.
- set up /vm-primary as a Linux partition.
- do not use software RAID.

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 3 on page 25 lists the GNF resource templates available for configuring Junos node slicing on external servers:

Table 3: 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 4 on page 26 lists the GNF resource templates available for configuring in-chassis Junos node slicing:

Table 4: 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 5 on page 26 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 5: Shared Server Resources Requirements (External Server Model)

Component	Specification
CPU	• Four cores to be allocated for JDM and Linux host processing.

Table 5: Shared Server Resources Requirements (External Server Model) ***(Continued)***

Component	Specification
Memory	• Minimum of 32 GB DRAM for JDM and Linux host processing.
Storage	• Minimum of 64 GB storage for JDM and Linux host.

Network Ports	• Two 10-Gbps Ethernet interfaces for control plane connection between the server and the router.
	  • 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 36 and "Installing Additional Packages for JDM" on page 37 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 36.

The servers must also have the BIOS setup as described in "x86 Server CPU BIOS Settings" on page 32 and the Linux GRUB configuration as described in "x86 Server Linux GRUB Configuration" on page 34.

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 | 4

Connecting the Servers and the Router | 29

Preparing for Junos Node Slicing Setup

IN THIS SECTION

- Connecting the Servers and the Router | 29
- x86 Server CPU BIOS Settings | 32
- x86 Server Linux GRUB Configuration | 34
- Updating Intel X710 NIC Driver for x86 Servers | 36
- Installing Additional Packages for JDM | 37
- Completing the Connection Between the Servers and the Router | 38

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 50.

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

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.

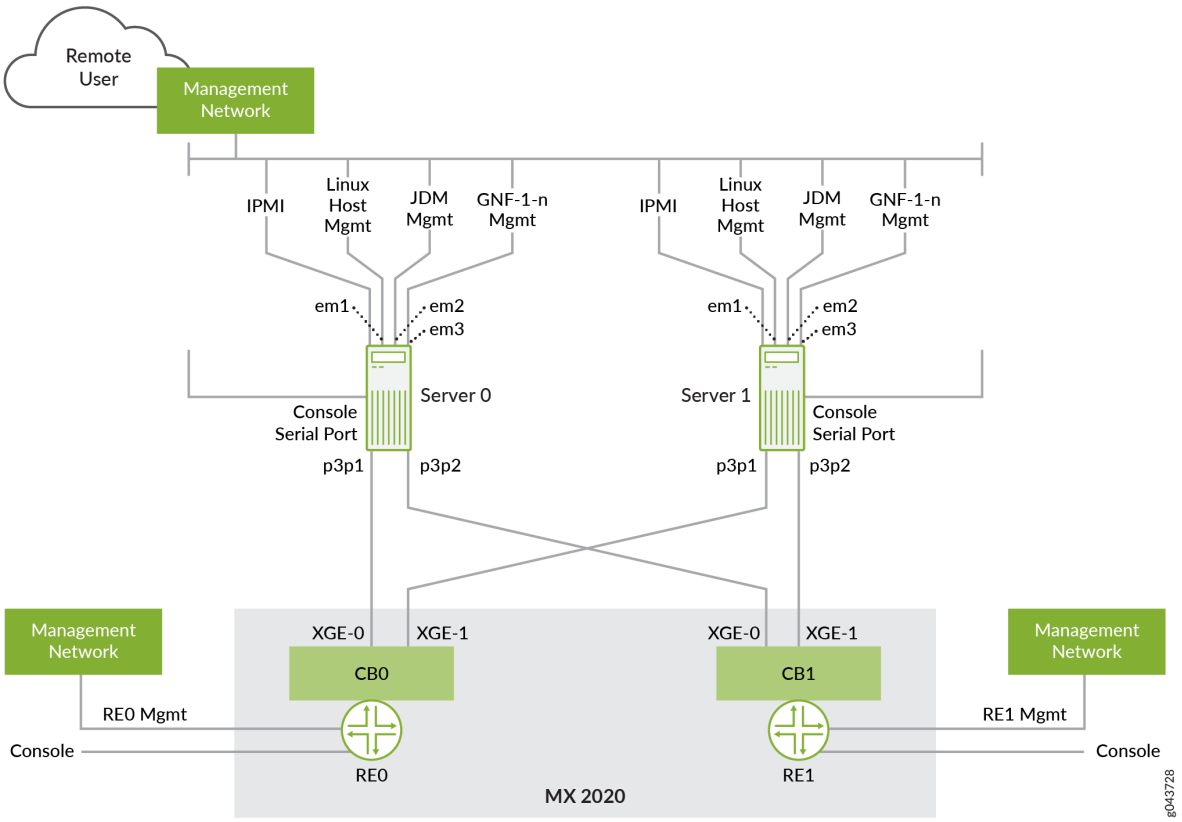
- To prevent the host server from any SSH brute force attack, we recommend that you add IPtables rules on the host server. The following is an example:

  ```
  iptables -N SSH_CONNECTIONS_LIMIT
  iptables -A INPUT -i jmgmt0 -p tcp -m tcp --dport 22 -m state --state NEW -j
  SSH_CONNECTIONS_LIMIT iptables -A SSH_CONNECTIONS_LIMIT -m recent --set --name SSH --
  rsource iptables -A SSH_CONNECTIONS_LIMIT -m recent --update --seconds 120 --hitcount 10 --
  name SSH --rsource -j DROP iptables -A SSH_CONNECTIONS_LIMIT -j ACCEPT
  ```

  The rule in the above example is used to rate-limit the incoming SSH connections. It allows you to block connections from the remote IP for a certain period of time when a particular number of SSH attempts are made. As per the example above, after 10 attempts, connections from remote IP will be blocked for 120 seconds.

Figure 6 on page 31 shows how an MX2020 router is connected to a pair of x86 external servers.

Figure 6: MX2020 Router—External x86 Server Connectivity



According to the example in Figure 6 on page 31, `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 6 on page 31, see:

- SCBE2-MX Description (for MX960 and MX480)

> **NOTE**: The XGE port numbers are not labeled on the SCBE2. On a vertically oriented SCBE2, the upper port is XGE-0 and the lower port is XGE-1. On a horizontally oriented SCBE2, the left port is XGE-0 and the right port is XGE-1.

- 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 32. 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=4-23' (isolate all CPU cores other than cores 0, 1, 2, and 3 for use by the GNF VMs). The `isolcpus` parameter is set to 'isolcpus=4-23' because of the following:

   - On each x86 server, JDM requires that all CPU cores other than CPU cores 0 and 1 be reserved for Junos node slicing.
   - CPU cores 2 and 3 are dedicated for GNF virtual disk and network I/O.

   NOTE: Previously, the `isolcpus` parameter 'isolcpus=2-23' was used. This has now been updated to 'isolcpus=4-23'. For more information, see KB35301.

   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=4-23”
```

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=4-23
```

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=4-23"
```

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 | 23

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*.
- 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

Minimum Hardware and Software Requirements for Junos Node Slicing | 23

Connecting the Servers and the Router

Sample Configuration for Junos Node Slicing

RELATED DOCUMENTATION

Junos Node Slicing Overview | 2

Components of Junos Node Slicing | 4

Minimum Hardware and Software Requirements for Junos Node Slicing | 23

Setting Up Junos Node Slicing

IN THIS SECTION

- Configuring an MX Series Router to Operate in BSYS Mode (External Server Model) | 40
- 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) | 44
- Configuring Non-Root Users in JDM (Junos Node Slicing) | 45
- Configuring JDM interfaces (External Server Model) | 47
- Configuring MX Series Router to Operate in In-Chassis Mode | 50
- Installing and Configuring JDM for In-Chassis Model | 51
- Assigning MAC Addresses to GNF | 57

- Configuring Guest Network Functions | 58
- Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | 61
- Optimizing Fabric Path for Abstracted Fabric Interface | 64
- SNMP Trap Support: Configuring NMS Server (External Server Model) | 65
- Chassis Configuration Hierarchy at BSYS and GNF | 68
- Configuring Sub Line Cards and Assigning Them to GNFs | 68
- Sample Configuration for Junos Node Slicing | 71
- Sample Configuration for Sub Line Cards | 80

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 29.

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 29.

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**.

2. 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
```

3. 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 37.

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

To download the package:

1. Go to the Juniper Support page.
2. Click **JDM** > ***Junos OS version number*** > **Juniper Device Manager *version number* (for Redhat)**.
3. 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.*13*.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 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
```

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 37.

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

To download the JDM Ubuntu package:

1. Go to the Juniper Support page.
2. Click **JDM** > ***Junos OS version number*** > **Juniper Device Manager *version number* (for Debian)**.
3. 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)*.
   - 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.

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 6 on page 46 contains the predefined login classes that JDM supports for non-root users:

Table 6: Predefined Login Classes

Login Class	Permissions
super-user	• Create, delete, start and stop GNFs. • Start and stop daemons inside the JDM. • Execute all CLIs. • Access the shell.
operator	• 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.

Table 6: Predefined Login Classes *(Continued)*

Login Class	Permissions
unauthorized	Ping and traceroute operations.

Configuring JDM interfaces (External Server Model)

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

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 29.

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 synchronize
```

Remember to configure commit synchronization as shown in the above step to ensure that the random MAC prefixes generated by the JDM instances are in sync. To check if the random MAC prefixes are in sync, use the show system random-mac-prefix command at JDM.

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.
- You must have done the ssh key exchange as described in the Step 3 before attempting the Step 4. If you attempt the Step 4 without completing the Step 3, the system displays an error message as shown in the following example:

  ```
  Failed to fetch JDM software version from server1. If authentication of peer server is not done
  yet, try running request server authenticate-peer-server.
  ```

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

```
root@jdm> show server connections

                                  Component              Interface              Status
Comments
                                  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 71.

If you want to modify the server interfaces configured in the JDM, you need to delete the GNFs (if they were configured), configure the interfaces as described above, reboot JDM from shell, reconfigure and activate the GNFs, and commit the changes,

Starting in Junos OS Release 19.2R1, Junos node slicing supports the assignment of a globally unique MAC address range (supplied by Juniper Networks) for GNFs. To know more, see *Assigning MAC Addresses to GNF*.

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)
  - REMX2008-X8-128G (used in MX2008 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

IN THIS SECTION

- Installing JDM RPM Package on MX Series Router (In-Chassis Model) | **52**
- Configuring JDM (In-Chassis Model) | **53**

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)

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*.

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
```

Starting in Junos OS 19.3R1, the JDM console does not display the message 'Domain JDM Started'. However, this message will be added to the system logs when the JDM is 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# commit synchronize
```

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.

- You must have done the ssh key exchange as described in the Step 7 before attempting the Step 8. If you attempt the Step 8 without completing the Step 7, the system displays an error message as shown in the following example:

  ```
  Failed to fetch JDM software version from server1. If authentication of peer server is not done yet,
  try running request server authenticate-peer-server.
  ```

Starting in Junos OS Release 19.2R1, Junos node slicing supports the assignment of a globally unique MAC address range (supplied by Juniper Networks) for GNFs. To know more, see *Assigning MAC Addresses to GNF*.

Assigning MAC Addresses to GNF

Starting in Junos OS Release 19.2R1, Junos node slicing supports the assignment of a globally unique MAC address range (supplied by Juniper Networks) for GNFs.

To receive the globally unique MAC address range for the GNFs, contact your Juniper Networks representative and provide your GNF license SSRN (Software Support Reference Number), which will have been shipped to you electronically upon your purchase of the GNF license. To locate the SSRN in your GNF license, refer to the Juniper Networks Knowledge Base article KB11364.

For each GNF license, you will then be provided an 'augmented SSRN', which includes the globally unique MAC address range assigned by Juniper Networks for that GNF license. You must then configure this augmented SSRN at the JDM CLI as follows:

```
root@jdm# set system vnf-license-supplement vnf-id gnf-id license-supplement-string augmented-ssrn-string
root@jdm# commit
```

NOTE:

- An augmented SSRN must be used for only one GNF ID. In the JDM, the GNF VMs are referred to as virtual network functions (VNFs). GNF ID is one of its attributes. Attributes of a VNF are fully described in the follow-on section *Configuring Guest Network Functions*.
- By default, the augmented SSRN will be validated. Should you ever need to skip this validation, you can use the no-validate attribute in the CLI as follows: Example: `set system vnf-license-supplement vnf-id gnf-id license-supplement-string augmented-ssrn-string [no-validate]`.

NOTE:

- You can configure the augmented SSRN for a GNF ID only when the GNF is not operational and has not yet been provisioned as well. You must first configure the augmented SSRN for a GNF ID before configuring the GNF.
- Ensure that the GNF ID for which the augmented SSRN is being configured has not already been provisioned. If the GNF ID is already provisioned, you must first delete the GNF for that GNF ID on both the servers (in case of the external server model) or on both the Routing Engines (in case of the in-chassis Junos node slicing model) before configuring the augmented SSRN.
- Analogously, you must first delete the GNF for a given GNF ID on both the servers (in case of the external server model) or on both the Routing Engines (in case of the in-chassis Junos node slicing model) before deleting the augmented SSRN for the GNF ID.
- You cannot apply an augmented SSRN to a GNF that is based on Junos OS 19.1R1 or older.
- To confirm that the assigned MAC address range for a GNF has been applied, when the GNF becomes operational, use the Junos CLI command `show chassis mac-addresses` - the output will match a substring of the augmented SSRN.

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 you have configured commit synchronization as part of JDM configuration so that the random MAC prefixes generated by the JDM instances are in sync. To check if the random MAC prefixes are in sync, use the show system random-mac-prefix command at JDM.
- 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`

root@test-jdm-server0# `set system vnf-license-supplement vnf-id 1 license-supplement-string RTU00023003204-01-AABBCCDDEE00-1100-01-411C`

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

Starting in Junos OS Release 19.2R1, Junos node slicing supports the assignment of a globally unique MAC address range (supplied by Juniper Networks) for GNFs. To know more, see *Assigning MAC Addresses to GNF*.

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`

NOTE: Remember to log out of the VNF console after your have completed your configuration tasks. We recommend that you set an idle time-out using the command `set system login idle-timeout` *minutes*. Otherwise, if a user forgets to log out of the VNF console session, another user can log in without providing the access credentials. For more information, see "system login (Junos Node Slicing)" on page 232.

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 71.
- 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.

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs

IN THIS SECTION

- Class of Service on Abstracted Fabric Interfaces | 63

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.

- In a Junos node slicing setup where each GNF is assigned with a single FPC, if the Packet Forwarding Engines of the FPC assigned to the remote GNF becomes unreachable over fabric, the associated abstracted fabric interface goes down. Examples of errors that could cause this behavior include pfe fabric reachability errors and cmerror events causing `pfe disable` action (use the `show chassis fpc errors` command for the details). If a GNF has multiple FPCs assigned to it, the local FPCs that report all peer Packet Forwarding Engines to be down, are removed from determining the abstracted fabric interface state.

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

1. 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 10.10.10.1/24
user@router-gnf-d# set interfaces af2 unit 0 family inet address 10.10.10.2/24
```

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 71.

