# In-Chassis Junos Node Slicing on MX480

Configuration Guide

August 2019

Juniper Networks, Inc.
1133 innovation Way
Sunnyvale California 94089
USA
408-745-2000
www.juniper.net

# Contents



 

# 1. Junos Node Slicing Overview

Junos node slicing allows a single MX chassis to be partitioned into multiple independent routers, each with its own control, data, and management planes. For a more complete description, refer to https://www.juniper.net/documentation/en_US/junos/topics/concept/node-slicing-understanding.html.

## A. In-chassis Junos Node Slicing Components

The enabler for in-chassis Junos node slicing is the MX Routing Engine model RE-S-2X00x6-128 for MX480/MX960, and RE-MX200x8-128G for the MX2010/MX2020. These Routing Engines have a virtualized architecture where Junos runs as a virtual machine (VM) over a Linux based host, and they support the ability to run additional VM instances. In the case of in-chassis Junos node slicing, each Routing Engine can host up to three other Junos VMs which serve as the control planes for the independent routers.

The diagram below shows the various components of in-chassis Junos node slicing and represents the logical view of the configuration. Each node slice, called a guest network function (GNF), is comprised of a master-backup pair of Routing Engine (RE) VMs, and one or more line cards in the MX chassis. For example, for GNF 'PE1', the control plane consists of the master Routing Engine Junos VM running on RE0 and the backup Routing Engine Junos VM running on RE1, while the data plane consists of the MPC3E-NG line card in slot 0.

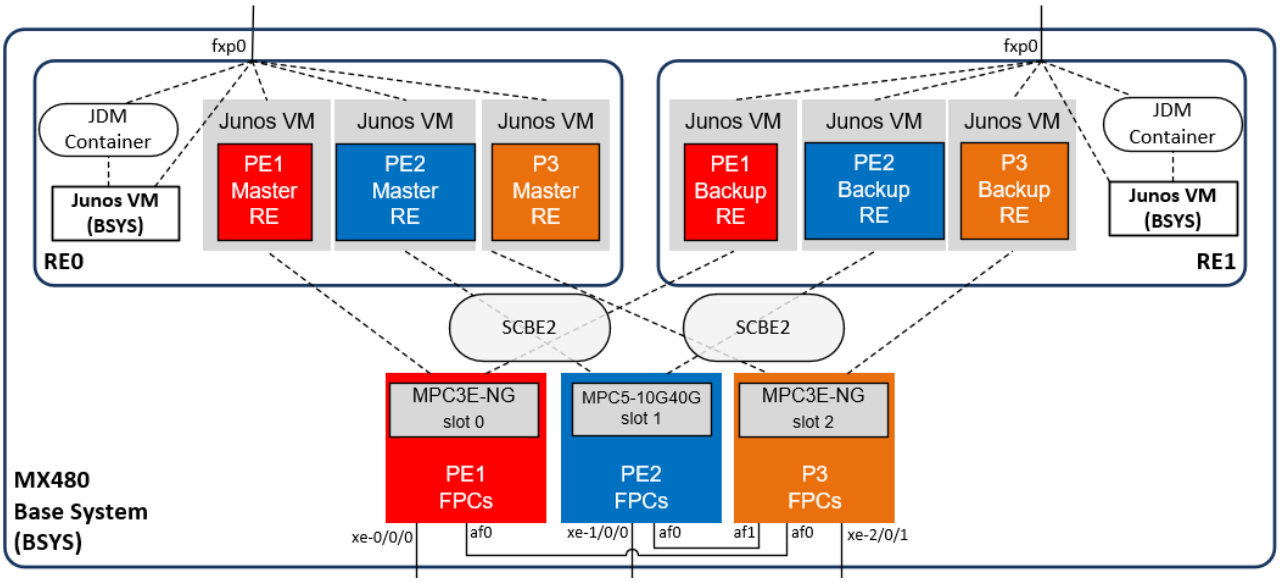


*Figure 1: Components of in-chassis Junos node slicing*

Juniper Device Manager (JDM) software manages the GNF Junos Routing Engine VMs running on the MX Series router's physical Routing Engine. JDM is a Linux container that provides a Junos-like CLI to orchestrate the GNF Junos VMs. It communicates with the libvirtd daemon which is used to manage the KVM/QEMU virtualization technologies running on the host Routing Engine.

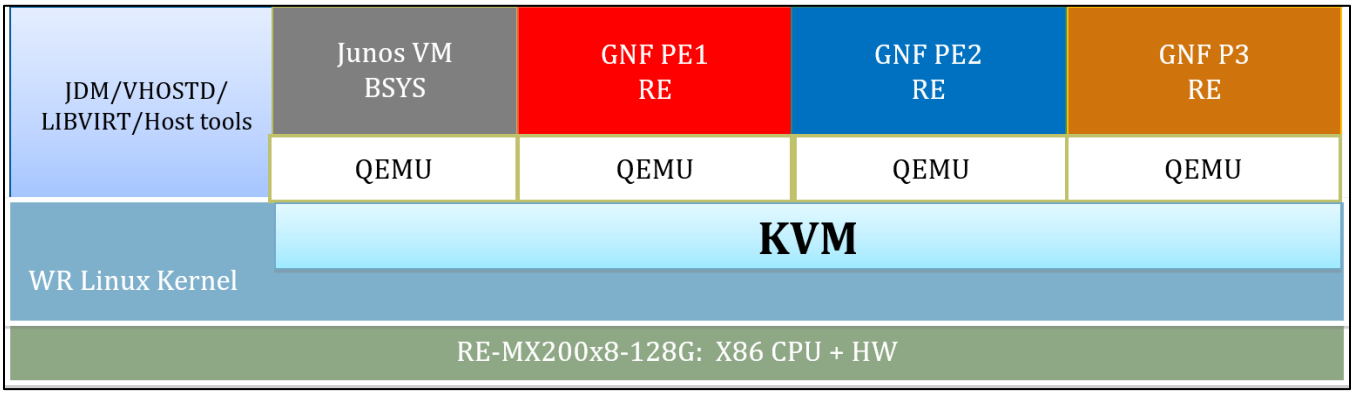


*Figure 2: Host Routing Engine Software Architecture*

Each of the Junos VMs and JDM are connected internally to a Linux bridge that maps to the physical management port on the Routing Engine. Therefore, you can access each VM and JDM directly via SSH. Other internal mechanisms, which are beyond the scope of this guide, provide logical separation between the GNFs.

