

NEXT GENERATION ROUTING ENGINE

Technical Introduction

JTAC White Paper

Technote ID: TN303

April 2016



Table of Contents

.....	1
1. Introduction to Next Generation Routing Engine (NG-RE):.....	3
1.1 Storage details:.....	5
1.2 Disk usage design and Resiliency:	6
1.3 NG-RE boot options:	11
1.3.1 NG-RE boot order for the VMHost image:	11
1.3.2 NG-RE start-up sequence:	11
1.3.3 Junos OS VM life cycle management:.....	12
2. Prerequisite for Next Generation Routing Engine:.....	13
2.1 Hardware Requirements:.....	13
2.2 Software Image differentiation between current generation RE and NG-RE:	13
2.3 Supported Images on VMhost (x86 CPU based hardware):	14
3. Upgrade/Downgrade Procedures:	15
3.1 VMHosts/Junos OS compatibility:	15
3.2 Install Options:.....	15
3.2.1 Fresh Install:	16
3.2.2 Regular Install:	16
3.3 Multiple node upgrade:	25
3.4 VMhost upgrade, snapshot and rollback:	26
3.5 MX-VC upgrade procedure:.....	27
3.6 Procedure for creating an USB based bootable for NG-RE:.....	27
3.7 ISSU:.....	28
4. New CLI commands related to Vmhost:.....	28
4.1 Request Support Information (RSI) extension:	28
4.2 VMHOST show CLI commands:	29
4.3 VMHOST request CLI commands:	30

1. Introduction to Next Generation Routing Engine (NG-RE):

NG-RE is a new RE model sharing common infrastructure for multiple platforms such as MX and PTX.

Below are the NG-RE models:

- **RE-S-X6-64G:** 6-core Haswell CPU, Wellsburg PCH based routing engine with 64GB DRAM and 2x 64GB SSD storage for MX240/MX480/MX960
- **REMX2K-X8-64G:** 8-core Haswell CPU, Wellsburg PCH based routing engine with 64GB DRAM and 2x 64GB SSD storage for MX2020 and MX2010
- **RE-PTX-X8-64G/CB2-PTX:** 8-core Haswell CPU, Wellsburg PCH based routing engine with 64GB DRAM and 2x 64GB SSD storage for PTX5K. In addition to the Mt Rainier RE, a new control board shall be supported.

The main purpose of the NG-RE is to support Enhanced Scale and Performance, Junos OS Virtualization and Secure boot compared to current RE-1800x Routing Engine. (Note: Secure Boot supported RE will FRS with new SKUs.)

The NG-RE has a virtualized architecture where Junos OS runs as a virtual machine over a Linux based host (VMHost). When the system boots, Junos OS is automatically launched. At the end of the boot, a login prompt is seen.

The NG-RE is based on the popular hypervisor/emulator combination of kvm/qemu.

At the time of FRS, Junos OS will be the only VM operating in the administrative context, although the scope is available for expansion with other VMs in the future.

Specifications for the VM hosting Junos OS:

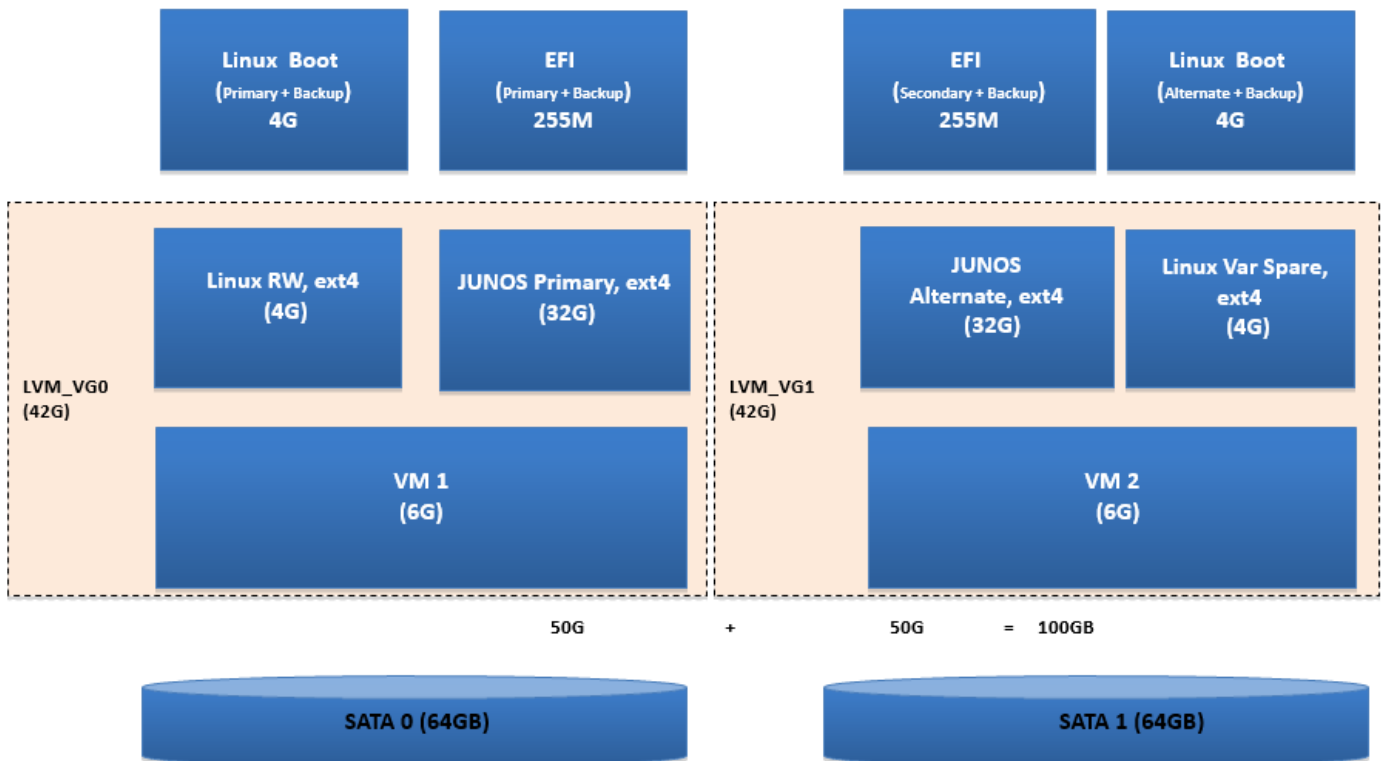
Component	Description
CPU	6 virtual CPUs on PTX, MX2K Series and 4 on MX; The virtual CPUs shall be tied to cores*.
Memory	48GB (Subject to tuning)
Disk	qemu raw disks, one for /, and swap and one for /config and /var.
Serial Ports	4 virtual serial ports; One each for console and Aux and two for cty
Management Port	One Para-virtualized 1Gb Ethernet interface(fxp0 on MX/ MX2K Series y and em0 on ptx)
Interconnection with MPCs	One 10Gb hardware-assisted Ethernet link to local CB switch. The CB to MPC link as per CB version.
Interconnection with the other RE	One 1Gb hardware-assisted Ethernet interface for MX(igb0); One 10Gb hardware-assisted Ethernet interface for MX2K Series/PTX(ixlv1 on ptx/ MX2K Series)
Watchdog	Emulated watchdog timer; Triggers reset of VM on expiry
Operating System	Occam based Junos OS

*Resource configurations to the Junos OS VM may change in subsequent releases.

1.1 Storage details:

There are two Solid State Drives (SSD) in the NG-RE.
There is no compact flash (CF).

One RE's storage partition view:



- The usable storage space per disk is about 50GB for a 64GB disk due to large over provisioning done to ensure performance and reliability.
- The partition sizes may vary based on storage device or software release in use.