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

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.

Optimizing Fabric Path for Abstracted Fabric Interface

You can optimize the traffic flowing over the abstracted fabric (af) interfaces between two guest network functions (GNFs), by configuring a fabric path optimization mode. This feature reduces fabric bandwidth consumption by preventing any additional fabric hop (switching of traffic flows from one Packet Forwarding Engine to another) before the packets eventually reach the destination Packet Forwarding Engine. Fabric path optimization, supported on MX2008, MX2010, and MX2020 with MPC9E and MX2K-MPC11E, prevents only a single additional traffic hop that results from abstracted fabric interface load balancing.

You can configure one of the following fabric path optimization modes:

- `monitor`—If you configure this mode, the peer GNF monitors the traffic flow and sends information to the source GNF about the Packet Forwarding Engine to which the traffic is being forwarded currently and the desired Packet Forwarding Engine that could provide an optimized traffic path. In this mode, the source GNF does not forward the traffic towards the desired Packet Forwarding Engine.
- `optimize`—If you configure this mode, the peer GNF monitors the traffic flow and sends information to the source GNF about the Packet Forwarding Engine to which the traffic is being forwarded currently and the desired Packet Forwarding Engine that could provide an optimized traffic path. The source GNF then forwards the traffic towards the desired Packet Forwarding Engine.

To configure a fabric path optimization mode, use the following CLI commands at BSYS.

```
user@router# set chassis network-slices guest-network-functions gnf id af-name collapsed-forward (monitor
| optimize)
user@router# commit
```

After configuring fabric path optimization, you can use the command `show interfaces af-interface-name` in GNF to view the number of packets that are currently flowing on the optimal / non-optimal path.

SEE ALSO

collapsed-forward | 129

show interfaces (Abstracted Fabric) | 190

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_LI
NK_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
```

JDM does not write any configuration to the host snmp configuration file (/etc/snmp/snmpd.conf). Hence, JDM installation and subsequent configuration do not have any impact on the host SNMP. The SNMP configuration CLI command in JDM is used only to configure the JDM's snmpd.conf file which is present within the container. To generate linkUp/Down trap, you must manually include the configuration as shown in the following example in the host server's snmpd.conf file (/etc/snmp/snmpd.conf):

```
createUser trapUser
iquerySecName trapUser
rouser trapUser
defaultMonitors yes
notificationEvent  linkUpTrap    linkUp   ifIndex ifAdminStatus ifOperStatus ifDescr
notificationEvent  linkDownTrap  linkDown ifIndex ifAdminStatus ifOperStatus ifDescr
monitor -r 10  -e linkUpTrap   "Generate linkUp" ifOperStatus != 2
monitor -r 10  -e linkDownTrap "Generate linkDown" ifOperStatus == 2
trap2sink <NMS-IP> host
```

In the above example, replace <NMS-IP> with the IP address of Network Management Station (NMS).

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 Sub Line Cards and Assigning Them to GNFs

For an overview of sub line cards, see "Sub Line Card Overview" on page 16.

NOTE:

- This feature is applicable to the MPC11E line card (model number: MX2K-MPC11E) on the MX2010 and MX2020 routers used in the external server-based Junos node slicing setup.
- Ensure that each Routing Engine of all GNFs and the BSYS run Junos OS Release 21.2R1 or later versions.

To slice an MPC11E further into sub line cards (SLCs), you must use the `fpc-slice` CLI option under the `set chassis network-slices guest-network-functions gnf` hierarchy in BSYS.

Before committing the configuration, you must configure all the SLCs supported by the line card and assign all the required resources such as core, DRAM and the Packet forwarding Engines to the SLCs. An MPC11E line card supports two SLCs.

GNFs support the following combinations of full line cards and SLCs:

- GNF with MPC11E SLCs
- GNF with MPC11E SLCs and MPC9
- GNF with MPC11E SLCs and MPC11E
- GNF with MPC11E SLCs, MPC9, MPC11E

To configure SLCs and assign them to GNFs, use the following steps:

NOTE:

- You must configure all the following CLI statements at once for all the SLCs (as shown in the steps below). Any modification to this configuration later causes the entire line card to reboot.
- If you configure any incorrect values (for example, unsupported Packet Forwarding Engine ranges, CPU cores, or DRAM values), the configuration commit fails with an appropriate message to indicate the error.

1. Configure SLCs.

```
root@bsys# set chassis network-slices guest-network-functions gnf 1 fpc-slice 2 slc 1
root@bsys# set chassis network-slices guest-network-functions gnf 2 fpc-slice 2 slc 2
```

NOTE: Do not assign:

- two or more SLCs of the same line card to the same GNF.
- the same SLC of a line card to more than one GNF.

2. Assign Packet Forwarding Engines to the SLCs. You must allocate all the Packet Forwarding Engines on the line card to the SLCs as shown in the following example:

```
root@bsys# set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 2 slc 1 pfe-id-list
[0-3]
root@bsys# set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 2 slc 2 pfe-id-list
[4-7]
```

NOTE: The configuration supports only the following Packet Forwarding Engine ranges:

- 0-3 for one SLC, and 4-7 for the other SLC (symmetric profile)
- 0-1 for one SLC, and 2-7 for the other SLC (asymmetric profile)
- 0-5 for one SLC and 6-7 for the other SLC (asymmetric profile)

3. Assign CPU cores to the SLCs as shown in the following example:

```
root@bsys# set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 2 slc 1 cores 4
root@bsys# set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 2 slc 2 cores 4
```

NOTE: 4 is the only value of CPU cores supported. You must configure the value 4 for each of the two SLCs.

4. Assign DRAMs to the SLCs as shown in the following example:

```
root@bsys# set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 2 slc 1 dram 13
root@bsys# set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 2 slc 2 dram 13
```

You must allocate a total DRAM of 26 GB for both the SLCs together. Only the following combinations of DRAM allocation are supported:

SLC1 DRAM (GB)	SLC2 DRAM (GB)	Sub Total (GB)	BLC/Linux Host DRAM (GB)	Total (GB)
13	13	26	6	32
9/17	17/9	26	6	32

NOTE: You cannot allocate resources to the BLC; they are automatically assigned by Junos OS.

5. Commit the changes.

```
root@bsys# commit
```

SEE ALSO

Managing Sub Line Cards | 106

Sample Configuration for Junos Node Slicing

IN THIS SECTION

- Sample JDM Configuration (External Server Model) | 72
- Sample JDM Configuration (In-Chassis Model) | 74
- Sample BSYS Configuration with Abstracted Fabric Interface | 75
- Sample Abstracted Fabric Configuration at GNF with Class of Service | 76
- Sample Output for Abstracted Fabric Interface State at a GNF | 79

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
```

Sample Configuration for Sub Line Cards

IN THIS SECTION

- Sample Configuration for Symmetric Sub Line Card Profile | 80
- Sample Configuration for Asymmetric Sub Line Card Profile | 81

This section provides sample configurations for sub line cards (SLCs).

Sample Configuration for Symmetric Sub Line Card Profile

In the symmetric profile, only one combination of resources is possible.

The following is a sample configuration to slice the FPC 1 (MPC11E) in symmetric sub line card profile:

```
set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 1 slc 1 pfe-id-list 0-3
set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 1 slc 1 cores 4
set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 1 slc 1 dram 13
set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 1 slc 2 pfe-id-list 4-7
set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 1 slc 2 cores 4
set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 1 slc 2 dram 13
```

This configuration would appear as shown below:

```
root@bsys> show chassis network-slices guest-network-functions
    gnf 1{
    fpc-slice {
        fpc 1{
            slc 1{
                pfe-id-list 0-3;
                cores 4;
```

```
                dram 13;
            }
        }
    }
}
gnf 2{
    fpc-slice {
        fpc 1{
            slc 2{
                pfe-id-list 4-7;
                cores 4;
                dram 13;
            }
        }
    }
}
```

Sample Configuration for Asymmetric Sub Line Card Profile

In the asymmetric profile, two configurations are possible, depending on how the PFEs or Packet Forwarding Engines [0-7] are split between the two SLCs. In one example configuration, the first two Packet Forwarding Engines [0-1] are assigned to one SLC, and the remaining Packet Forwarding Engines [2-7] to the other SLC. In the other example configuration, the last two Packet Forwarding Engines [6-7] are assigned to one SLC, and the remaining Packet Forwarding Engines [0-5] to the other SLC.

The sample configuration below is an example of [0-1 2-7] split.

In the example below, the CPU core and DRAM assignments for the SLCs match one of the columns under the ‘Asymmetric Profile’ resource combination as shown in the table **SLC Profiles Supported by MPC11E** on the "Sub Line Card Overview" on page 16 page.

```
set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 1 slc 1 pfe-id-list 0-1
set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 1 slc 1 cores 4
set chassis network-slices guest-network-functions gnf 1 fpc-slice fpc 1 slc 1 dram 17
set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 1 slc 2 pfe-id-list 2-7
set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 1 slc 2 cores 4
set chassis network-slices guest-network-functions gnf 2 fpc-slice fpc 1 slc 2 dram 9
```

This configuration would appear as below:

```
root@bsys> show chassis network-slices guest-network-functions
    gnf 1{
    fpc-slice {
        fpc 1{
            slc 1{
                pfe-id-list 0-1;
                cores 4;
                dram 17;
            }
        }
    }
}
gnf 2{
    fpc-slice {
        fpc 1{
            slc 2{
                pfe-id-list 2-7;
                cores 4;
                dram 9;
            }
        }
    }
}
```

RELATED DOCUMENTATION

- Minimum Hardware and Software Requirements for Junos Node Slicing | 23
- Connecting the Servers and the Router | 29
- Components of Junos Node Slicing | 4

3

CHAPTER

Upgrading and Managing Junos Node Slicing

Junos Node Slicing Upgrade | 84

Managing Junos Node Slicing | 103

Junos Node Slicing Upgrade

IN THIS SECTION

- Upgrading Junos Node Slicing | 84
- Downgrading JDM for External Server Model | 89
- Downgrading JDM for In-Chassis Model | 92
- Unified ISSU Support | 93
- Managing Multiversion Software Interoperability | 94
- Restarting External Servers | 97
- Updating Host OS on the External Servers | 99
- Applying Security Updates to Host OS | 99
- Applying Security Patches for Ubuntu Container | 101

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

Upgrading Junos Node Slicing

IN THIS SECTION

- Upgrading JDM for External Server Model | 85
- Upgrading JDM for In-Chassis Model | 86
- Upgrading GNF and BSYS | 87
- Upgrading JDM to Support WRL 9 based VM Host - In-Chassis Model | 87

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 19 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

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 package.deb
   ```

   NOTE:

   - A JDM upgrade does not affect any of the running GNFs.
   - Before upgrading JDM, ensure that both JDM deployments are in sync. This means:

- Junos running on a given GNF should support the same SMBIOS version across both the servers.
- Before upgrade, ensure that all GNFs exist on both the servers.

- After upgrading both the JDM servers, you must run `commit synchronize` before configuring any new GNF. If you do not run `commit synchronize` and create new GNFs on server1, you will not be able to do `commit synchronize` later from server0 to server1.
- You must upgrade both the JDMs.

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).

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 terminated. 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`).

Upgrading JDM to Support WRL 9 based VM Host - In-Chassis Model

If the Routing Engine is to run Junos OS 19.3R1 or later, you must upgrade JDM to 19.3R1 or later.

NOTE: Junos OS versions released prior to 19.3R1 use WRL6 version of the VM Host software. Junos OS 19.3R1 brings in WRL9 version of the VM Host software. To check the VM Host version, on the BSYS VM, use the Junos CLI command `show vmhost version`.

Use the following steps to upgrade the JDM.

1. At each of the GNFs, assign primary role to the backup GNFs running on Routing Engine1 (re1).

```
root@router> request chassis routing-engine master switch no-confirm
```

2. On re0, first stop the GNFs from the JDM, and then stop the JDM itself from BSYS.

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

3. Ensure that re0 VM Host version is Junos OS 19.3R1 or later. To check the VM Host version, use the Junos CLI command `show vmhost version`.

   You can use the following Junos CLIs to upgrade VM Host software:

```
root@router> request vmhost software add package-name
root@router> request vmhost reboot
```

   For more information, see Installing, Upgrading, Backing Up, and Recovery of VM Host.

4. When re0 is back up after the reboot, copy the new JDM RPM package (19.3R1 or later) to a directory (for example, /var/tmp).

5. Install the new JDM RPM package on re0 and then start the JDM.