The base system (BSYS) represents the underlying MX hardware, including line cards, fabrics, Routing Engines, power supplies, and fans, and Junos software running as a VM. The BSYS is configured to map the line cards to each GNF. Abstracted Fabric (`af`) interfaces are configured at the BSYS to provide connectivity between GNFs via the fabric.

In this scenario, the BSYS is an MX480 chassis containing three FPCs, two SCBE2 control boards, and the aforementioned RE-S-2X00x6 Routing Engines which are required for in-chassis Junos node slicing.

```
root@BSYS1-MX480> show chassis hardware | match "Routing|CB|FPC"
Routing Engine 0 REV 05   750-072925   CAMC2384          RE-S-2X00x6
Routing Engine 1 REV 05   750-072925   CAMC2375          RE-S-2X00x6
CB 0             REV 07   750-055976   CAFP6426          Enhanced MX SCB 2
CB 1             REV 07   750-055976   CAFD6595          Enhanced MX SCB 2
FPC 0            REV 23   750-054901   CAEH6682          MPC3E NG HQoS
FPC 1            REV 56   750-045715   CADG7365          MPC5E 3D Q 24XGE+6XLGE
FPC 2            REV 23   750-054902   CAEK8106          MPC3E NG PQ & Flex Q
```

## B. Topology

In the following example, the BSYS is part of a larger topology. The three GNFs are connected to other devices including traffic generator ports and another physical MX Series router. GNF3 ‘P3’ is connected to the other two GNFs via unique `af` interfaces.

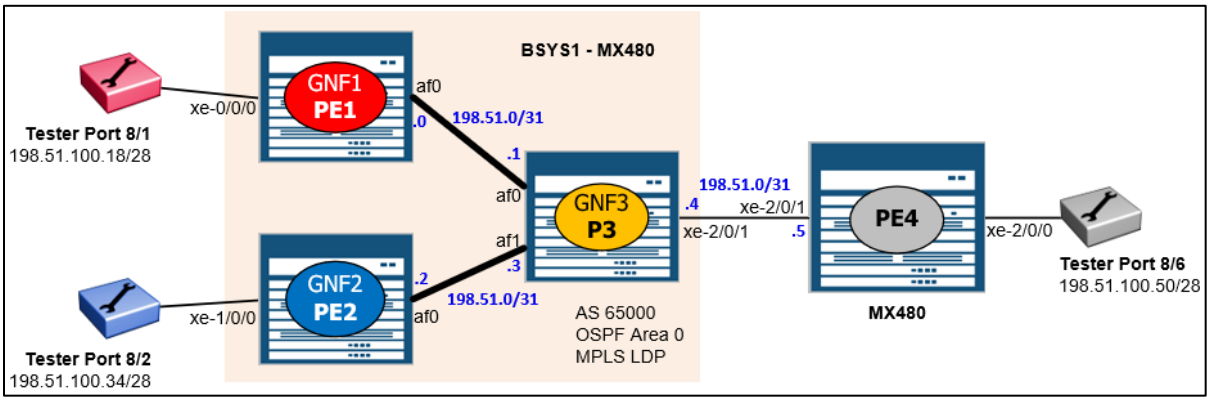


*Figure 3: Topology - In-chassis Junos node slicing*

The details of the configuration are not very important for the purposes of this guide. However, it is important to know that each of the GNFs is configured and managed similar to a physical MX Series router. And the `af` interfaces are configured and managed like any other interface in Junos.

# 2. Initial Preparation

## A. Juniper Software Packages

You need the following three software packages to implement Junos node slicing:

1. Base system (BSYS) - this is the software running on the MX chassis. It is the standard Junos software and runs as a VM. For in-chassis Junos node slicing, the minimum Junos version required is 19.1R1.
2. Juniper Device Manager (JDM) - this is a software package that includes the Linux container that is used to orchestrate (that is, create, delete, modify) the GNF Routing Engine VMs. For in-chassis Junos node slicing, the name of this file starts with 'jns-jdm-vmhost-...'.
3. GNF Routing Engine Image - this refers to the Junos Routing Engine image that is used to create the GNF Routing Engine VM. The filename starts with 'junos-install-ns-mx-x86-64-...'

You can download each of these software packages from the Juniper support site. The JDM and GNF Routing Engine images can be found at https://www.juniper.net/support/downloads/solutions/jns/.

## B. BSYS

On the MX Series router, install the BSYS software by following the standard Junos software installation procedure. For demonstration purposes, you need not obtain a license because the Junos node slicing license is honor-based. For this example, the MX chassis is running the Junos version 19.1R1.

Initially you should configure the BSYS with the basics for management which include configuring an IP address and enabling the SSH service. If the BSYS contains two Routing Engines, enable the basic redundancy mechanisms, including Graceful Routing Engine Switchover (GRES) and commit synchronization. This is important as the initial preparation on the BSYS is applicable to both Routing Engines.

### i. Reduce BSYS Junos VM Footprint

To prepare the MX BSYS for in-chassis Junos node slicing, you need to first reduce the Routing Engine resources used by the default Junos VM. This allows more resources to be available for the GNF Routing Engine VMs.