1.2 Disk usage design and Resiliency:

NG-RE has two SSDs: SSD1 and SSD2. The first disk connected with channel 0 is the primary disk. The other disk is considered as backup. There are two sets of software boot images on the primary disk. We boot with one set, and when an upgrade is needed, it switches to use the other set. Similarly, on rollback, we switch to the other set.

Until an upgrade/rollback is done, the BIOS is programmed to boot from the same set of the SSD for any unplanned to planned VMhost reboot.

When a fresh installation (using USB) is done on a system, the following directories are populated with Junos OS image on both SSDs:

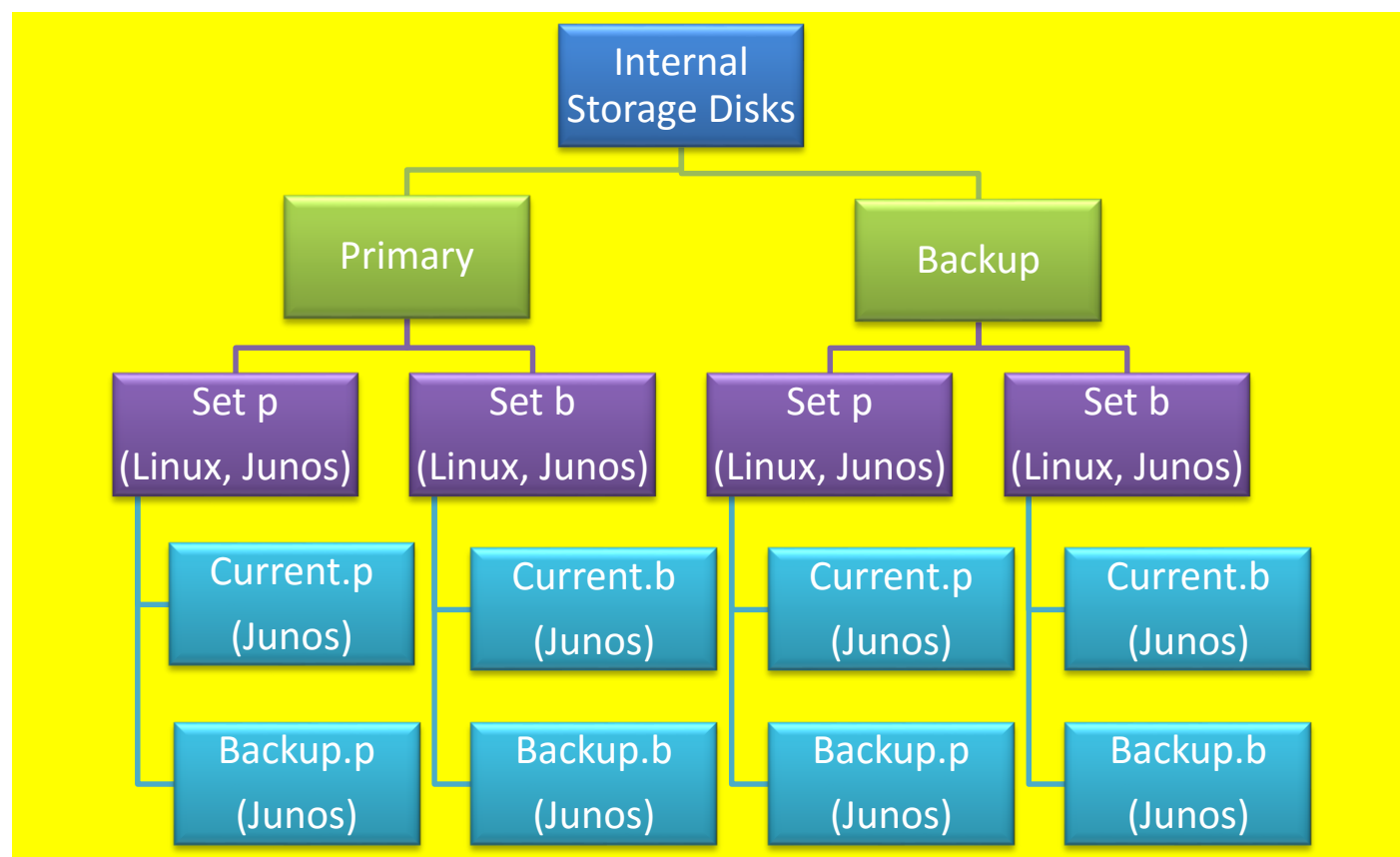
- Current.p
- Backup.p
- Backup.b

Current.b in both SSDs remain empty. Images in Backup directories are in compressed form and is used in-case there is any Corruption in Current directory image.