```
root@router> request vmhost jdm add package-name
root@router> request vmhost jdm start
```

   The GNFs on re0 automatically start after this step.

6. Repeat the steps 1 to 5 on Routing Engine 1 (r1).

7. Run the `request server authenticate-peer-server` command at the JDM on both the Routing Engines.

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

8. Configure `set system commit synchronize` and then apply `commit` on re0 JDM.

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

NOTE: The JDM software version 19.3R1 is capable of running on Junos OS version 19.3R1 as well as on Junos OS versions prior to 19.3R1.

SEE ALSO

Installing and Configuring JDM for In-Chassis Model | 51

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. Assign primary role 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
```

NOTE: If you are using Ubuntu, use the command `dpkg --purge jns-jdm` to uninstall JDM.

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 primary Routing Engine (which is the GNF running on server1).

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

9. Assign primary role 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 primary role 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 primary).

```
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 primary Routing Engine (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.
- Starting in Junos OS Release 21.4R1, the MPC11E with SLCs (sub line cards) supports ISSU in zero packet loss mode. If you are running an earlier Junos OS version, do not attempt to perform ISSU on a Junos node slicing setup that has SLCs.

Managing Multiversion Software Interoperability

IN THIS SECTION

- Viewing Software Incompatibility Alarms | **96**
- Viewing Incompatibilities Between Software Versions | **97**

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 terminate 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 terminate, 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-x86-64-17.4I20170713_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

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 97.

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 97.

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 36 .

Applying Security Updates to Host OS

IN THIS SECTION

- Steps to Apply Host OS Security Updates | **100**

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* 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*.

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 Primary Routing Engine switch-over is required to move the Primary 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 Primary 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 primary 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. If you are using the external server model, 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 command:

   root@jdm> `start shell user root`

   If you are using the in-chassis Junos node slicing, use the following command on the BSYS VM to enter JDM:

   root@router> `request vmhost jdm login`

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. If you are using the external server model, from the host OS, restart the JDM.

   user@server# `sudo jdm restart`

   If you are using the in-chassis Junos node slicing, use the following commands on the BSYS VM to restart the JDM:

   root@router> `request vmhost jdm stop`

   root@router> `request vmhost jdm start`

RELATED DOCUMENTATION

Junos Node Slicing Overview | 2

Managing Junos Node Slicing

IN THIS SECTION

- Deleting Guest Network Functions | 103
- Disabling Junos Node Slicing | 104
- Managing Sub Line Cards | 106

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 103.
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*.
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.

Managing Sub Line Cards

IN THIS SECTION

- Operational Commands on BSYS for Line Card Slices | **107**
- Operational Commands on GNF for Line Card Slices | **110**

For an overview of sub-linecards, see "Sub Line Card Overview" on page 16.

For configuring sub-linecards, please refer to "Configuring Sub Line Cards and Assigning Them to GNFs" on page 68.

To manage the sub line cards, you can use the same CLI operational commands that are used to manage full line cards.

You can operate on SLCs from both BSYS as well as their associated GNFs.

From the BSYS, you can see the status of all SLCs on all FPCs, and take actions on any SLC.

From a GNF, you can see the status of, and take actions on, only those SLCs that are assigned to that GNF.

When you run a show command at the BSYS for a sliced line card, the output shows values from BLC and all SLCs of that line card. The annotation `fpc-slot:slc-id` is used to indicate that an output field is from a sub line card. When the same show command is run at a GNF for that line card, the output shows the value only from the specific slice that has been assigned to that GNF.

When you need to take action on an SLC from the BSYS, you must use the new keyword `slc`, together with an SLC ID, to indicate the specific SLC of a specific FPC. When you need to take action on the SLC from its associated GNF, you only need to specify the FPC slot (the SLC ID is implicit).

Operational Commands on BSYS for Line Card Slices

The following are the sample command outputs on the BSYS, where FPC 1 has been sliced.

To view the status of a sliced line card at the BSYS, use the CLI command `show chassis fpc`, as shown below. The slot entries 1:1 and 1:2 indicate the outputs from SLC1 and SLC2 respectively. The slot entry '1' indicates the output of the BLC.

The status of each SLC of a Line card is displayed using the `fpc-slot-id:slc-id` nomenclature as below along with the GNF assignment details:

```
user@bsys> show chassis fpc 1

                     Temp  CPU Utilization (%)   CPU Utilization (%)  Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      1min   5min   15min  DRAM (MB) Heap
Buffer    GNF
  1  Online            58      1          0        1      1      1    5120       36          0
1:1  Online                   11          0       10     10     10    17408      14
0       1
1:2  Online                   21          0       19     19     19    9216       30
0       2
```

To examine the operational values of the line card resources of a sliced line card, use the CLI command `show chassis fpc pic-status`. These operational values match the configured values of the line card resources.

```
user@bsys> show chassis fpc pic-status 1

Slot 1   Online       MPC11E 3D MRATE-40xQSFPP
  SLC 1   Online       FPC1 PFE0-1 4core-17gb                      GNF 1
```

```
  PIC 0  Online       MRATE-5xQSFPP
  PIC 1  Online       MRATE-5xQSFPP
 SLC 2   Online       FPC1 PFE2-7 4core-9gb                       GNF 2
  PIC 2  Online       MRATE-5xQSFPP
  PIC 3  Online       MRATE-5xQSFPP
  PIC 4  Online       MRATE-5xQSFPP
  PIC 5  Online       MRATE-5xQSFPP
  PIC 6  Online       MRATE-5xQSFPP
  PIC 7  Online       MRATE-5xQSFPP
```

To view the software version, uptime and the individual Packet Forwarding Engine assignments of the SLCs, use the CLI command `show chassis fpc slot detail`, as shown below.

```
user@bsys> show chassis fpc 1 detail

Slot 1 information:
  State                               Online
  Temperature                      58 degrees C / 136 degrees F
  Total CPU DRAM                  5120 MB
  Total HBM                       65536 MB
  Start time                          2021-01-06 09:47:31 PST
  Uptime                              20 hours, 57 minutes, 58 seconds
  Max power consumption            1980 Watts
  Operating Bandwidth              4000 G
SLC 1  information:
  State                               Online
  Total CPU cores                     4
  Total CPU DRAM                  17408 MB
  Total HBM                       65536 MB
  Start time                          2021-01-06 09:48:48 PST
  Uptime                              20 hours, 56 minutes, 41 seconds
  Version                           JUNOS 21.1-202012301103.0-EVO <--snip-->
SLC 2  information:
  State                               Online
  Total CPU cores                     4
  Total CPU DRAM                  9216 MB
  Total HBM                       65536 MB
  Start time                          2021-01-06 09:50:22 PST
  Uptime                              20 hours, 55 minutes, 7 seconds
  Version                           JUNOS 21.1-202012301103.0-EVO <--snip-->

PFE Information:
```

```
  PFE  Power ON/OFF  Bandwidth         SLC
  0    ON            500G                1
  1    ON            500G                1
  2    ON            500G                2
  3    ON            500G                2
  4    ON            500G                2
  5    ON            500G                2
  6    ON            500G                2
  7    ON            500G                2
```

To view the node-level assignments of line cards and sub line cards to different GNFs, use the CLI command `show chassis network-slices fpcs` at the BSYS, as shown below. In this example, GNF 1 has 2 line card slices, SLC1 of FPC1 and SLC1 of FPC9, and no full line cards. GNF 2 here has one full line card, FPC6, and 2 line card slices, SLC2 of FPC1 and SLC2 of FPC9.

```
user@bsys> show chassis network-slices fpcs

guest-network-functions:
GNF    FPCs
 1     1:1 9:1
 2     6 1:2 9:2
```

You can take a sub line card offline, bring it online or restart it in the same way as you would with full line cards. You must use the additional keyword `slc` and an SLC ID to indicate the specific SLC. For example, to restart SLC1 of FPC1, use the CLI command `request chassis fpc slot 1 slc 1`.

```
user@bsys> request chassis fpc slot 1 slc 1 ?

Possible completions:
  offline              Take FPC offline
  online               Bring FPC online
  restart              Restart FPC
```

On the BSYS, when you take a sliced FPC offline, and do not specify any SLC, the command is applied to all SLCs of that FPC, that is, all the SLCs will first be taken offline, followed by the FPC itself. Similarly, when you bring a sliced FPC online, the FPC will be first brought online, followed by all the SLCs.

Operational Commands on GNF for Line Card Slices

The operational commands used on a GNF are identical to those that are used on full line cards. When you specify an `fpc-slot`, which has a sliced FPC, the command implicitly operates only on the specific SLC of that FPC that has been assigned to that GNF, that is, the SLC ID is implied.

In the example below, the CLI command `show chassis fpc 1` is entered on GNF A; so, it shows the status of only one SLC, SLC1, since that is the SLC of FPC1 associated with GNF A. It also shows the BLC status of that FPC. Note that this output does not show the status of SLC2 of FPC1, since SLC2 is associated with a different GNF (GNF B).

```
user@gnf-a> show chassis fpc 1

bsys-re0:
--------------------------------------------------------------------------
                     Temp  CPU Utilization (%)   CPU Utilization (%)  Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      1min   5min   15min  DRAM (MB) Heap
Buffer    GNF
  1  Online            58      1          0        1      1      1    5120       36          0
1:1  Online                   12          0       11     11     11    17408      14
0       1
```

In the example below, the same CLI command `show chassis fpc 1` is entered on GNF B; so, it shows only SLC2, which is the SLC of FPC1 associated with GNF B. It also shows the BLC status of that FPC. Note that this output does not show the status of SLC1 of FPC1, since SLC1 is associated with a different GNF (GNF A).

```
user@gnf-b> show chassis fpc 1

bsys-re0:
--------------------------------------------------------------------------
                     Temp  CPU Utilization (%)   CPU Utilization (%)  Memory    Utilization (%)
Slot State            (C)  Total  Interrupt      1min   5min   15min  DRAM (MB) Heap
Buffer    GNF
  1  Online            58      1          0        1      1      1    5120       36          0
1:2  Online                   20          0       19     19     19    9216       30
0       2
```

As another example, the following is the output of the CLI command `show chassis fpc 5 detail` from GNF A. Note that it shows only SLC1 information.

```
user@gnf-a> show chassis fpc 5 detail
bsys-re0:
--------------------------------------------------------------------------
Slot 1 information:
  State                               Online
  Temperature                      58 degrees C / 136 degrees F
  Total CPU DRAM                  5120 MB
  Total HBM                       65536 MB
  Start time                           2021-01-06 09:47:31 PST
  Uptime                               21 hours, 40 minutes, 31 seconds
  Max power consumption            1980 Watts
  Operating Bandwidth               4000 G
SLC 1  information:
  State                               Online
  Total CPU cores                     4
  Total CPU DRAM                  17408 MB
  Total HBM                       65536 MB
  Start time                           2021-01-06 09:48:48 PST
  Uptime                               21 hours, 39 minutes, 14 seconds
  Version                           JUNOS 21.1-202012301103.0-EVO <snip>

PFE Information:

  PFE  Power ON/OFF  Bandwidth         SLC
  0    ON            500G                1
  1    ON            500G                1
```

To take offline, bring online or restart an SLC from its associated GNF, specify only its FPC slot, as shown in the following example (the SLC ID is implied).

```
user@gnf-a> request chassis fpc slot 1 ?

Possible completions:
  offline              Take FPC offline
  online               Bring FPC online
  restart              Restart FPC
```

For a sample sub linecard configuration, please refer to "Sample Configuration for Sub Line Cards" on page 80.

RELATED DOCUMENTATION

Junos Node Slicing Overview | 2

Components of Junos Node Slicing | 4

Sample Configuration for Junos Node Slicing | 71

4

CHAPTER

Configuration Statements for BSYS

af-name | 114

control-plane-bandwidth-percent (Node Slicing) | 116

description (GNF) | 117

description (Abstracted Fabric) | 119

fpc-slice | 121

fpcs (Node Slicing) | 123

network-slices | 125

gnf | 127

collapsed-forward | 129

guest-network-functions | 131

peer-gnf | 133

power | 134

vlan-isolation | 136

vmhost resize vjunos compact | 138

af-name

IN THIS SECTION

- Syntax | 114
- Hierarchy Level | 114
- Description | 115
- Options | 115
- Required Privilege Level | 115
- Release Information | 115

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]
```

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

Release Information

Statement introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | 61

control-plane-bandwidth-percent (Node Slicing)

IN THIS SECTION

- Syntax | 116
- Hierarchy Level | 116
- Description | 116
- Options | 117
- Required Privilege Level | 117
- Release Information | 117

Syntax

```
control-plane-bandwidth-percent percent;
```

Hierarchy Level

```
[edit chassis network-slices guest-network-functions gnf]
```

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

percent 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.

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

network-slices | 125
guest-network-functions | 131
gnf | 127
description (GNF) | 117
fpcs (Node Slicing) | 123

description (GNF)

IN THIS SECTION

- Syntax | 118
- Hierarchy Level | 118
- Description | 118

- Options | 118
- Required Privilege Level | 118
- Release Information | 119

Syntax

```
description description;
```

Hierarchy Level

```
[edit chassis network-slices guest-network-functions gnf]
```

Description

Provide a description string for the specified guest network function (GNF).

Options

description 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.

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

- network-slices | 125
- guest-network-functions | 131
- gnf | 127
- control-plane-bandwidth-percent (Node Slicing) | 116
- fpcs (Node Slicing) | 123

description (Abstracted Fabric)

IN THIS SECTION

- Syntax | 119
- Hierarchy Level | 120
- Description | 120
- Options | 120
- Required Privilege Level | 120
- Release Information | 120

Syntax

```
description description;
```

Hierarchy Level

```
[edit chassis network-slices guest-network-functions gnf af-name]
```

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.

Release Information

Statement introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | **61**

gnf | **127**

control-plane-bandwidth-percent (Node Slicing) | **116**

fpcs (Node Slicing) | **123**

fpc-slice

IN THIS SECTION

- Syntax | 121
- Hierarchy Level | 121
- Description | 122
- Options | 122
- Required Privilege Level | 123
- Release Information | 123

Syntax

```
fpc-slice fpc fpc-slot {
    slc slc-id {
        pfe-id-list pfe-id-list;
        cores number-of-cores;
        dram dram-value;
        }
    }
```

Hierarchy Level

```
[edit chassis network-slices guest-network-functions gnf gnf-id]
```

Description

Configure smaller logical partitions or sub line cards (SLC) of a line card, allocate resources to the SLCs, and assign each SLC to a guest network function (GNF).