Use the following steps to reduce the Routing Engine resources used by the default Junos VM:

1. From the Junos CLI, enter configuration mode and enter the 'set vmhost resize compact' command as shown in the following example:

```
root@BSYS1-MX480> configure
Entering configuration mode

{master}[edit]
root@BSYS1-MX480# set vmhost resize vjunos compact

{master}[edit]
root@BSYS1-MX480# commit synchronize and-quit
re0:
[edit]
  'vmhost'
```

```
    warning: VMHOST configuration for 'resize vjunos compact' has been changed. A vmhost reboot is
mandatory. Please reboot *ALL* routing engines NOW using 'request vmhost reboot'. Continuing
without a reboot might result in unexpected system behavior.
configuration check succeeds
re1:
[edit]
  'vmhost'
    warning: VMHOST configuration for 'resize vjunos compact' has been changed. A vmhost reboot is
mandatory. Please reboot *ALL* routing engines NOW using 'request vmhost reboot'. Continuing
without a reboot might result in unexpected system behavior.
commit complete
re0:
commit complete
Exiting configuration mode
```

2. As instructed, from Junos CLI operational mode, reboot both Routing Engines.

```
request vmhost reboot routing-engine both
```

Once the 'footprint' of the VM is reduced, the BSYS is assigned the following resources:

- 1 CPU core
- 16 GB RAM
- 14 GB Storage

3. Use the CLI command 'show vmhost status' to verify that the default Junos VM is in the 'Compact' state:

```
root@BSYS1-MX480> show vmhost status re0
re0:
--------------------------------------------------------------------------

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

vJunos Resource Status: Compact


root@BSYS1-MX480> show vmhost status re1
re1:
--------------------------------------------------------------------------

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

vJunos Resource Status: Compact
```

## C. JDM

Download the JDM software to the /var/tmp/ directory on both Routing Engines in the MX BSYS. For this example, the 19.1R1 version of JDM software for in-chassis Junos node slicing is used.

```
file copy scp://admin@192.0.2.106:/home/admin/ftp/sw/19.1/jns-jdm-vmhost-19.1-R1.6.x86_64.rpm
/var/tmp
file copy /var/tmp/jns-jdm-vmhost-19.1-R1.6.x86_64.rpm re1:/var/tmp/
```

## i. JDM Installation

Install JDM on RE0 by using the following Junos CLI operational mode command. Note that you need to execute this command on RE1 as well (not shown below for brevity). You can use the 'request routing-engine login other-routing-engine' CLI command to connect to the backup RE1.

```
root@BSYS1-MX480> request vmhost jdm add /var/tmp/jns-jdm-vmhost-19.1-R1.6.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: jns-jdm-vmhost-19.1-R1.6.x86_64.rpm
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
Installation Successful !
```

## ii. Initial JDM Configuration

Initial JDM configuration involves the following steps:

1. From the Junos CLI, start the JDM container on both Routing Engines. After a minute or so, verify that JDM is 'Running' on each Routing Engine (only RE0 output is shown below for brevity):

```
root@BSYS1-MX480> 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

root@BSYS1-MX480> show vmhost jdm status
JDM Information
---------------------------
Package    :  jns-jdm-vmhost-19.1-R1.6.x86_64
Status     :  Running
PID        :  26177
Free Space :  129633 (MiB)
```

2. Log in to JDM using the Junos CLI operational mode command as shown below. Note that this command is only available to users with 'root' privilege.

```
root@BSYS1-MX480> 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-R1.6.x86_64                               *
* Free Disk Space on JDM's root-fs ("/"): 98043(MiB)                         *
******************************************************************************
Last login: Wed Jun 26 11:54:02 2019 from 192.168.1.1
root@bsys1-jdm0:~# cli
root@bsys1-jdm0> show version
Hostname: jdm
Model: junos_node_slicing
Server slot : 0
JDM package version : 19.1-R1.6
Host Software [Wind River Linux 6.0.0.31 (OVP)]
JDM container Software [Ubuntu 14.04.1 LTS]
JDM daemon jdmd [Version: 19.1R1.6-secure]
JDM daemon jdmmon [Version: 19.1R1.6-secure]
Host daemon jlinkmon [Version: 19.1R1.6-secure]
```

The output above shows the JDM version and that the 'Server slot' is '0'. This means that this instance of JDM is running on BSYS RE0. Note that previous versions of Junos node slicing required external servers. Therefore, the 'server' terminology is present in the CLI.

3. Configure the management interfaces of the JDM instances. The JDM management interface is mapped to physical management port of the BSYS' Routing Engine. Therefore, it is assigned a unique IP address on the same subnet as the BSYS Junos VM.

   The JDM management interface is named 'jmgmt0'. Since the JDM CLI is similar to Junos, enter configuration mode from the JDM instance on RE0. Then set the root password and the apply-groups corresponding to the BSYS routing engines, 'server0' and 'server1':

```
set system root-authentication plain-text-password

set groups server0 interfaces jmgmt0 unit 0 family inet address 192.0.2.207/24
set groups server0 routing-options static route 0.0.0.0/0 next-hop 192.0.2.254

set groups server1 interfaces jmgmt0 unit 0 family inet address 192.0.2.208/24
set groups server1 routing-options static route 0.0.0.0/0 next-hop 192.0.2.254

set groups server0 system host-name bsys1-jdm0
set groups server1 system host-name bsys1-jdm1

set apply-groups server0
set apply-groups server1

commit and
```

   The JDM instances running on the BSYS Routing Engines are not yet synchronized. Therefore, the 'root-authentication' must be set and committed on the JDM instance running on RE1 as well.

```
set system root-authentication plain-text-password
```

4. From the JDM operational mode command line, initiate the exchange of SSH keys with the JDM peer. It is important to run this command on the JDM instance on both Routing Engines.

```
root@bsys1-jdm0> request server authenticate-peer-server
The authenticity of host '192.168.2.245 (192.168.2.245)' can't be established.
ECDSA key fingerprint is 64:fe:97:b6:94:a3:24:eb:00:26:54:38:ad:cc:ce:4f.
```

```
Are you sure you want to continue connecting (yes/no)? yes
/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@192.168.2.245's password: <password>

Number of key(s) added: 1
<snip>
```

The result is a trusted connection between the two JDM instances.