All the logs and SSH keys will be preserved across software upgrades/rollbacks as long as the NG-RE boots from any of the sets (P or B) as all the information is stored on a common location accessible to either of the boot images.

All the contents along with partition details are erased only when an SSD is recovered using the alternate disk or a USB.

The following diagram provides the SSD architecture:



The following output shows the NG-RE that was booted from the primary disk using set p:

```
labroot@lab2_diyal> show vmhost version
```

```
Current root details,          Device sda, Label: jrootp_P, Partition: sda3
```

```
Current boot disk: Primary
```

```
Current root set: p
```

```
UEFI      Version: NGRE_v00.53.00.01
```

```
Primary Disk, Upgrade Time: <fresh install>
```

```
Version: set p
```

```
VMHost Version: 2.932

VMHost Root: vmhost-x86_64-15.1I20160130_0011_rbu-builder

VMHost Core: vmhost-core-x86_64-15.1I20160130_0011_rbu-builder

kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt

Junos Disk: junos-install-x86-64-15.1F-20160131.0

Version: set b

VMHost Version: 2.932

VMHost Root: vmhost-x86_64-15.1I20160130_0011_rbu-builder

VMHost Core: vmhost-core-x86_64-15.1I20160130_0011_rbu-builder

kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt

Junos Disk: junos-install-x86-64-15.1F-20160131.0
```

Performing the code upgrade using the following command, NG-RE will boot from the primary disk using set b:

```
> request vmhost software add junos-vmhost-install-x86-64-15.1F4.14.tgz
```

The following output shows the NG-RE that was booted from the primary disk using set b, which was booted using the set p prior to upgrade:

```
labroot@lab2_diy1> show vmhost version

Current root details,           Device sda, Label: jrootb_P, Partition: sda4

Current boot disk: Primary

Current root set: b

UEFI      Version: NGRE_v00.53.00.01

Primary Disk, Upgrade Time: Tue Mar  1 17:44:08 UTC 2016

Version: set p

VMHost Version: 2.932
```



```
VMHost Root: vmhost-x86_64-15.1I20160130_0011_rbu-builder
VMHost Core: vmhost-core-x86_64-15.1I20160130_0011_rbu-builder
kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
Junos Disk: junos-install-x86-64-15.1F-20160131.0
Version: set b
VMHost Version: 2.900
VMHost Root: vmhost-x86_64-15.1F420151130_1049_builder
VMHost Core: vmhost-core-x86_64-15.1F420151130_1049_builder
kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
Junos Disk: junos-install-x86-64-15.1F4.14
```

The following rollback option enables the NG-RE boot using the same set used prior to the upgrade:

```
> request vmhost software rollback
```

The following snapshot shows the execution of the rollback command:

```
labroot@lab2_diyal> request vmhost software rollback
Current root details,           Device sda, Label: jrootb_P, Partition: sda4
Finding alternate root for rollback
Rollback to software on jrootp_P ...
sh /etc/install/mk-mtre-rollback.sh jrootp_P p
Mounting device in preparation for rollback...
Updating boot partition for rollback...
Rollback complete, please reboot the node for it to take effect.
Cmos Write successfull
Cmos Write successfull for Boot_retry
```

Cmos Write successfull for Boot_retry

The following capture shows the NG-RE boot up using set p after the rollback command was executed and was booted using set b:

```
labroot@lab2_diyal> show vmhost version
```