This feature is applicable only to the MPC11E line card. An MPC11E line card supports two sub line cards (SLCs). You can assign each GNF with an SLC, which functions like an independent line card.

NOTE:

- You must configure all the SLCs supported by a line card, assign the required resources such as core, DRAM, and Packet forwarding Engines the SLCs before committing the configuration. You must allocate all the Packet forwarding Engines on the line card to the SLCs.

Options

fpc-slot	The slot number of the line card on which you want to configure SLCs.
slc *slc-id*	Specify the sub line card (SLC) ID which you want to assign to the selected GNF. **NOTE**: Do not assign: • two or more SLCs of the same line card to the same GNF. • the same SLC of a line card to more than one GNF.
pfe-id-list *pfe-id-list*	Specify the Packet Forwarding Engine IDs to be allocated to the SLC. You must allocate all the Packet Forwarding Engines on a line card to the SLCs.
cores *number-of-cores*	Specify the number of cores to be allocated to the SLC. Currently, number of cores supported is 4.
dram *dram-value*	Specify the DRAM to be allocated to the SLC. The MPC11E line card supports the following combinations:

SLC1 DRAM (GB)	SLC2 DRAM (GB)	Sub Total (GB)
13	13	26
9	17	26

Required Privilege Level

system—To view this statement in the configuration.

system-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

request chassis fpc slc | **141**

fpcs (Node Slicing)

IN THIS SECTION

- Syntax | **124**
- Hierarchy Level | **124**
- Description | **124**
- Options | **124**

Required Privilege Level | 125
Release Information | 125

Syntax

```
fpcs fpcs;
```

Hierarchy Level

```
[edit chassis network-slices guest-network-functions gnf]
```

Description

Assign FPCs to a guest network function.

NOTE: A given FPC cannot be assigned to more than one GNF.

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.

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

- network-slices | 125
- guest-network-functions | 131
- gnf | 127
- control-plane-bandwidth-percent (Node Slicing) | 116
- description (GNF) | 117

network-slices

IN THIS SECTION

- Syntax | 126
- Hierarchy Level | 126
- Description | 126
- Required Privilege Level | 126
- Release Information | 127

Syntax

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

Hierarchy Level

```
[edit chassis]
```

Description

Configure 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.

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

- Optimizing Fabric Path for Abstracted Fabric Interface | 13
- Configuring an MX Series Router to Operate in BSYS Mode (External Server Model) | 40
- Configuring Guest Network Functions | 58

gnf

IN THIS SECTION

- Syntax | 127
- Hierarchy Level | 128
- Description | 128
- Options | 128
- Required Privilege Level | 128
- Release Information | 128

Syntax

```
gnf id {
    control-plane-bandwidth-percent percent;
    description description;
    fpcs fpcs;
    af-name ;
    collapsed-forward
}
```

Hierarchy Level

```
[edit chassis network-slices guest-network-functions]
```

Description

Define a GNF by assigning an ID to it.

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).

Options

id GNF ID

- **Range:** `1–10`

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.

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

Configuring Guest Network Functions | 58

network-slices | 125

guest-network-functions | 131

collapsed-forward

IN THIS SECTION

- Syntax | 129
- Hierarchy Level | 129
- Description | 130
- Options | 130
- Required Privilege Level | 130
- Release Information | 131

Syntax

```
collapsed-forward (monitor | optimize);
```

Hierarchy Level

```
[edit chassis network-slices guest-network-functions gnf id af-name]
```

Description

Configure a fabric path optimization mode to optimize the fabric path of the traffic flowing over abstracted fabric (af) interfaces between two guest network functions (GNFs). After configuring fabric path optimization, you can use the command `show interfaces af-interface-name` in GNF to view the number of packets that are currently flowing on the optimal / non-optimal path.

NOTE:

- You must configure the `collapsed-forward` statement on both the peer guest network functions (GNFs) on the abstracted fabric interface for the commit to be successful.
- The `collapsed-forward` statement is configured at the abstracted fabric interface level. All logical interfaces on the abstracted fabric interface inherit this configuration.
- We strongly recommend that you do not configure the `collapsed-forward` statement on GNFs that have a mix of supported and unsupported MPCs (MPCs that support fabric path optimization and MPCs that do not) to avoid any erratic behavior.

Options

monitor If you configure this mode, the peer GNF monitors the traffic flow and sends information to the source GNF about the Packet Forwarding Engine to which the traffic is being forwarded currently and the desired Packet Forwarding Engine that could provide an optimized path. However, the source GNF does not forward the traffic towards the desired Packet Forwarding Engine.

optimize If you configure this mode, the peer GNF monitors the traffic flow and sends information to the source GNF about the Packet Forwarding Engine to which the traffic is being forwarded currently and the desired Packet Forwarding Engine that could provide an optimized path. The source GNF then forwards the traffic towards the desired Packet Forwarding Engine.

Required Privilege Level

view-level—To view this statement in the configuration.

control-level—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 19.4R1.

RELATED DOCUMENTATION

- Optimizing Fabric Path for Abstracted Fabric Interface | 13
- network-slices | 125
- guest-network-functions | 131
- gnf | 127
- control-plane-bandwidth-percent (Node Slicing) | 116
- description (GNF) | 117

guest-network-functions

IN THIS SECTION

- Syntax | 131
- Hierarchy Level | 132
- Description | 132
- Required Privilege Level | 132
- Release Information | 132

Syntax

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

```
    }
}
```

Hierarchy Level

```
[edit chassis network-slices]
```

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.

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

Configuring Guest Network Functions | 58

network-slices | 125

peer-gnf

IN THIS SECTION

- Syntax | 133
- Hierarchy Level | 133
- Description | 133
- Options | 134
- Required Privilege Level | 134
- Release Information | 134

Syntax

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

Hierarchy Level

```
[edit chassis network-slices guest-network-functions gnf id af-name]
```

Description

Configure the details of the GNF peer connected using the abstracted fabric (af) interface.

Options

id*peer-gnf -id* Name of the GNF peer connected using the abstracted fabric (af) interface.

- **Range:** 1 through 10

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

Required Privilege Level

system

Release Information

Statement introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | 61

power

IN THIS SECTION

- Syntax | 135
- Hierarchy Level | 135
- Description | 135
- Default | 135
- Options | 135
- Required Privilege Level | 136

Release Information | 136

Syntax

```
power (off | on);
```

Hierarchy Level

```
[edit chassis fpc slot-number slc slc-id]
```

Description

Power on or power off sub line cards (SLCs).

Default

on

Options

slc *pfe-id* — The ID of the SLC to be powered off.

power (off | on) — Power off or on the selected SLC.

- **Default:** on

Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

Sub Line Card Overview | 16

vlan-isolation

IN THIS SECTION

- Syntax | 136
- Hierarchy Level | 137
- Description | 137
- Required Privilege Level | 137
- Release Information | 138

Syntax

```
vlan-isolation;
```

Hierarchy Level

```
[edit chassis network-slices]
```

Description

Configure VLAN isolation. For the VLAN isolation configuration to take effect, you must configure the `set chassis network-slices vlan-isolation` CLI configuration statement on all the Routing Engines of the BSYS and GNFs, and then reboot the router.

On an MX without Junos Node Slicing enabled, all control traffic between the Routing Engine and the FPCs, as well as between the Routing Engines themselves, occur over an internal network. This network includes both the Routing Engines and all the FPCs. When you configure Junos node slicing, this internal network is shared by all the guest network function (GNF) Routing Engine pairs and their respective set of FPCs. The internal network has sufficient bandwidth to accommodate this traffic without additional configuration. However, if you want to isolate the internal traffic of each GNF from all others, use the `set chassis network-slices vlan-isolation` CLI configuration statement. This configuration assigns a separate internal VLAN for each GNF, resulting in all control traffic within each GNF being VLAN tagged and isolated from those of other GNFs.

NOTE:

- VLAN isolation is disabled by default.
- If you want to configure the sub line card feature, you must ensure that VLAN isolation is disabled. To disable VLAN isolation, delete the `set chassis network-slices vlan-isolation` configuration statement. See also: "Sub Line Card Overview" on page 16.

Required Privilege Level

system

Release Information

Statement introduced in Junos OS Release 21.2R1.

RELATED DOCUMENTATION

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | 61

vmhost resize vjunos compact

IN THIS SECTION

- Syntax | 138
- Hierarchy Level | 138
- Description | 139
- Required Privilege Level | 139
- Release Information | 139

Syntax

```
vmhost resize vjunos compact
```

Hierarchy Level

```
[edit]
```

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. Configuring the `vmhost resize vjunos compact` statement enables hyperthreading in Routing Engine CPU. The in-chassis Junos node slicing requires that hyperthreading be enabled.

Required Privilege Level

view-level—To view this statement in the configuration.

control-level—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 19.1R1.

5

CHAPTER

Operational Commands for BSYS

request chassis fpc slc | 141

request vmhost jdm add (In-Chassis Model) | 142

request vmhost jdm delete (In-Chassis Model) | 144

request vmhost jdm start (In-Chassis Model) | 145

request vmhost jdm login (In-Chassis Model) | 146

request vmhost jdm stop (In-Chassis Model) | 148

show vmhost status (In-Chassis Model) | 149

show vmhost jdm status (In-Chassis Model) | 151

show chassis network-slices | 153

show chassis fpc pic-status (Node Slicing) | 158

show chassis fpc (Node Slicing) | 160

show chassis adc (Node Slicing) | 164

show chassis network-slices fpcs | 167

show system anomalies gnf-id | 169

request chassis fpc slc

IN THIS SECTION

- Syntax | 141
- Description | 141
- Options | 141
- Required Privilege Level | 142
- Release Information | 142

Syntax

```
request chassis fpc fpc-slot slc slc-id (offline | restart | online)
```

Description

(Applies to the base system or BSYS). Start, stop or restart the sub line cards (SLCs).

NOTE: On BSYS, entering the `request chassis fpc fpc-slot offline/restart/online` command initiates the offline/online/restart process of all the SLCs (sub line cards) of the selected line card (FPC). On GNF, entering the `request chassis fpc fpc-slot offline/restart/online` command initiates the offline/online/restart process of only those SLCs that are assigned to the selected GNF.

Options

fpc-slot	Specify the line card slot.
slc-id	Specify the SLC you want to take offline, bring online or restart.

Required Privilege Level

View

Release Information

Command introduced in Junos OS Release 21.1R1.

RELATED DOCUMENTATION

fpc-slice | 121

request vmhost jdm add (In-Chassis Model)

IN THIS SECTION

- Syntax | 142
- Description | 143
- Options | 143
- Required Privilege Level | 143
- Release Information | 143

Syntax

```
request vmhost jdm add package-name
```

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

Release Information

Command introduced in Junos OS Release 19.1R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

show virtual-network-functions | 251

request server authenticate-peer-server | 247

request vmhost jdm delete (In-Chassis Model)

IN THIS SECTION

- Syntax | 144
- Description | 144
- Required Privilege Level | 144
- Release Information | 144

Syntax

```
request vmhost jdm delete
```

Description

Uninstalls the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos node slicing.

Required Privilege Level

maintenance

Release Information

Command introduced in Junos OS Release 19.1R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

show virtual-network-functions | 251

request server authenticate-peer-server | 247

request vmhost jdm start (In-Chassis Model)

IN THIS SECTION

- Syntax | 145
- Description | 145
- Required Privilege Level | 145
- Sample Output | 146
- Release Information | 146

Syntax

```
request vmhost jdm start
```

Description

Starts the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos node slicing.

Required Privilege Level

maintenance

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
```

Release Information

Command introduced in Junos OS Release 19.1R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- show virtual-network-functions | 251
- request server authenticate-peer-server | 247

request vmhost jdm login (In-Chassis Model)

IN THIS SECTION

- Syntax | 147
- Description | 147
- Required Privilege Level | 147
- Sample Output | 147
- Release Information | 147

Syntax

```
request vmhost jdm login
```

Description

Logs in to the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos node slicing.

Required Privilege Level

maintenance

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
```

Release Information

Command introduced in Junos OS Release 19.1R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

show virtual-network-functions | 251

request server authenticate-peer-server | 247

request vmhost jdm stop (In-Chassis Model)

IN THIS SECTION

- Syntax | 148
- Description | 148
- Required Privilege Level | 148
- Release Information | 149

Syntax

```
request vmhost jdm stop
```

Description

Stops the Juniper Device Manager (JDM). This command is applicable to in-chassis Junos node slicing.

Required Privilege Level

maintenance

Release Information

Command introduced in Junos OS Release 19.1R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

show virtual-network-functions | 251

request server authenticate-peer-server | 247

show vmhost status (In-Chassis Model)

IN THIS SECTION

- Syntax | 149
- Description | 150
- Options | 150
- Required Privilege Level | 150
- Sample Output | 151
- Release Information | 151

Syntax

```
show vmhost status (re0 | re1)
<routing-engine (backup | both | local | master | other)>
<invoke-on  (all-routing-engines | other-routing-engine)>
```

Description

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`.