5. From the JDM instance running on RE0, configure JDM to ‘commit synchronize’ and verify that the configuration change is applied to both ‘server0’ and ‘server1’:

```
root@bsys1-jdm0> configure
Entering configuration mode

[edit]
root@bsys1-jdm0# set system commit synchronize

[edit]
root@bsys1-jdm0# commit and-quit
server0:
configuration check succeeds
server1:
commit complete
server0:
commit complete
Exiting configuration mode
```

Now that the configuration of the two JDM instances are synchronized, the “groups” configuration appears in the JDM configuration running on RE1.

6. Verify that the various JDM interfaces and connections are up. Many of the interfaces are internal and should automatically be ‘up’. Confirm that the configured ‘jmgmt0’ interface is ‘up’ and that the ‘JDM to JDM’ links are ‘up’ indicating that they are in sync.

```
root@bsys1-jdm0> show server connections
Component               Interface               Status  Comments
Host to JDM port        jnpr-int-br             up
Physical CB0 port       eth1                    up
Physical CB1 port       eth2                    up
Physical JDM mgmt port  eth0                    up
JDM-GNF bridge          bridge_jdm_vm           up
JDM mgmt port           jmgmt0                  up
JDM to HOST port        bme1                    up
JDM to GNF port         bme2                    up
JDM to JDM link0*       cb0                     up      StrictKey peer SSH - OK
JDM to JDM link1        cb1                     up      StrictKey peer SSH - OK
```

At this point JDM is ready to be used to create the GNF Routing Engine VMs.

# 3. Create GNFs

There are two parts to creating the Guest Network Functions. First, the BSYS is configured to assign a unique identifier for each GNF and to associate each line card with a GNF. Then, using JDM, the GNF Routing Engine VMs are created and configured to map to each GNF based on the unique identifier.

## A. BSYS

On the MX BSYS, define three GNFs (IDs 1 - 3) and associate a line card with each GNF. Note that a given line card can only be associated with a single GNF. In addition, define the `af` interfaces which provide point to point connectivity between GNFs. For reference, view Figure 1: Components of in-chassis Junos node slicing that shows the mapping of the line cards to each GNF, and Figure 3: Topology - In-chassis Junos node slicing which shows the `af` interface assignments.

```
set chassis network-slices guest-network-functions gnf 1 fpcs 0
set chassis network-slices guest-network-functions gnf 2 fpcs 1
set chassis network-slices guest-network-functions gnf 3 fpcs 2

set chassis network-slices guest-network-functions gnf 1 af0 peer-gnf id 3 af0
set chassis network-slices guest-network-functions gnf 2 af0 peer-gnf id 3 af1
set chassis network-slices guest-network-functions gnf 3 af0 peer-gnf id 1 af0
set chassis network-slices guest-network-functions gnf 3 af1 peer-gnf id 2 af0
```

Once this configuration is 'commit'ed, the FPCs automatically restart.

The state of the line cards transitions from 'Offline - GNF initiated Restart' to 'Present'. Use the command 'show chassis fpc' to monitor the status. The line cards are no longer controlled by the BSYS Routing Engine and will later be controlled by the GNF Routing Engine VM. Once the GNF Routing Engines are up and running, the line cards state transitions to 'Online'. Note the addition of the 'GNF' column at the far right of the output.

```
root@BSYS1-MX480> show chassis fpc
                    Temp  CPU Utilization (%)   CPU Utilization (%)  Memory    Utilization (%)
Slot State           (C)  Total  Interrupt      1min   5min   15min  DRAM (MB) Heap     Buffer    GNF
  0  Offline    ---GNF initiated Restart---                                                          1
  1  Offline    ---GNF initiated Restart---                                                          2
  2  Offline    ---GNF initiated Restart---                                                          3
  3  Empty
  4  Empty
  5  Empty
```

The current state of the GNFs from the perspective of the BSYS is also reflected in the 'show chassis network-slices' output:

```
root@BSYS1-MX480> show chassis network-slices
guest-network-functions:
GNF    Description             State          Uptime
 1                              Empty
 2                              Empty
 3                              Empty
```

## B. JDM

### i. Download GNF Routing Engine Image

First, from the shell of the JDM container on RE0, copy the GNF Routing Engine image, as described in the 'Initial Preparation' section above, to the '/var/jdm-usr/gnf-images' directory.

```
root@bsys1-jdm0:~# scp admin@192.0.2.106:ftp/sw/19.1/junos-install-ns-mx-x86-64-19.1R1.6.tgz
/var/jdm-usr/gnf-images
admin@192.0.2.106's password:
junos-install-ns-mx-x86-64-19.1R1.6.tgz                                    19%  524MB
106.3MB/s
```

This software image will be referenced when using JDM to create the GNF Routing Engine VMs. In this example the version of the GNF Routing Engine VM matches the BSYS version. However, note that one of the attributes of Junos node slicing is the ability for each GNF to run a different version of Junos, which can be higher or lower than the BSYS version by up to two versions.

Confirm that the GNF VM image has been downloaded to the /var/jdm-usr/gnf-images directory. In this example three GNF Routing Engine VMs will be created. The same image is used when creating the VMs as JDM will make a unique copy for each instance:

```
root@bsys1-jdm0:~# ls -l /var/jdm-usr/gnf-images/
total 2732868
-rw-r--r--. 1 root root 2798451080 May 23 11:24 junos-install-ns-mx-x86-64-19.1R1.6.tgz
```

### ii. Create GNF Routing Engine VMs

The next step is to use the image file to create the GNF Routing Engine VMs. These VMs are referred to as Virtual Network Functions (or VNFs) in JDM.

From the JDM operational mode CLI, use the following command to create the VNF corresponding to GNF ‘PE1’. Note the ‘all-servers’ parameter which instructs JDM to create the VNF on both RE0 (server0) and RE1 (server1).

```
root@bsys1-jdm0> request virtual-network-functions add-image /var/jdm-usr/gnf-images/junos-install-
ns-mx-x86-64-19.1R1.6.tgz all-servers PE1
server0:
--------------------------------------------------------------------------
Added image: /vm-primary/PE1/PE1.img

server1:
--------------------------------------------------------------------------
Added image: /vm-primary/PE1/PE1.img
```

As indicated by the output above, the result is that a unique disk image named ‘PE1’ is created.