Options

re0 Displays Routing Engine 0 (re0) VM host status.

re1 Displays Routing Engine 1 (re1) VM host status.

routing-engine Displays the VM host status of a specific routing engine. You have the following sub-options:

- backup
- primary
- both
- local
- other

invoke-on Displays the VM host status of the remote Routing Engines or all Routing Engines. You have the following options:

- all-routing-engines
- other-routing-engine

Required Privilege Level

View

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
```

Release Information

Command introduced in Junos OS Release 19.1R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- show virtual-network-functions | 251
- request server authenticate-peer-server | 247

show vmhost jdm status (In-Chassis Model)

IN THIS SECTION

- Syntax | 152
- Description | 152

- Required Privilege Level | 152
- Sample Output | 152
- Release Information | 153

Syntax

```
show vmhost jdm status
```

Description

Displays the Juniper Device Manager (JDM) status. This command is applicable to in-chassis Junos node slicing.

Required Privilege Level

View

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)
```

Release Information

Command introduced in Junos OS Release 19.1R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239
show virtual-network-functions | 251
request server authenticate-peer-server | 247

show chassis network-slices

IN THIS SECTION

- Syntax | 153
- Description | 154
- Required Privilege Level | 154
- Output Fields | 154
- Sample Output | 155
- Release Information | 157

Syntax

```
show chassis network-slices
<gnf gnf-id>
```

Description

Display Junos node slicing information for the guest network functions (GNFs) configured on the base system (BSYS). The `gnf` *`gnf-id`* option displays the information about a particular GNF.

Required Privilege Level

view

Output Fields

Table 7 on page 154 lists the output fields for the `show chassis network-slices` command. Output fields are listed in the approximate order in which they appear.

Table 7: 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. • `Online`—GNF online and running. • `Offline`—GNF is offline.
Uptime	Duration for which the GNFs have been up and running. In case of a GNF Routing Engine switchover, the uptime value is reset as the new primary GNF Routing Engine establishes a fresh connection with the BSYS. This behavior is specific to GNFs and does not have any functional impact.
GNF ID	Shows the GNF ID.

Table 7: show chassis network-slices Output Fields *(Continued)*

Field Name	Field Description
GNF description	Shows the description of GNF.
FPCs assigned	Shows the FPC slot numbers assigned to the GNF.
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 primary RE	Primary Routing Engine slot.
GNF uptime	Duration for which the GNF has been up and running. In case of a GNF Routing Engine switchover, the GNF uptime value is reset as the new primary GNF Routing Engine establishes a fresh connection with the BSYS. This behavior is specific to GNFs and does not have any functional impact.
GNF Routing Engine Status:	Show the details of the Routing Engines in the slot 0 and 1. It includes the Current State, (primary 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
```

Release Information

Command introduced in Junos OS Release 17.2R1.

RELATED DOCUMENTATION

show chassis fpc pic-status (Node Slicing) | 158
show chassis fpc (Node Slicing) | 160
show chassis adc (Node Slicing) | 164
show chassis network-slices fpcs | 167

show chassis fpc pic-status (Node Slicing)

IN THIS SECTION

- Syntax | 158
- Description | 158
- Required Privilege Level | 159
- Sample Output | 159
- Release Information | 160

Syntax

```
show chassis fpc pic-status
```

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

Sample Output

user@router> show chassis fpc pic-status

command-name

```
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
```

Release Information

Command introduced before Junos OS Release 7.4.

RELATED DOCUMENTATION

- show chassis network-slices | 153
- show chassis fpc (Node Slicing) | 160
- show chassis adc (Node Slicing) | 164
- show chassis network-slices fpcs | 167

show chassis fpc (Node Slicing)

IN THIS SECTION

- Syntax | 161
- Description | 161
- Required Privilege Level | 161
- Output Fields | 161
- Sample Output | 163
- Release Information | 164

Syntax

```
show chassis fpc
```

Description

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

Required Privilege Level

view

Output Fields

Table 8 on page 162 lists the output fields for the `show chassis fpc` command. Output fields are listed in the approximate order in which they appear.

Table 8: 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: • `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.
`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.

Table 8: show chassis fpc Output Fields *(Continued)*

Field Name	Field Description
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

command-name

user@router> show chassis fpc

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

```
5  Online   58      22          0     21      20      20     3136      16      22    4
6  Online   49      17          0     15      16      16     3136      13      20    1
7  Online   44      11          0     10      10      10     3584       6      25    5
8  Online   40      19          0     18      18      18     3584       8      25    5
9  Online   44      19          0     20      20      20     3584       8      25    5
```

Release Information

Command introduced before Junos OS Release 7.4.

RELATED DOCUMENTATION

- show chassis network-slices | 153
- show chassis fpc pic-status (Node Slicing) | 158
- show chassis adc (Node Slicing) | 164
- show chassis network-slices fpcs | 167

show chassis adc (Node Slicing)

IN THIS SECTION

- Syntax | 165
- Description | 165
- Required Privilege Level | 165
- Output Fields | 165
- Sample Output | 166
- Release Information | 166

Syntax

```
show chassis adc
```

Description

Display chassis information about the adapter cards (ADCs).

Required Privilege Level

view

Output Fields

Table 9 on page 165 lists the output fields for the `show chassis adc` command. Output fields are listed in the approximate order in which they appear.

Table 9: show chassis adc Output Fields

Field Name	Field Description
`Slot`	Slot number.
`State`	Status of the adapter card. • `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.

Table 9: show chassis adc Output Fields *(Continued)*

Field Name	Field Description
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
```

Release Information

Command introduced in Junos OS Release 12.3.

RELATED DOCUMENTATION

show chassis network-slices | 153

show chassis fpc pic-status (Node Slicing) | 158

show chassis fpc (Node Slicing) | 160

show chassis network-slices fpcs

IN THIS SECTION

- Syntax | 167
- Description | 167
- Required Privilege Level | 167
- Output Fields | 168
- Sample Output | 168
- Release Information | 168

Syntax

```
show chassis network-slices fpcs
```

Description

Display information about the FPCs associated with different guest network functions (GNFs).

Required Privilege Level

view

Output Fields

Table 10 on page 168 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 10: 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

command-name

```
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
```

Release Information

Command introduced in Junos OS Release 17.2R1.

RELATED DOCUMENTATION

show chassis network-slices | 153

show chassis fpc pic-status (Node Slicing) | 158

show chassis fpc (Node Slicing) | 160

show chassis adc (Node Slicing) | 164

show system anomalies gnf-id

IN THIS SECTION

- Syntax | 169
- Description | 169
- Options | 170
- Required Privilege Level | 170
- Output Fields | 170
- Sample Output | 171
- Release Information | 172

Syntax

```
show system anomalies gnf-id id (all-anomalies | config | fru | system)
```

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

gnf-id *id* Specify the GNF ID for which you want to view the software incompatibilities.

all-anomalies Display the multiversion software incompatibilities from all categories—system, configuration and FRU.

config Display feature-level incompatibilities between software versions.

fru Display the FRU-level incompatibilities between software versions. This can be an incompatibility pertaining to the support for a specific FRU.

system Display the system-level incompatibilities between software versions. These include IPC incompatibility, CLI or SNMP incompatibility.

Required Privilege Level

View

Output Fields

Table 11 on page 170 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 11: show system anomalies gnf-id Output Fields

Field Name	Field Description
`Anomaly Type`	Shows the software incompatibility type. The following are the possible values: • SYS—Indicates system-specific incompatibilities. • FRU—Indicates FRU-specific incompatibilities. • CONFIG—Indicates feature-specific incompatibilities.
`Anomaly ID`	Shows the incompatibility ID.

Table 11: 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: • WARN—Indicates the presence of a minor incompatibility. It causes a warning. • ABORT—Indicates the presence of a major incompatibility. It causes a termination 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.

Table 11: show system anomalies gnf-id Output Fields *(Continued)*

Firmware	Firmware of the FRU.

Sample Output

command-name

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

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

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- show system anomalies (GNF) | 220
- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247

CHAPTER 6

Operational Commands for GNF

show chassis gnf | 174

show chassis gnf | 177

show chassis hardware (GNF) | 180

show chassis fpc (GNF) | 183

show chassis fpc pic-status (GNF) | 186

show chassis adc (GNF) | 188

show interfaces (Abstracted Fabric) | 190

show system anomalies (GNF) | 220

show chassis gnf

IN THIS SECTION

- Syntax | 174
- Description | 174
- Required Privilege Level | 174
- Output Fields | 174
- Sample Output | 176
- Release Information | 176

Syntax

```
show chassis gnf
```

Description

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

Required Privilege Level

view

Output Fields

Table 12 on page 175 lists the output fields for the `show chassis gnf` command. Output fields are listed in the approximate order in which they appear.

Table 12: show chassis gnf Output Fields

Field Name	Field Description
GNF ID	Shows the GNF ID.
`GNF State`	Status of the GNF. • `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 primary RE	Primary Routing Engine slot.
`GNF uptime`	Duration for which the GNF has been up and running.
GNF Routing Engine Status:	The details of the Routing Engines in the slot 0 and 1. The details include the Current State, (primary 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
```

Release Information

Command introduced in Junos OS Release 18.2R1.

RELATED DOCUMENTATION

- show chassis hardware (GNF) | 180
- show chassis fpc pic-status (GNF) | 186
- show chassis adc (GNF) | 188
- show chassis network-slices | 153

show chassis gnf

IN THIS SECTION

- Syntax | 177
- Description | 177
- Required Privilege Level | 177
- Output Fields | 177
- Sample Output | 179
- Release Information | 179

Syntax

```
show chassis gnf
```

Description

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

Required Privilege Level

view

Output Fields

Table 13 on page 178 lists the output fields for the `show chassis gnf` command. Output fields are listed in the approximate order in which they appear.

Table 13: show chassis gnf Output Fields

Field Name	Field Description
GNF ID	Shows the GNF ID.
`GNF State`	Status of the GNF. • `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 primary RE	Primary Routing Engine slot.
`GNF uptime`	Duration for which the GNF has been up and running.
GNF Routing Engine Status:	The details of the Routing Engines in the slot 0 and 1. The details include the Current State, (primary 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
```

Release Information

Command introduced in Junos OS Release 18.2R1.

RELATED DOCUMENTATION

- show chassis hardware (GNF) | 180
- show chassis fpc pic-status (GNF) | 186
- show chassis adc (GNF) | 188
- show chassis network-slices | 153

show chassis hardware (GNF)

IN THIS SECTION

- Syntax | 180
- Description | 180
- Required Privilege Level | 180
- Output Fields | 180
- Sample Output | 181
- Release Information | 182

Syntax

```
show chassis hardware
```

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

Output Fields

Table 14 on page 181 lists the output fields for the `show chassis hardware` command. Output fields are listed in the approximate order in which they appear.

Table 14: show chassis hardware Output Fields

Field Name	Field Description
Item	Chassis component: • 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

command-name

```
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
```

Release Information

Command introduced before Junos OS Release 7.4.

RELATED DOCUMENTATION

show chassis fpc (GNF) | 183

show chassis fpc pic-status (GNF) | 186

show chassis adc (GNF) | 188

show chassis fpc (GNF)

IN THIS SECTION

- Syntax | 183
- Description | 183
- Required Privilege Level | 183
- Output Fields | 184
- Sample Output | 185
- Release Information | 186

Syntax

```
show chassis fpc
```

Description

Display information about the Flexible PIC Concentrators (fpcs) assigned to the guest network function (GNF).

Required Privilege Level

view

Output Fields

Table 15 on page 184 lists the output fields for the `show chassis fpc` command. Output fields are listed in the approximate order in which they appear.

Table 15: 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: • `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.
`1 min CPU utilization (%)`	Information about the Routing Engine's CPU utilization in the past 1 minute.

Table 15: show chassis fpc Output Fields *(Continued)*

Field Name	Field Description
`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 might 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

command-name

```
user@router>show chassis fpc
                     Temp CPU Utilization(%) CPU Utilization(%) Memory Utilization(%)
Slot State      (C)  Total Interrupt    1min 5min  15min  DRAM(MB)  Heap  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
```

Release Information

Command introduced before Junos OS Release 7.4.

RELATED DOCUMENTATION

show chassis hardware (GNF) | 180
show chassis fpc pic-status (GNF) | 186
show chassis adc (GNF) | 188

show chassis fpc pic-status (GNF)

IN THIS SECTION

Syntax | 186
Description | 186
Required Privilege Level | 187
Sample Output | 187
Release Information | 187

Syntax

```
show chassis fpc pic-status
```

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

Sample Output

command-name

```
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
```

Release Information

Command introduced before Junos OS Release 7.4.

RELATED DOCUMENTATION

- show chassis hardware (GNF) | 180
- show chassis fpc (GNF) | 183
- show chassis adc (GNF) | 188

show chassis adc (GNF)

IN THIS SECTION

- Syntax | 188
- Description | 188
- Required Privilege Level | 188
- Output Fields | 188
- Sample Output | 189
- Release Information | 189

Syntax

```
show chassis adc
```

Description

Display chassis information about the adapter cards (ADCs) assigned to the guest network function (GNF).

Required Privilege Level

view

Output Fields

Table 16 on page 189 lists the output fields for the `show chassis adc` command. The output fields are listed in the approximate order in which they appear.

Table 16: show chassis adc Output Fields

Field Name	Field Description
`Slot`	Slot number.
`State`	Status of the adapter card. • `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

command-name

```
user@router> show chassis adc

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

Release Information

Command introduced in Junos OS Release 12.3.

RELATED DOCUMENTATION

show chassis hardware (GNF) | 180

show chassis fpc (GNF) | 183

show chassis fpc pic-status (GNF) | 186

show interfaces (Abstracted Fabric)

IN THIS SECTION

- Syntax | 190
- Description | 190
- Options | 191
- Required Privilege Level | 191
- Output Fields | 191
- Sample Output | 197
- Release Information | 219

Syntax

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

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.
- Starting from Junos OS Release 21.4R1, the `show interface` *af-name* `command does not display the information about the load balancing of transmit traffic on each peer GNF Forwarding Element (FE). For this information, use the show interfaces lb-stats` *af-name* command.

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

Output Fields

No Link Title describes the output fields for the `show interfaces` command. Output fields are listed in the approximate order in which they appear.

Table 17: show interfaces 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 *Common Output Fields Description*.	All levels
`Interface flags`	Information about the interface. Possible values are described in the “Interface Flags” section under *Common Output Fields Description*.	All levels
`Link type`	Link type. For example, Full-Duplex.	All levels
`Damping`	Damping information.	`detail extensive`

Table 17: show interfaces Output Fields *(Continued)*

Field Name	Field Description	Level of Output
`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`
`Traffic statistics`	Number and rate of bytes and packets received and transmitted on the physical interface. • `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`

Table 17: show interfaces Output Fields *(Continued)*

Field Name	Field Description	Level of Output
`Input errors`	Input errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • `Errors`—Sum of the incoming frame terminates 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: • `Carrier transitions`—Number of times the interface has gone from `down` to `up`. • `Errors`—Sum of the outgoing frame terminates and FCS errors. • `Drops`—Number of packets dropped by the output queue. **NOTE**: • `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

Table 17: show interfaces Output Fields *(Continued)*

Field Name	Field Description	Level of Output
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
Collapsed Forwarding Optimized Statistics	This section has the following subsections to display fabric optimization information. • Optimal Packets—shows the packets that are already forwarded to optimal Packet Forwarding Engine based on the abstracted fabric interface load balancing. • Non-optimal Packets—By default, these packets are forwarded to the non-optimal Packet Forwarding Engine by abstracted fabric interface. If you have configured the `optimize` mode using the command `set chassis network-slices guest-network-functions gnf` *`id`* `collapsed-forward optimize`, this section shows the packets that are optimized because of this configuration. If you have configured the `monitor` mode using the command `set chassis network-slices guest-network-functions gnf` *`id`* `collapsed-forward monitor`, this section shows non-optimal packets.	All levels
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 *Common Output Fields Description*.	All levels

Table 17: show interfaces Output Fields *(Continued)*

Field Name	Field Description	Level of Output
`Encapsulation`	Encapsulation on the logical interface.	All levels
`Protocol`	Protocol family. Possible values are described in the “Protocol Field” section under *Common Output Fields Description*.	`detail extensive` none
`MTU`	Maximum transmission unit size on the logical interface.	`detail extensive` none
`Traffic statistics`	Number and rate of bytes and packets received and transmitted on the specified interface set. • `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 *Common Output Fields Description*.	`detail extensive`
`Addresses, Flags`	Information about the address flags. Possible values are described in the “Addresses Flags” section under *Common Output Fields Description*.	`detail extensive` none

Table 17: show interfaces 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 *Common Output Fields Description*.	`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 brief

This output is applicable to Junos OS Releases 21.4R1 and later, and versions prior to 18.3R1.

```
user@router> show interfaces af3 brief
Physical interface: af3, Enabled, Physical link is Up
  Type: Ethernet, Link-level type: Ethernet, MTU: 9212, Clocking: Unspecified, Speed: 1040000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000

  Logical interface af3.0
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 0x8100.200 ]  Encapsulation: ENET2
    inet  34.34.34.1/24
```

```
    mpls
    multiservice

  Logical interface af3.1
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.300 0x8100.400 ]  Encapsulation: ENET2
    inet6 34::1/64
          fe80::290:6901:2c2c:df00/64
    mpls
    multiservice

  Logical interface af3.32767
    Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ]  Encapsulation: ENET2
    multiservice
```

show interfaces af-interface-name

This output is applicable to Junos OS Releases 21.4R1 and later, and versions prior to 18.3R1.

```
user@router> show interfaces af3
Physical interface: af3, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 931
  Type: Ethernet, Link-level type: Ethernet, MTU: 9212, Speed: 1040000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Current address: 00:90:69:2c:c0:63, Hardware address: 00:90:69:2c:c0:63
  Last flapped   : 2021-10-17 17:24:10 PDT (1w2d 08:14 ago)
  Input rate     : 616 bps (1 pps)
  Output rate    : 1088 bps (0 pps)
  Bandwidth      : 1040 Gbps
  Peer GNF id    : 3
  Collapsed Forwarding : Disabled
  Peer GNF Forwarding element(FE) view :
  FPC slot:FE num  FE Bandwidth(Gbps) Status
       7:0                   130         Up
       7:1                   130         Up
       7:2                   130         Up
       7:3                   130         Up
      17:0                   130         Up
      17:1                   130         Up
      17:2                   130         Up
```

```
    17:3                  130         Up
Collapsed Forwarding Statistics :
          Optimal Packets             Non-optimal Packets
                       0                               0
          Optimal Bytes               Non-optimal Bytes
                      0                               0

Logical interface af3.0 (Index 375) (SNMP ifIndex 1123)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 0x8100.200 ]
  Encapsulation: ENET2
  Input packets : 8371
  Output packets: 1683613372
  Protocol inet, MTU: 9190
  Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1,
  Curr new hold cnt: 0, NH drop cnt: 0
    Flags: Sendbcast-pkt-to-re
    Addresses, Flags: Is-Preferred Is-Primary
      Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255
  Protocol mpls, MTU: 9178, Maximum labels: 3
    Flags: Is-Primary
  Protocol multiservice, MTU: Unlimited
    Flags: Is-Primary

Logical interface af3.1 (Index 376) (SNMP ifIndex 1125)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.300 0x8100.400 ]
  Encapsulation: ENET2
  Input packets : 1528
  Output packets: 1524
  Protocol inet6, MTU: 9190
  Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1,
  Curr new hold cnt: 0, NH drop cnt: 0
    Flags: Is-Primary
    Addresses, Flags: Is-Preferred Is-Primary
      Destination: 34::/64, Local: 34::1
    Addresses, Flags: Is-Preferred
      Destination: fe80::/64, Local: fe80::290:6901:2c2c:df00
  Protocol mpls, MTU: 9178, Maximum labels: 3
  Protocol multiservice, MTU: Unlimited

Logical interface af3.32767 (Index 377) (SNMP ifIndex 1124)
  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 detail

This output is applicable to Junos OS Releases 21.4R1 and later, and versions prior to 18.3R1.

```
user@router> show interfaces af3 detail
Physical interface: af3, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 931, Generation: 148
  Type: Ethernet, Link-level type: Ethernet, MTU: 9212, Clocking: Unspecified, Speed: 1040000mbps
  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: 00:90:69:2c:c0:63, Hardware address: 00:90:69:2c:c0:63
  Alternate link address: Unspecified
  Last flapped   : 2021-10-17 17:24:10 PDT (1w2d 08:40 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :     1713156145106484                  280 bps
   Output bytes  :     1704691343752013                 1008 bps
   Input  packets:        3467927904602                    0 pps
   Output packets:        3466096797054                    0 pps
   IPv6 transit statistics:
   Input  bytes  :                1792
   Output bytes  :                   0
   Input  packets:                  30
   Output packets:                   0
  Bandwidth      : 1040 Gbps
  Peer GNF id    : 3
  Collapsed Forwarding : Disabled
  Peer GNF Forwarding element(FE) view :
  FPC slot:FE num  FE Bandwidth(Gbps) Status
       7:0                   130         Up
       7:1                   130         Up
       7:2                   130         Up
       7:3                   130         Up
```

```
      17:0                  130         Up
      17:1                  130         Up
      17:2                  130         Up
      17:3                  130         Up
  Collapsed Forwarding Statistics :
            Optimal Packets             Non-optimal Packets
                          0                               0
            Optimal Bytes               Non-optimal Bytes
                        0                               0

  Logical interface af3.0 (Index 375) (SNMP ifIndex 1123) (Generation 448)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 0x8100.200 ]  Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :               673655
     Output bytes  :         824966843167
     Input  packets:                 9811
     Output packets:           1683614943
    Local statistics:
     Input  bytes  :               443858
     Output bytes  :               616248
     Input  packets:                 6217
     Output packets:                 7233
    Transit statistics:
     Input  bytes  :               229797                   8 bps
     Output bytes  :         824966226919                 280 bps
     Input  packets:                 3594                   0 pps
     Output packets:           1683607710                   0 pps
    Protocol inet, MTU: 9190
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold cnt: 0, NH drop
cnt: 0
    Generation: 416, Route table: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255, Generation: 302
    Protocol mpls, MTU: 9178, Maximum labels: 3, Generation: 417, Route table: 0
      Flags: Is-Primary
    Protocol multiservice, MTU: Unlimited, Generation: 418, Route table: 0
      Flags: Is-Primary
      Policer: Input: __default_arp_policer__

  Logical interface af3.1 (Index 376) (SNMP ifIndex 1125) (Generation 449)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.300 0x8100.400 ]  Encapsulation: ENET2
    Traffic statistics:
```

```
      Input  bytes  :               149864
      Output bytes  :               181586
      Input  packets:                 1780
      Output packets:                 1773
      IPv6 transit statistics:
      Input  bytes  :               149864
      Output bytes  :               181586
      Input  packets:                 1780
      Output packets:                 1773
    Local statistics:
      Input  bytes  :               149672
      Output bytes  :               181586
      Input  packets:                 1777
      Output packets:                 1773
    Transit statistics:
      Input  bytes  :                  192                    0 bps
      Output bytes  :                    0                    0 bps
      Input  packets:                    3                    0 pps
      Output packets:                    0                    0 pps
      IPv6 transit statistics:
       Input  bytes  :                 192                    0 bps
       Output bytes  :                   0                    0 bps
       Input  packets:                   3                    0 pps
       Output packets:                   0                    0 pps
    Protocol inet6, MTU: 9190
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold cnt: 0, NH drop
cnt: 0
    Generation: 419, Route table: 0
      Flags: Is-Primary
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 34::/64, Local: 34::1
    Generation: 304
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::290:6901:2c2c:df00
    Protocol mpls, MTU: 9178, Maximum labels: 3, Generation: 306
    Protocol multiservice, MTU: Unlimited, Generation: 420, Route table: 0
    Generation: 421, Route table: 0
      Policer: Input: __default_arp_policer__

  Logical interface af3.32767 (Index 377) (SNMP ifIndex 1124) (Generation 450)
    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: 422, Route table: 0
      Flags: None
      Policer: Input: __default_arp_policer__
```

show interfaces af-interface-name extensive

This output is applicable to Junos OS Releases 21.4R1 and later, and versions prior to 18.3R1.

```
user@router> show interfaces af3 extensive
Physical interface: af3, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 931, Generation: 148
  Type: Ethernet, Link-level type: Ethernet, MTU: 9212, Clocking: Unspecified, Speed: 1040000mbps
  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: 00:90:69:2c:c0:63, Hardware address: 00:90:69:2c:c0:63
  Alternate link address: Unspecified
  Last flapped   : 2021-10-17 17:24:10 PDT (1w2d 08:42 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :     1713156145118374                 768 bps
   Output bytes  :     1704691343766497                   0 bps
   Input  packets:        3467927904777                   1 pps
   Output packets:        3466096797232                   0 pps
```

```
    IPv6 transit statistics:
    Input  bytes  :                1792
    Output bytes  :                   0
    Input  packets:                  30
    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      : 1040 Gbps
  Peer GNF id    : 3
  Collapsed Forwarding : Disabled
  Peer GNF Forwarding element(FE) view :
  FPC slot:FE num  FE Bandwidth(Gbps) Status
       7:0                   130         Up
       7:1                   130         Up
       7:2                   130         Up
       7:3                   130         Up
      17:0                   130         Up
      17:1                   130         Up
      17:2                   130         Up
      17:3                   130         Up
  Collapsed Forwarding Statistics :
            Optimal Packets             Non-optimal Packets
                          0                               0
            Optimal Bytes               Non-optimal Bytes
                        0                               0

  Logical interface af3.0 (Index 375) (SNMP ifIndex 1123) (Generation 448)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 0x8100.200 ]  Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :              683507
     Output bytes  :        824966855195
     Input  packets:                9962
     Output packets:          1683615097
    Local statistics:
     Input  bytes  :              450058
     Output bytes  :              624868
     Input  packets:                6310
     Output packets:                7333
    Transit statistics:
     Input  bytes  :              233449                  768 bps
```

```
     Output bytes  :         824966230327                   0 bps
     Input  packets:                 3652                   1 pps
     Output packets:           1683607764                   0 pps
    Protocol inet, MTU: 9190
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold cnt: 0, NH drop
cnt: 0
    Generation: 416, Route table: 0
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255, Generation: 302
    Protocol mpls, MTU: 9178, Maximum labels: 3, Generation: 417, Route table: 0
      Flags: Is-Primary
    Protocol multiservice, MTU: Unlimited, Generation: 418, Route table: 0
      Flags: Is-Primary
      Policer: Input: __default_arp_policer__

  Logical interface af3.1 (Index 376) (SNMP ifIndex 1125) (Generation 449)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.300 0x8100.400 ]  Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :              152024
     Output bytes  :              184042
     Input  packets:                1806
     Output packets:                1797
     IPv6 transit statistics:
     Input  bytes  :              152024
     Output bytes  :              184042
     Input  packets:                1806
     Output packets:                1797
    Local statistics:
     Input  bytes  :              151832
     Output bytes  :              184042
     Input  packets:                1803
     Output packets:                1797
    Transit statistics:
     Input  bytes  :                 192                   0 bps
     Output bytes  :                   0                   0 bps
     Input  packets:                   3                   0 pps
     Output packets:                   0                   0 pps
     IPv6 transit statistics:
      Input  bytes  :                 192                   0 bps
      Output bytes  :                   0                   0 bps
      Input  packets:                   3                   0 pps
      Output packets:                   0                   0 pps
```

```
    Protocol inet6, MTU: 9190
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold cnt: 0, NH drop
cnt: 0
    Generation: 419, Route table: 0
      Flags: Is-Primary
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 34::/64, Local: 34::1
    Generation: 304
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::290:6901:2c2c:df00
    Protocol mpls, MTU: 9178, Maximum labels: 3, Generation: 306
    Protocol multiservice, MTU: Unlimited, Generation: 420, Route table: 0
    Generation: 421, Route table: 0
      Policer: Input: __default_arp_policer__

  Logical interface af3.32767 (Index 377) (SNMP ifIndex 1124) (Generation 450)
    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: 422, Route table: 0
      Flags: None
      Policer: Input: __default_arp_policer__
```

show interfaces af-interface-name

This output is applicable to Junos OS Releases starting from 18.3R1 to 21.3R1.

```
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 Releases starting from 18.3R1 to 21.3R1.