Repeat the previous step in order to create the GNF Routing Engine VMs for the other two GNFs named ‘PE2’ and ‘P3’.

```
request virtual-network-functions add-image /var/jdm-usr/gnf-images/junos-install-ns-mx-x86-64-
19.1R1.6.tgz all-servers PE2

request virtual-network-functions add-image /var/jdm-usr/gnf-images/junos-install-ns-mx-x86-64-
19.1R1.6.tgz all-servers P3
```

### iii. Configure and Spawn GNF Routing Engine VMs

In JDM configuration mode, set the attributes associated with each VM. The parameters include:

- VNF name - this must match the image name that was specified when creating the VM, ex. ‘PE1’.
- GNF ID - this must match the ID configured on the BSYS, ex. ‘1’, ‘2’, or ‘3’.
- Base configuration – a configuration file that is loaded by the GNF Routing Engine VM upon initialization. The configuration file must be present in the ‘/var/jdm-usr/gnf-config/’ directory. This is an optional setting.

- Resource template - the number of CPUs and RAM consumed by the GNF Routing Engine VM.

| Template Name | CPU Cores | Memory (GB) |
|---|---|---|
| 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 |

There are a few factors to be considered before setting the resource template for each GNF. The first is the application the GNF will be used for and its scaling requirements. The second is the resources available on the host (BSYS) Routing Engine itself.

To view the current resource usage, use the 'show system' CLI commands shown below. Note that there are 4 CPUs available for the GNF Routing Engine VMs on the MX480's host Routing Engine, and about 106 GB of RAM.

```
root@bsys1-jdm0> show system cpu

Free CPU-id(s) : 5,3,4,2
Host Isolcpu(s): 1-5
Emulator Pins  : 0

root@bsys1-jdm0> show system memory

Memory Usage Information
--------------------------
       Total  Used   Free
       ------ ------ ------
Host:  125.6G 17.2G  105.8G

JDM :  2G     1.9G   3.6M
```

For the example in this guide, the following configuration is applied:

```
set virtual-network-functions PE1 id 1
set virtual-network-functions PE1 resource-template 1core-16g
set virtual-network-functions PE1 base-config /var/jdm-usr/gnf-config/pe1.cfg

set virtual-network-functions PE2 id 2
set virtual-network-functions PE2 resource-template 2core-16g
set virtual-network-functions PE2 base-config /var/jdm-usr/gnf-config/pe2.cfg

set virtual-network-functions P3 id 3
set virtual-network-functions P3 resource-template 1core-32g
set virtual-network-functions P3 base-config /var/jdm-usr/gnf-config/p3.cfg
```

# 4. Verification

## A. JDM

Once the JDM configuration is committed, each VM is spawned and initialized, similar to a Junos Routing Engine booting up. This process takes several minutes; so be patient and monitor the VNF status periodically using the 'show' command shown below. The "Liveness" state will be 'down' until initialization is complete. Again, note the 'all-servers' parameter which allows you to view the status on both Routing Engines:

```
root@bsys1-jdm0> show virtual-network-functions all-servers
server0:
------------------------------------------------------------------------------
ID       Name                                              State      Liveness
------------------------------------------------------------------------------
1        PE1                                               Running    up
2        PE2                                               Running    up
3        P3                                                Running    up

server1:
------------------------------------------------------------------------------
ID       Name                                              State      Liveness
------------------------------------------------------------------------------
1        PE1                                               Running    up
2        PE2                                               Running    up
3        P3                                                Running    up
```

To view the VNF initialization process from the virtual console, use the following command (Note: press 'Ctrl'-']' to exit the console and return to JDM CLI):

```
root@bsys1-jdm0> request virtual-network-functions console PE1
Connected to domain PE1
Escape character is ^]

<Console Output>
```

To view the details of a specific VNF, use the following command:

```
root@bsys1-jdm0> show virtual-network-functions PE1
VNF Information
---------------------------
ID                  1
Name:               PE1
Status:             Running
Liveness:           up
IP Address:         192.168.2.1
Cores:              1
Memory:             16GiB
Resource Template:  1core-16g
Qemu Process id:    28239
SMBIOS version:     v1
```

Also, since the VNFs have been provisioned, view the current resources consumed. The output below shows that all the CPUs are in use and free memory is down to 41.2 GB.

```
root@bsys1-jdm0> show system cpu

VNF CPU Utilization and Allocation Information
------------------------------------------------------------------------------------------------
VNF                                    CPU-Id(s)               Usage  Qemu Pid  State
-------------------------------------- ----------------------- ------ --------  -----------
PE1                                    5                       102%   28239     Running
PE2                                    3,4                     101%   29987     Running
```

```
P3                                      2                     102%   31555     Running

Free CPU-id(s) : None
Host Isolcpu(s): 1-5
Emulator Pins  : 0

root@bsys1-jdm0> show system memory

Memory Usage Information
--------------------------
       Total  Used   Free
       ------ ------ ------
Host:  125.6G 81.8G  41.2G

JDM :  2G     1.9G   13.9M

VNF Memory Information
-----------------------------------------------------------------
Name                                                Actual Resident
--------------------------------------------------- ------ --------
PE1                                                 16.0G  16.1G
PE2                                                 16.0G  16.1G
P3                                                  32.0G  32.1G
```

## B. Connecting to GNF

When the GNF Routing Engine VMs were spawned, they initialized with a configuration file that contained IP address settings for the management 'fxp0' interface. Therefore, an SSH session can be established directly to any of the GNFs.