```
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: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255, 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: Sendbcast-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 Releases starting from 18.3R1 to 21.3R1.

```
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: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
```

```
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255, 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: Sendbcast-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 Releases starting from 18.3R1 to 21.3R1.

```
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: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255
    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 af2 (with collapsed forwarding configured)

```
user@router> show interfaces af2

Physical interface: af2, Enabled, Physical link is Up
  Interface index: 152, SNMP ifIndex: 626
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Speed: 1600000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
```

```
Current address: 00:90:69:00:c0:4a, Hardware address: 00:90:69:00:c0:4a
Last flapped   : 2019-10-01 06:19:50 PDT (2d 00:54 ago)
Input rate     : 312 bps (0 pps)
Output rate    : 144327758752 bps (18371659 pps)
Bandwidth      : 1600 Gbps
Peer GNF id    : 7
Collapsed Forwarding : Optimized
Peer GNF Forwarding element(FE) view :
FPC slot:FE num  FE Bandwidth(Gbps) Status      Transmit Packets         Transmit Bytes
     1:0                   400         Up           111437126843        110991378335628
     1:1                   400         Up                      0                      0
     1:2                   400         Up                      0                      0
     1:3                   400         Up                      0                      0

Residual Transmit Statistics :
Packets :                    0 Bytes :                    0

Fabric Queue Statistics :
FPC slot:FE num    High priority(pkts)        Low priority(pkts)
     1:0                            0              111437126843
     1:1                            0                         0
     1:2                            0                         0
     1:3                            0                         0
FPC slot:FE num    High priority(bytes)      Low priority(bytes)
     1:0                              0          110991378335628
     1:1                              0                        0
     1:2                              0                        0
     1:3                              0                        0
Residual Queue Statistics :
    High priority(pkts)       Low priority(pkts)
                     0                        0
    High priority(bytes)      Low priority(bytes)
                       0                        0
Collapsed Forwarding Optimized Statistics :
          Optimal Packets             Non-optimal Packets
                        0                    111437126843
          Optimal Bytes               Non-optimal Bytes
                        0                  110991378335628

Logical interface af2.0 (Index 342) (SNMP ifIndex 689)
  Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ]  Encapsulation: ENET2
  Input packets : 532
  Output packets: 41761133660
```

```
    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: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255
    Protocol multiservice, MTU: Unlimited

  Logical interface af2.1 (Index 341) (SNMP ifIndex 690)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.2 ]  Encapsulation: ENET2
    Input packets : 536
    Output packets: 41764698890
    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: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.1
    Protocol multiservice, MTU: Unlimited

  Logical interface af2.32767 (Index 343) (SNMP ifIndex 692)
    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 af2 (with sub line cards or SLCs configured on MX2K-MPC11E)

```
user@router> show interfaces af2

Physical interface: af2, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 501
  Type: Ethernet, Link-level type: Ethernet, MTU: 1518, Speed: 3600000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Current address: 00:90:69:1d:20:47, Hardware address: 00:90:69:1d:20:47
  Last flapped   : 2021-03-02 22:40:47 PST (4d 21:31 ago)
  Input rate     : 4128 bps (9 pps)
  Output rate    : 5248 bps (12 pps)
```

```
Bandwidth      : 3600 Gbps
Peer GNF id    : 2
Collapsed Forwarding : Optimized
Peer GNF Forwarding element(FE) view :
FPC slot:FE num  FE Bandwidth(Gbps) Status      Transmit Packets         Transmit Bytes
     0:0                   400         Up               3927296              204218108
     0:1                   400         Up                911595               47346520
     0:2                   400         Up                   176                   8416
     0:3                   400         Up                 48627                4771730
     2:4                   500         Up                     0                      0
     2:5                   500         Up                     0                      0
     2:6                   500         Up                     0                      0
     2:7                   500         Up                     0                      0

Residual Transmit Statistics :
Packets :              4894889 Bytes :            256741776

Fabric Queue Statistics :
FPC slot:FE num    High priority(pkts)        Low priority(pkts)
     0:0                             0                   3927296
     0:1                             0                    911595
     0:2                             0                       176
     0:3                             0                     48627
     2:4                             0                         0
     2:5                             0                         0
     2:6                             0                         0
     2:7                             0                         0
FPC slot:FE num    High priority(bytes)      Low priority(bytes)
     0:0                               0                204218108
     0:1                               0                 47346520
     0:2                               0                     8416
     0:3                               0                  4771730
     2:4                               0                        0
     2:5                               0                        0
     2:6                               0                        0
     2:7                               0                        0
Residual Queue Statistics :
    High priority(pkts)       Low priority(pkts)
                      0                  4894889
    High priority(bytes)      Low priority(bytes)
                       0                256741776
Collapsed Forwarding Optimized Statistics :
          Optimal Packets             Non-optimal Packets
```

```
                    960612                          3927082
            Optimal Bytes               Non-optimal Bytes
                 52136510                        204208264

  Logical interface af2.0 (Index 361) (SNMP ifIndex 1032)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.1 ]  Encapsulation: ENET2
    Input packets : 8659876
    Output packets: 9897785
    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: Sendbcast-pkt-to-re
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2.0/24, Local: 192.0.2.1, Broadcast: 192.0.2.255
    Protocol iso, MTU: 1497
      Flags: Is-Primary
      Addresses, Flags: Is-Primary
        Local: 49.0002.0192.0168.0001
    Protocol mpls, MTU: 1488, Maximum labels: 3
      Flags: Is-Primary
    Protocol multiservice, MTU: Unlimited
      Flags: Is-Primary

  Logical interface af2.32767 (Index 364) (SNMP ifIndex 1033)
    Flags: Up SNMP-Traps 0x4004000 VLAN-Tag [ 0x0000.0 ]  Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Protocol multiservice, MTU: Unlimited
      Flags: None
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

Configuring Abstracted Fabric Interfaces Between a Pair of GNFs | 61

show system anomalies (GNF)

IN THIS SECTION

- Syntax | 220
- Description | 220
- Options | 220
- Required Privilege Level | 221
- Output Fields | 221
- Sample Output | 222
- Release Information | 222

Syntax

```
show system anomalies (all-anomalies | config | fru | system)
```

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

all-anomalies	Display the multiversion software incompatibilities from all categories—system, configuration, and FRU.
config	Display the feature incompatibilities between software versions.
fru	Display the FRU-level incompatibilities between software versions. This can be an incompatibility pertaining to the support for a specific FRU.

system Display the system-level incompatibilities between software versions. These include interprocess communication (IPC) message, CLI, or SNMP incompatibility.

Required Privilege Level

View

Output Fields

Table 18 on page 221 lists the output fields for the `show system anomalies` command. Output fields are listed in the approximate order in which they appear.

Table 18: show system anomalies gnf-id Output Fields

Field Name	Field Description
`Anomaly Type`	Shows the software incompatibility type. The following are the possible values: • SYS—Indicates system-specific incompatibilities. • FRU—Indicates FRU-specific incompatibilities. • CONFIG—Indicates feature-specific incompatibilities.
`Anomaly ID`	Shows the incompatibility ID.
`Default Action`	Shows the default actions associated with incompatibilities. The following are the possible values: • WARN—Indicates the presence of a minor incompatibility. It causes a warning. • ABORT—Indicates the presence of a major incompatibility. It causes a termination action.
`Class`	Indicates of the incompatibility is present in GNF, or BSYS, or both.

Table 18: show system anomalies gnf-id Output Fields *(Continued)*

Field Name	Field Description
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

command-name

```
user@router> show system anomalies

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

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

show system anomalies gnf-id | 169

Generic Guidelines for Using JDM Server Commands | 239

request virtual-network-functions | 249

request server authenticate-peer-server | 247

CHAPTER 7

Configuration Statements for JDM

virtual-network-functions | 225

server | 227

interfaces (Junos Node Slicing) | 229

routing-options (Junos Node Slicing) | 230

system login (Junos Node Slicing) | 232

root-login (JDM) | 234

vnf-license-supplement | 236

virtual-network-functions

IN THIS SECTION

- Syntax | 225
- Hierarchy Level | 225
- Description | 225
- Options | 226
- Required Privilege Level | 226
- Release Information | 226

Syntax

```
virtual-network-functions vnf-name {
    base-config base-config ;
    chassis-type chassis-type;
    id id;
    resource-template resource-template;
}
```

Hierarchy Level

[edit]

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

Release Information

Statement introduced in Junos OS Release 17.2R1.

RELATED DOCUMENTATION

server | 227

server

IN THIS SECTION

- Syntax | 227
- Hierarchy Level | 227
- Description | 227
- Options | 228
- Required Privilege Level | 228
- Release Information | 228

Syntax

```
server {
    interfaces {
        cb0 cb0-interface;
        cb1 cb1-interface;
        jdm-management jdm-management-interface;
        vnf-management gnf-management-interface;
    }
}
```

Hierarchy Level

[edit]

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

Release Information

Statement introduced in Junos OS Release 17.2R1.

RELATED DOCUMENTATION

virtual-network-functions | 225

interfaces (Junos Node Slicing)

IN THIS SECTION

- Syntax | 229
- Hierarchy Level | 229
- Description | 229
- Options | 230
- Required Privilege Level | 230
- Release Information | 230

Syntax

```
interfaces interface-name{
    unit unit {
        family (inet|inet6) {
            address address;
        }
    }
}
```

Hierarchy Level

[edit]

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

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

server | 227

virtual-network-functions | 225

routing-options (Junos Node Slicing)

IN THIS SECTION

Syntax | 231

- Hierarchy Level | 231
- Description | 231
- Options | 231
- Required Privilege Level | 232
- Release Information | 232

Syntax

```
routing-options {
    static {
        route route {
            next-hop next-hop;
        }
    }
}
```

Hierarchy Level

[edit]

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

Release Information

Statement introduced in Junos OS Release 17.2.

RELATED DOCUMENTATION

server | 227

virtual-network-functions | 225

system login (Junos Node Slicing)

IN THIS SECTION

- Syntax | 232
- Hierarchy Level | 233
- Description | 233
- Options | 233
- Required Privilege Level | 233
- Release Information | 234

Syntax

```
system login {
    user username ;
    class class-name;
```

```
    idle-timeout idle-timeout;
}
```

Hierarchy Level

[edit]

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

user *username* Username of the account.

class *class-name* Predefined login classes that JDM supports for non-root users.

- super-user
- operator
- read-only
- unauthorized

idle-timeout Maximum idle time before logout. We recommend that you set an idle time-out. Otherwise, if a user forgets to log out of a VNF console session, another user can log in to it without providing the access credentials.

- **Range:** 1 through 60 minutes

Required Privilege Level

Release Information

Statement introduced in Junos OS Release 18.3 R1.

RELATED DOCUMENTATION

Configuring Non-Root Users in JDM (Junos Node Slicing) | 45

Configuring JDM on the x86 Servers (External Server Model) | 44

root-login (JDM)

IN THIS SECTION

- Syntax | 234
- Hierarchy Level | 235
- Description | 235
- Default | 235
- Options | 235
- Required Privilege Level | 235
- Release Information | 235

Syntax

```
root-login (allow | deny );
```

Hierarchy Level

```
[edit system services ssh]
```

Description

Control user access to the JDM through SSH.

Default

`root-login allow` is the default.

Options

`allow`—Allow users to log in to the JDM as root through SSH.

`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.

Required Privilege Level

admin—To view this statement in the configuration.

admin-control—To add this statement to the configuration.

Release Information

Statement introduced in Junos OS Release 18.3.

RELATED DOCUMENTATION

Configuring JDM on the x86 Servers (External Server Model) | 44

vnf-license-supplement

IN THIS SECTION

- Syntax | 236
- Hierarchy Level | 236
- Description | 237
- Options | 237
- Required Privilege Level | 237
- Release Information | 237

Syntax

```
vnf-license-supplement {
    vnf-id gnf-id
    license-supplement-string augmented-ssrn
}
```

Hierarchy Level

[edit system]

Description

Allocate a MAC address to a licensed guest network function (GNF). For more details, see "Assigning MAC Addresses to GNF" on page 57.

Options

vnf-id *gnf-id*	Specify the GNF ID that will be assigned the licensed MAC address range. Each license can only be assigned to a single GNF.
license-supplement-string *augmented-ssrn*	Specify the augmented software support reference number (SSRN) string, which contains the MAC address details. Example: `RTU00023003204-01-AABBCCDDEEFF-1100-01-ABCD`.

Required Privilege Level

Release Information

Statement introduced in Junos OS Release 19.2R1.

CHAPTER 8

Operational Commands for JDM

Generic Guidelines for Using JDM Server Commands | 239

clear log (JDM) | 239

monitor list (JDM) | 241

monitor start (JDM) | 242

monitor stop (JDM) | 245

request server authenticate-peer-server | 247

request virtual-network-functions | 249

show virtual-network-functions | 251

show version vnf | 261

show version (JDM) | 264

show system cpu (JDM) | 267

show system mac-addresses (JDM) | 270

show system max-smbios-version (JDM) | 273

show system memory (JDM) | 274

show system network (JDM) | 277

show system max-smbios-version (JDM) | 283

show system random-mac-prefix (JDM) | 284

restart (JDM) | 286

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 gnf1 restart all-servers`.
- Append `server0` or `server1` to an operational command to take action on server0 or server1. Example: `request virtual-network-functions gnf1 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 | 251

request virtual-network-functions | 249

request server authenticate-peer-server | 247

clear log (JDM)

IN THIS SECTION

Syntax | 240

Description | 240

- Required Privilege Level | 240
- Output Fields | 240
- Sample Output | 240
- Release Information | 241

Syntax

```
clear log file-name
```

Description

Clear the system log or trace files.

Required Privilege Level

View

Output Fields

This command produces no output.

Sample Output

clear log

```
user@jdm> clear log syslog
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

monitor start (JDM) | 242

monitor list (JDM)

IN THIS SECTION

- Syntax | 241
- Description | 241
- Required Privilege Level | 242
- Output Fields | 242
- Sample Output | 242
- Release Information | 242

Syntax

```
monitor list
```

Description

Show status of monitored files.