### i. GNF 'show chassis'

After establishing an SSH session to GNF 'PE1', the 'show chassis' commands indicate that only FPC 0 is associated with this GNF. In addition, all the shared BSYS hardware components are displayed including the Routing Engines, SCBs, power supplies, and fans. Information at the end of the output indicates that the GNF is type "MX480-GNF" with two Routing Engine VMs.

```
admin1@PE1> show chassis fpc
bsys-re0:
--------------------------------------------------------------------------
                    Temp  CPU Utilization (%)   CPU Utilization (%)  Memory    Utilization (%)
Slot State           (C)  Total  Interrupt      1min   5min   15min  DRAM (MB) Heap     Buffer    GNF
  0  Online           43     10          0       10     10     10    3584        6         25       1

{master}
admin1@PE1> show chassis hardware
bsys-re0:
--------------------------------------------------------------------------
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                JNxxxxxxxxxx      MX480
Midplane         REV 09   750-047862   ACRF5325          Enhanced MX480 Midplane
FPM Board        REV 02   710-017254   ABBA7086          Front Panel Display
PEM 2            Rev 03   740-027736   QCS1018T00W       DC 2.4kW Power Entry Module
PEM 3            Rev 03   740-027736   QCS1018T01L       DC 2.4kW Power Entry Module
Routing Engine 0 REV 05   750-072925   CAMC2384          RE-S-2X00x6
Routing Engine 1 REV 05   750-072925   CAMC2375          RE-S-2X00x6
CB 0             REV 07   750-055976   CAFP6426          Enhanced MX SCB 2
CB 1             REV 07   750-055976   CAFD6595          Enhanced MX SCB 2
FPC 0            REV 23   750-054901   CAEH6682          MPC3E NG HQoS
  CPU            REV 12   711-045719   CAEN1731          RMPC PMB
  MIC 0          REV 07   750-033307   CAAZ1721          10X10GE SFPP
    PIC 0                 BUILTIN      BUILTIN           10X10GE SFPP
      Xcvr 0     REV 01   740-031980   ARE0Z7E           SFP+-10G-SR
  MIC 1          REV 19   750-033199   CAAJ1843          1X100GE CFP
```

```
    PIC 2                 BUILTIN      BUILTIN          1X100GE CFP
      Xcvr 0     REV 01   740-035329   UPD0F6L          CFP-100G-SR10
Fan Tray                                                Enhanced Left Fan Tray

gnf1-re0:
--------------------------------------------------------------------------
Chassis                                 GN5D14DC6FBD     MX480-GNF
Routing Engine 0                                         RE-GNF-2000x1
Routing Engine 1                                         RE-GNF-2001x1
```

## ii. Abstracted Fabric Interface Configuration and Management

Each GNF is an independent Junos router and has its own configuration similar to a standalone Junos device. In this example, Abstracted Fabric (`af`) interfaces were previously configured on the BSYS. These interfaces automatically appear in the GNF and are configured and managed like any other standard Junos interface type. For example, on GNF PE1 the af0 interface provides connectivity to GNF P3 interface af0 via OSPF and LDP:

```
set interfaces af0 description "To P3"
set interfaces af0 unit 0 family inet address 198.51.100.0/31
set interfaces af0 unit 0 family mpls

set protocols ospf area 0.0.0.0 interface af0.0 interface-type p2p
set protocols mpls interface af0.0
set protocols ldp interface af0.0
```

The following 'show' commands indicate that OSPF and LDP adjacencies have been formed via the af0 interface:

```
admin1@PE1> show ospf neighbor
Address          Interface              State     ID               Pri  Dead
198.51.100.1         af0.0              Full      203.0.113.3      128    39

{master}
admin1@PE1> show ldp neighbor
Address                            Interface       Label space ID     Hold time
198.51.100.1                           af0.0       203.0.113.3:0          13
```

The `af` interface information shows more details about the connection to its peer. In this case, GNF PE1 contains a single MPC3E line card which has a single PFE supporting 130 Gbps throughput. Its af0 interface uses this PFE to connect to "Peer GNF id" 3, or GNF P3. The GNF peer's FPC in slot '2' is also an MPC3E with 'Bandwidth' of '130 Gbps'. The point to point connection between the GNFs is 'Up'.

```
admin1@PE1> show interfaces terse | match af
af0                     up    up
af0.0                   up    up   inet     198.51.100.0/31

admin1@PE1> show interfaces af0
Physical interface: af0, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 501
  Description: To P3
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 130000mbps
  Device flags   : Present Running
  Interface flags: Internal: 0x4000
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 00:90:69:00:a5:36, Hardware address: 00:90:69:00:a5:36
  Last flapped   : 2019-06-27 11:14:00 EDT (01:28:42 ago)
  Input rate     : 323074096 bps (32001 pps)
  Output rate    : 323068920 bps (31999 pps)
  Bandwidth      : 130 Gbps
  Peer GNF id    : 3
  Peer GNF Forwarding element(FE) view :
  FPC slot:FE num  FE Bandwidth(Gbps) Status      Transmit Packets         Transmit Bytes
```

```
      2:0                   130         Up               20022659           25268595658

  Residual Transmit Statistics :
  Packets :                    0 Bytes :                    0

  Fabric Queue Statistics :
  FPC slot:FE num    High priority(pkts)        Low priority(pkts)
       2:0                             0                 20022659
  FPC slot:FE num    High priority(bytes)       Low priority(bytes)
       2:0                             0              25268595658
<snip>
```

For reference, a similar output from GNF P3, which has 2 `af` interfaces, is shown below. The `af0` interface on P3 connects to PE1 which has 130 Gbps of bandwidth to P3. The `af1` interface connects to PE2 which has 240 Gbps of bandwidth to P3. GNF PE2 contains an MPC5E line card which has two 120 Gbps PFEs.

```
admin3@P3> show interfaces af0 | find Band
  Bandwidth      : 130 Gbps
  Peer GNF id    : 1
  Peer GNF Forwarding element(FE) view :
  FPC slot:FE num  FE Bandwidth(Gbps) Status      Transmit Packets         Transmit Bytes
       0:0                   130         Up               81437535           102774169170
<snip>

admin3@P3> show interfaces af1 | find Band
  Bandwidth      : 240 Gbps
  Peer GNF id    : 2
  Peer GNF Forwarding element(FE) view :
  FPC slot:FE num  FE Bandwidth(Gbps) Status      Transmit Packets         Transmit Bytes
       1:0                   120         Up               36475368            46031914416
       1:1                   120         Up               47010190            59326859780
<snip>
```

Note that GNF PE2 is configured to use only one interface (`xe-1/0/0`) on the MPC5E; this interface is mapped to one of the PFEs on the line card. Interestingly, traffic from P3 to PE2 is still load-balanced to the two PFEs. Therefore, some traffic destined to PE2 will incur a small additional latency as traffic is forwarded from the receiving PFE to the destination PFE where the `xe-1/0/0` interface resides.

## C. BSYS

On the BSYS, verify that the state of the line cards and GNFs is “Online”:

```
root@BSYS1-MX480> show chassis fpc
                    Temp  CPU Utilization (%)   CPU Utilization (%)  Memory    Utilization (%)
Slot State           (C)  Total  Interrupt      1min   5min   15min  DRAM (MB) Heap     Buffer    GNF
  0  Online           43     11          0        10     10      10    3584        6         25       1
  1  Online           70     17          0        18     18      18    3584        6         25       2
  2  Online           42      9          0         9      9       9    3584        6         25       3
  3  Empty
  4  Empty
  5  Empty

{master}
root@BSYS1-MX480> show chassis network-slices
guest-network-functions:
GNF    Description             State          Uptime
 1                             Online         2 hours, 8 minutes, 44 seconds
 2                             Online         2 hours, 8 minutes, 31 seconds
 3                             Online         2 hours, 8 minutes, 35 seconds
```

# 5. Appendix

## A. Configuration Files

### PE1

```
groups {
    re0 {
        system {
            root-authentication {
                encrypted-password "<password>";
## SECRET-DATA
            }
            services {
                ftp;
                ssh;
                telnet;
                netconf {
                    ssh;
                }
            }
            host-name PE1-MX480-re0;
            backup-router 192.0.2.254
destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
            }
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.188/24;
                        address 192.0.2.190/24 {
                            master-only;
                        }
                    }
                }
            }
        }
    }
    re1 {
        system {
            root-authentication {
                encrypted-password "<password>";
## SECRET-DATA
            }
            services {
```

### PE2

```
groups {
    re0 {
        system {
            root-authentication {
                encrypted-password "<password>"; ## SECRET-
DATA
            }
            services {
                ftp;
                ssh;
                telnet;
                netconf {
                    ssh;
                }
            }
            host-name PE2-MX480-re0;
            backup-router 192.0.2.254 destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
            }
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.191/24;
                        address 192.0.2.193/24 {
                            master-only;
                        }
                    }
                }
            }
        }
    }
    re1 {
        system {
            root-authentication {
                encrypted-password "<password>"; ## SECRET-
DATA
            }
            services {
                ftp;
```

```
                ftp;
                ssh;
                telnet;
                netconf {
                    ssh;
                }
            }
            host-name PE1-MX480-re1;
            backup-router 192.0.2.254
destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
            }
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.189/24;
                        address 192.0.2.190/24 {
                            master-only;
                        }
                    }
                }
            }
        }
    }
}
apply-groups [ re0 re1 ];
system {
    commit synchronize;
    services {
        ssh {
            max-sessions-per-connection 32;
        }
    }
    host-name PE1;
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
```

```
                ssh;
                telnet;
                netconf {
                    ssh;
                }
            }
            host-name PE2-MX480-re1;
            backup-router 192.0.2.254 destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
            }
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.192/24;
                        address 192.0.2.193/24 {
                            master-only;
                        }
                    }
                }
            }
        }
    }
}
apply-groups [ re0 re1 ];
system {
    commit synchronize;
    host-name PE2;
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
chassis {
    redundancy {
        graceful-switchover;
    }
    network-services enhanced-ip;
}
```

```
}
chassis {
    redundancy {
        graceful-switchover;
    }
    fpc 0 {
        pic 0 {
            port 0 {
                speed 100g;
            }
            port 1 {
                speed 100g;
            }
            port 2 {
                speed 100g;
            }
            port 3 {
                speed 100g;
            }
            port 4 {
                speed 100g;
            }
            port 5 {
                speed 100g;
            }
            port 6 {
                speed 100g;
            }
            port 7 {
                speed 100g;
            }
        }
    }
    network-services enhanced-ip;
}
interfaces {
    xe-0/0/0 {
        description "To Tester Port";
        unit 0 {
            family inet {
                address 198.51.100.17/28;
            }
        }
    }
    af0 {
        description "To P3";
        unit 0 {
            family inet {
                address 198.51.100.0/31;
            }
            family mpls;
        }
    }
```

```
interfaces {
    xe-1/0/0 {
        description "To Tester Port";
        unit 0 {
            family inet {
                address 198.51.100.33/28;
            }
        }
    }
    af0 {
        description "To P3";
        unit 0 {
            family inet {
                address 198.51.100.2/31;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 203.0.113.2/32;
            }
        }
    }
}
routing-options {
    nonstop-routing;
    router-id 203.0.113.2;
}
protocols {
    ldp {
        track-igp-metric;
        interface af0.0;
        interface lo0.0;
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0;
            interface xe-1/0/0.0 {
                passive;
            }
            interface af0.0 {
                interface-type p2p;
            }
        }
    }
    mpls {
        interface lo0.0;
        interface af0.0;
    }
    lldp {
        interface all;
```

```
    lo0 {
        unit 0 {
            family inet {
                address 203.0.113.1/32;
            }
        }
    }
}
routing-options {
    nonstop-routing;
    router-id 203.0.113.1;
    autonomous-system 65001;
}
protocols {
    ldp {
        track-igp-metric;
        interface af0.0;
        interface lo0.0;
    }
    ospf {
        area 0.0.0.0 {
            interface xe-0/0/0.0 {
                passive;
            }
            interface lo0.0;
            interface af0.0 {
                interface-type p2p;
            }
        }
    }
    mpls {
        interface af0.0;
        interface lo0.0;
    }
    lldp {
        interface all;
    }
}
```