Required Privilege Level

View

Output Fields

Sample Output

monitor list

```
user@jdm> monitor list
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

monitor start (JDM) | 242

monitor start (JDM)

IN THIS SECTION

- Syntax | 243
- Description | 243

- Additional Information | 243
- Required Privilege Level | 243
- Output Fields | 243
- Sample Output | 244
- Release Information | 245

Syntax

```
monitor start file-name
```

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

Output Fields

Table 19 on page 244 describes the output fields for the `monitor start` command. Output fields are listed in the approximate order in which they appear.

Table 19: 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
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

monitor stop (JDM) | 245

monitor stop (JDM)

IN THIS SECTION

- Syntax | 246
- Description | 246
- Additional Information | 246
- Required Privilege Level | 246
- Output Fields | 246

Release Information | 246

Syntax

```
monitor stop file-name
```

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

Output Fields

This command produces no output.

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

monitor start (JDM) | 242

request server authenticate-peer-server

IN THIS SECTION

- Syntax | 247
- Description | 247
- Required Privilege Level | 248
- Sample Output | 248
- Release Information | 248

Syntax

```
request server authenticate-peer-server
```

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

Sample Output

command-name

```
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.
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- show virtual-network-functions | 251
- request virtual-network-functions | 249

request virtual-network-functions

IN THIS SECTION

- Syntax | 249
- Description | 249
- Options | 249
- Required Privilege Level | 250
- Release Information | 250

Syntax

```
request virtual-network-functions vnf-name (add-image | all-servers | delete-image | force | restart | server |
start | stop)
```

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

Release Information

Command introduced in Junos OS Release 17.2R1.

The `force` option is available from Junos OS Release 17.4R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

show virtual-network-functions | 251

request server authenticate-peer-server | 247

show virtual-network-functions

IN THIS SECTION

- Syntax | 251
- Description | 251
- Options | 252
- Required Privilege Level | 252
- Output Fields | 252
- Sample Output | 254
- Sample Output | 255
- Sample Output | 255
- Release Information | 261

Syntax

```
show virtual-network-functions (all-servers | server| slot)
<vnf-name>
<vnf-name detail>
<vnf-name vnf-xml-profile>
```

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

Output Fields

Table 20 on page 252 lists the output fields for the `show virtual-network-functions` command.

Table 20: 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. • `Running`—The VNF is online and running. • `Shut off`—The VNF is in the shut down state.

Table 20: show virtual network functions Output Fields *(Continued)*

Field Name	Field Description
Liveness	Indicates the availability of VNF. • Up • Down
Name	Name of the VNF.
IP Address	IP address of the VNF.
Status	Status of a particular VNF. • 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.

Table 20: show virtual network functions Output Fields *(Continued)*

VNF CPU Utilization and Allocation Information	Shows the GNF CPU utilization details. See also: "show system cpu (JDM)" on page 267.

VNF Memory Information	Displays the following memory information about the GNFs: • 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: • 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)" on page 277.
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 gnf1


Virtual Machine Information
---------------------------
Name:               gnf1
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 gnf1 detail


VNF Information
---------------------------
ID                  1
Name:               gnf1
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
--------------------------------------- ---------------------- -----  --------
gnf1                                    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                  Rcvd Bytes    Rcvd packets Rcvd Error Rcvd Drop Trxd bytes   Trxd
Packets Trxd Error Trxd Drop
-------------------------- ------------  ------------ ---------- --------- ------------
------------ ---------- ---------
macvtap0                   19077011      335687       0          0         261601960
3545272      0          0
macvtap1                   99369778      689729       0          0         172763580
2786344      0          0
vnet1                      24686         527          0          0         11620
250          0          0
macvtap2                   428385        7405         0          0         861020752
10813152     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>
  <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>
```

Release Information

Command introduced in Junos OS Release 17.2R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

request virtual-network-functions | 249

request server authenticate-peer-server | 247

show version vnf

IN THIS SECTION

- Syntax | 261
- Description | 261
- Options | 262
- Required Privilege Level | 262
- Sample Output | 262
- Release Information | 264

Syntax

```
show version vnf vnf-name
```

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

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 AACL 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]
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247

show version (JDM)

IN THIS SECTION

- Syntax | 264
- Description | 265
- Options | 265
- Required Privilege Level | 265
- Sample Output | 265
- Release Information | 266

Syntax

```
show version (all-servers | server | vnf | all-vnfs | detail | brief)
```

Description

Display the version information about the Juniper Device Manager (JDM).

Options

all-servers Display the version details of the JDM instances on both the servers.

server Display the version details of the JDM instance on one specific server.

- **Range:** `0` through `1`

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*.

all-vnfs Display the version details for all the GNFs.

(detail | brief) Display the specified level of output.

Required Privilege Level

View

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
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247

show system cpu (JDM)

IN THIS SECTION

- Syntax | 267
- Description | 267
- Required Privilege Level | 267
- Output Fields | 268
- Sample Output | 268
- Release Information | 269

Syntax

```
show system cpu
```

Description

Display the CPUs and their usage status.

Required Privilege Level

View

Output Fields

Table 21 on page 268 describes the output fields for the `show system cpu` command. Output fields are listed in the approximate order in which they appear.

Table 21: 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 (External Server Model)

```
user@jdm> show system cpu

VNF CPU Utilization and Allocation Information
------------------------------------------------------------------------------
                                                                        -------------
VNF                                     CPU-Id(s)              Usage  Qemu
Pid
State
---------------------------------------- ---------------------- ------
--------
 -----------
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 cpu (In-chassis Junos Node Slicing)

```
user@jdm> show system cpu
> VNF CPU Utilization and Allocation Information
 ------------------------------------------------------------------------------------------
 VNF                                  CPU-Id(s)              Usage  Qemu Pid  State
 ------------------------------------ ---------------------- ------ --------  -----------
 gnf-bkhl                             2                      102%   21045     Running
 gnf-core                             3,4                    101%   22205     Running

 Free CPU-id(s) : 5
 Host Isolcpu(s): 1-5
 Emulator Pins  : 0
```

For in-chassis Junos node slicing, it is normal to see 100 percentage or higher CPU usage in the `Usage` column.

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247

show system mac-addresses (JDM)

IN THIS SECTION

- Syntax | 270
- Description | 270
- Required Privilege Level | 270
- Output Fields | 270
- Sample Output | 272
- Release Information | 273

Syntax

```
show system mac-addresses
```

Description

Display information related to the Juniper Device Manager (JDM) management interface MAC address and available MAC address range.

Required Privilege Level

View

Output Fields

Table 22 on page 271 describes the output fields for the `show system mac-addresses` command. Output fields are listed in the approximate order in which they appear.

Table 22: show system mac-addresses Output Fields

Field Name	Field Description
Management Interface MAC Addresses	
Domain	This could be a Juniper Device Manager (JDM) domain or guest network function (GNF) domain.
Interface	Interface name.
Management-MAC	The MAC address of the JDM or GNF.
Donor-GNF-ID	Displays name of the GNF which is the donor for the JDM management interface (jmgmt0) MAC address. This field is marked as not applicable (NA) for the GNFs listed in the column 'Management Interface MAC Addresses'. **NOTE**: • If there are no licensed GNFs in the system, no donor will be available for the `jmgmt0` MAC address. In that case, a virtual license is used and this field is marked as '-'. • For in-chassis Junos node slicing, `jmgmt0` and `fxp0` MAC addresses come from Juniper's licensed address space available on BSYS, not from a licensed GNF.
VNF MAC Address Pool	
Base-Mac-Address	Starting address of the MAC pool.
Size	Size of the MAC address pool.

Table 22: show system mac-addresses Output Fields *(Continued)*

Field Name	Field Description
Licensed	Indicates if the GNF is licensed or not. The following are the possible options: • Y • N
Donor	Indicates if the GNF is donor for jmgmt0 MAC address. The following are the possible options: • Y • N

Sample Output

show system mac-addresses

```
user@jdm> show system mac-addresses

Management Interface MAC Addresses
-------------------------------------------------------------------------------------
Domain                                    Interface   Management-MAC     Donor-GNF-ID
----------------------------------------  ---------   ------------------  ------------
JDM                                       jmgmt0      7C:E2:CA:FF:36:73   NA
gnf1                                      fxp0        7C:E2:CA:FF:36:74   NA
gnf3                                      fxp0        7C:E2:CA:FF:36:75   NA

MAC Blocks
-------------------------------------------------
GNF-ID  Base-Mac-Address    Size   Licensed  Donor
------  ------------------  -----  --------  -----
1       aa:bb:cc:dd:ee:00   4096   Y         N
3       00:90:69:3a:40:00   4096   N         N
```

Release Information

Command introduced in Junos OS Release 19.2R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

request virtual-network-functions | 249

request server authenticate-peer-server | 247

show system max-smbios-version (JDM)

IN THIS SECTION

- Syntax | 273
- Description | 273
- Required Privilege Level | 274
- Sample Output | 274
- Release Information | 274

Syntax

```
show system max-smbios-version
```

Description

Display the latest SMBIOS version supported by the Juniper Device Manager (JDM).

Required Privilege Level

View

Sample Output

show system max-smbios-version

```
user@jdm> show system max-smbios-version
Version : v2
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247

show system memory (JDM)

IN THIS SECTION

- Syntax | 275
- Description | 275
- Required Privilege Level | 275
- Output Fields | 275

Sample Output | 276
Release Information | 277

Syntax

```
show system memory
```

Description

Display the memory usage information about the host server, Juniper Device Manager (JDM), and guest network functions (GNF).

Required Privilege Level

View

Output Fields

Table 23 on page 276 describes the output fields for the `show system memory` command. Output fields are listed in the approximate order in which they appear.

Table 23: show system memory Output Fields

Field Name	Field Description
Memory Usage Information	Displays the following memory usage information about host server and JDM: • Total—Total memory. • Used—Used memory. • Free—Available memory.
VNF Memory Information	Displays the following memory information about the GNFs: • 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
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247

show system network (JDM)

IN THIS SECTION

- Syntax | 277
- Description | 278
- Required Privilege Level | 278
- Output Fields | 278
- Sample Output | 281
- Release Information | 282

Syntax

```
show system network
```

Description

Display the statistics information for physical interface, JDM interface, and interfaces per guest network function (GNF).

Required Privilege Level

View

Output Fields

Table 24 on page 278 describes the output fields for the `show system network` command. Output fields are listed in the approximate order in which they appear.

Table 24: 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.

Table 24: show system network Output Fields *(Continued)*

Field Name	Field Description
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.
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.

Table 24: show system network Output Fields *(Continued)*

Field Name	Field Description
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.
Trxd Packets	The number of packets transmitted.
Trxd Error	The packets transmitted with error.
Trxd Drop	The packets dropped during transmission.
Flags	Interface flags.

Table 24: show system network Output Fields *(Continued)*

Field Name	Field Description
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
0         1254382      16782        0          0
macvtap1                    ens3f1        00:f1:60:3d:20:24  561558       3727         0
0         803396       12958        0          0
vnet1                       bridge_jdm_vm 00:f1:60:3d:20:25  2888         32           0
0         2282         25           0          0
macvtap2                    enp3s0f2      00:f1:60:3d:20:26  152563       1080         0
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

.....................
```

Starting in Junos OS Release 19.2R1, the `show system network` output does not display information related to MAC address pool. For MAC address information use the command `show system mac-addresses`.

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

request virtual-network-functions | 249

request server authenticate-peer-server | 247

show system max-smbios-version (JDM)

IN THIS SECTION

- Syntax | 283
- Description | 283
- Required Privilege Level | 283
- Sample Output | 284
- Release Information | 284

Syntax

```
show system max-smbios-version
```

Description

Display the latest SMBIOS version supported by the Juniper Device Manager (JDM).

Required Privilege Level

View

Sample Output

show system max-smbios-version

```
user@jdm> show system max-smbios-version
Version : v2
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247

show system random-mac-prefix (JDM)

IN THIS SECTION

- Syntax | 285
- Description | 285
- Required Privilege Level | 285
- Sample Output | 285
- Release Information | 285

Syntax

```
show system random-mac-prefix
```

Description

Display the random MAC prefix generated by JDM. This MAC prefix is used to prevent any MAC block overlap between guest network functions (GNFs) managed by two different JDMs.

NOTE: Before creating a GNF, you must ensure that you have configured commit synchronization as part of JDM interfaces configuration so that the random MAC prefixes generated by the JDM instances are in sync.

Required Privilege Level

View

Sample Output

show system random-mac-prefix

```
user@jdm> show system random-mac-prefix
Mac-Prefix : 0x1e
```

Release Information

Command introduced in Junos OS Release 19.2R1.

RELATED DOCUMENTATION

Generic Guidelines for Using JDM Server Commands | 239

request virtual-network-functions | 249

request server authenticate-peer-server | 247

restart (JDM)

IN THIS SECTION

- Syntax | 286
- Description | 286
- Options | 287
- Required Privilege Level | 287
- Output Fields | 288
- Sample Output | 288
- Release Information | 288

Syntax

```
restart (gracefully | immediately | soft)
<jdmd>
<jdmmon>
<jinventoryd>
<jlinkmon>
```

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.

NOTE: Starting in Junos OS Release 19.2R1, JDM does not support the `jinventoryd` option in the `restart` command.

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.

Required Privilege Level

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
```

Release Information

Command introduced in Junos OS Release 17.4R1.

RELATED DOCUMENTATION

- Generic Guidelines for Using JDM Server Commands | 239
- request virtual-network-functions | 249
- request server authenticate-peer-server | 247