## PE3

```
groups {
    re0 {
        system {
            root-authentication {
                encrypted-password "<password>";
## SECRET-DATA
            }
            services {
                ftp;
                ssh;
                telnet;
                netconf {
    }
}
```

## JDM

```
groups {
    server0 {
        system {
            host-name bsys1-jdm0;
        }
        interfaces {
            jmgmt0 {
                unit 0 {
                    family inet {
                        address 192.0.2.207/24;
                    }
                }
            }
        }
        routing-options {
            static {
                route {
                    0.0.0.0/0 next-hop 192.0.2.254;
                }
            }
        }
    }
    server1 {
        system {
            host-name bsys1-jdm1;
        }
        interfaces {
            jmgmt0 {
                unit 0 {
                    family inet {
                        address 192.0.2.208/24;
                    }
                }
            }
        }
        routing-options {
            static {
                route {
                    0.0.0.0/0 next-hop 192.0.2.254;
                }
            }
        }
    }
}
apply-groups [ server0 server1 ];
system {
    commit synchronize;
    root-authentication {
        encrypted-password "<password>"; ## SECRET-DATA
```

```
                    ssh;
                }
            }
            host-name P3-MX480-re0;
            backup-router 192.0.2.254
destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
            }
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.194/24;
                        address 192.0.2.196/24 {
                            master-only;
                        }
                    }
                }
            }
        }
    }
    re1 {
        system {
            root-authentication {
                encrypted-password "<password>";
## SECRET-DATA
            }
            services {
                ftp;
                ssh;
                telnet;
                netconf {
                    ssh;
                }
            }
            host-name P3-MX480-re1;
            backup-router 192.0.2.254
destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
            }
        }
        interfaces {
    }
    services {
        ssh;
        netconf {
            ssh;
            rfc-compliant;
        }
    }
}
virtual-network-functions {
    PE1 {
        id 1;
        resource-template 1core-16g;
        base-config /var/jdm-usr/gnf-config/pe1.cfg;
    }
    PE2 {
        id 2;
        resource-template 2core-16g;
        base-config /var/jdm-usr/gnf-config/pe2.cfg;
    }
    P3 {
        id 3;
        resource-template 1core-32g;
        base-config /var/jdm-usr/gnf-config/p3.cfg;
    }
}
```

## BSYS1

```
groups {
    re0 {
        system {
            root-authentication {
                encrypted-password "<password>"; ## SECRET-
DATA
            }
            services {
                ftp;
                ssh {
                    root-login allow;
                }
                telnet;
                netconf {
                    ssh;
                }
            }
            host-name Rochester-MX480-re0;
            backup-router 192.0.2.254 destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
```

```
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.195/24;
                        address 192.0.2.196/24 {
                            master-only;
                        }
                    }
                }
            }
        }
    }
}
apply-groups [ re0 re1 ];
system {
    commit synchronize;
    host-name P3;
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
chassis {
    redundancy {
        graceful-switchover;
    }
    network-services enhanced-ip;
}
interfaces {
    xe-2/0/1 {
        description "To PE4";
        unit 0 {
            family inet {
                address 198.51.100.4/31;
            }
            family mpls;
        }
    }
    af0 {
        description "To PE1";
        unit 0 {
            family inet {
                address 198.51.100.1/31;
            }
            family mpls;
            }
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.120/24;
                        address 192.0.2.122/24 {
                            master-only;
                        }
                    }
                }
            }
        }
    }
    re1 {
        system {
            root-authentication {
                encrypted-password "<password>"; ## SECRET-
DATA
            }
            services {
                ftp;
                ssh {
                    root-login allow;
                }
                telnet;
                netconf {
                    ssh;
                }
            }
            host-name Rochester-MX480-re1;
            backup-router 192.0.2.254 destination 0.0.0.0/0;
            time-zone America/New_York;
            name-server {
                192.0.2.106;
            }
            ntp {
                server 192.0.2.106;
            }
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 192.0.2.121/24;
                        address 192.0.2.122/24 {
                            master-only;
                        }
                    }
                }
            }
        }
```

```
        }
    }
    af1 {
        description "To PE2";
        unit 0 {
            family inet {
                address 198.51.100.3/31;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 203.0.113.3/32;
            }
        }
    }
}
routing-options {
    nonstop-routing;
    router-id 203.0.113.3;
}
protocols {
    ldp {
        track-igp-metric;
        interface xe-2/0/1.0;
        interface af0.0;
        interface af1.0;
        interface lo0.0;
    }
    ospf {
        area 0.0.0.0 {
            interface lo0.0;
            interface af0.0 {
                interface-type p2p;
            }
            interface af1.0 {
                interface-type p2p;
            }
            interface xe-2/0/1.0 {
                interface-type p2p;
            }
        }
    }
    mpls {
        interface lo0.0;
        interface af0.0;
        interface af1.0;
        interface xe-2/0/1.0;
    }
    lldp {
        interface all;
    }
}
apply-groups [ re0 re1 ];
system {
    commit synchronize;
    services {
        ssh {
            root-login allow;
            max-sessions-per-connection 32;
        }
    }
    host-name BSYS1-MX480;
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
chassis {
    redundancy {
        graceful-switchover;
    }
    fpc 1 {
        pic 1 {
            power off;
        }
        pic 2 {
            power off;
        }
    }
    network-services enhanced-ip;
    network-slices {
        guest-network-functions {
            gnf 1 {
                fpcs 0;
                af0 {
                    peer-gnf id 3 af0;
                }
            }
            gnf 2 {
                fpcs 1;
                af0 {
                    peer-gnf id 3 af1;
                }
            }
            gnf 3 {
```

```
    }
}
```

```
                fpcs 2;
                af0 {
                    peer-gnf id 1 af0;
                }
                af1 {
                    peer-gnf id 2 af0;
                }
            }
        }
    }
}
vmhost {
    resize {
        vjunos {
            compact;
        }
    }
}
routing-options {
    nonstop-routing;
}
protocols {
    lldp {
        interface all;
    }
}
```