Current root details, Device sda, Label: jrootp_P, Partition: sda3

Current boot disk: Primary

Current root set: p

UEFI Version: NGRE_v00.53.00.01

Primary Disk, Upgrade Time: Tue Mar 1 17:44:08 UTC 2016

Version: set p

VMHost Version: 2.932

VMHost Root: vmhost-x86_64-15.1I20160130_0011_rbu-builder

VMHost Core: vmhost-core-x86_64-15.1I20160130_0011_rbu-builder

kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt

Junos Disk: junos-install-x86-64-15.1F-20160131.0

Version: set b

VMHost Version: 2.900

VMHost Root: vmhost-x86_64-15.1F420151130_1049_builder

VMHost Core: vmhost-core-x86_64-15.1F420151130_1049_builder

kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt

Junos Disk: junos-install-x86-64-15.1F4.14

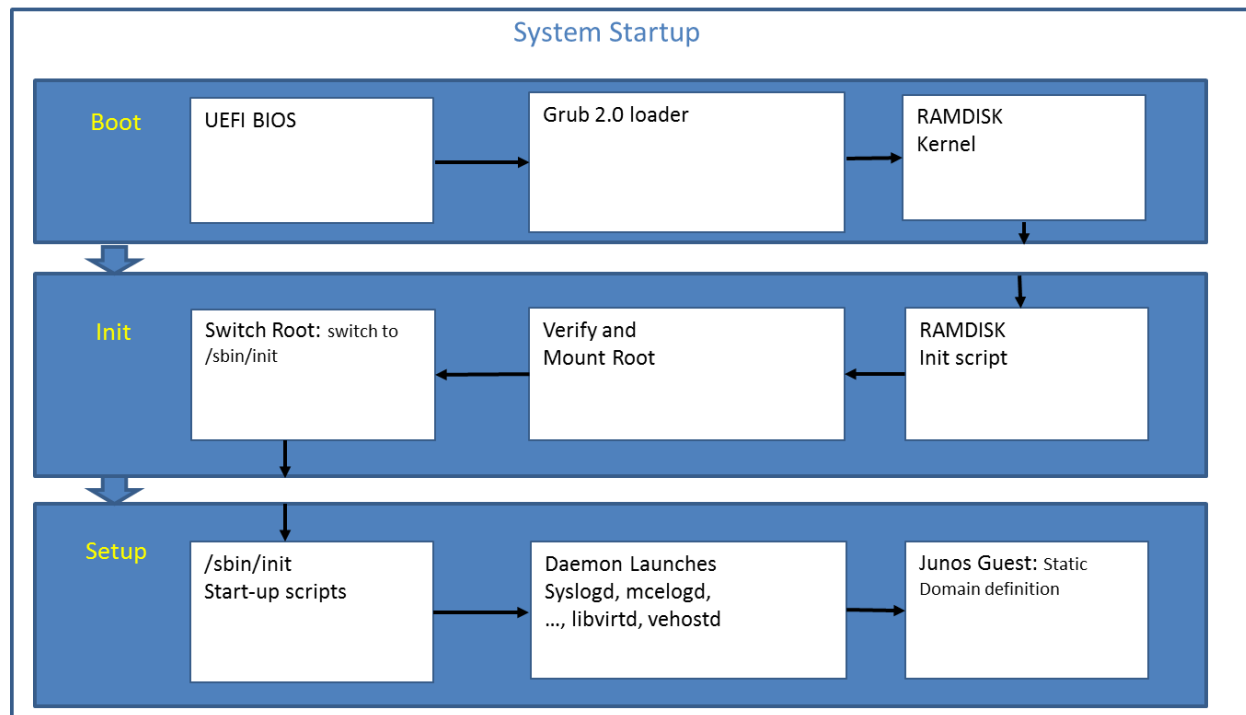
1.3 NG-RE boot options:

1.3.1 NG-RE boot order for the VMHost image:

Boot-list order: USB, SSD1, SSD2, LAN

Boot sequencing/retry in case of boot failure due to corruption or hardware failures: SSD1 and SSD2 will be tried twice during the boot sequencing.

1.3.2 NG-RE start-up sequence:

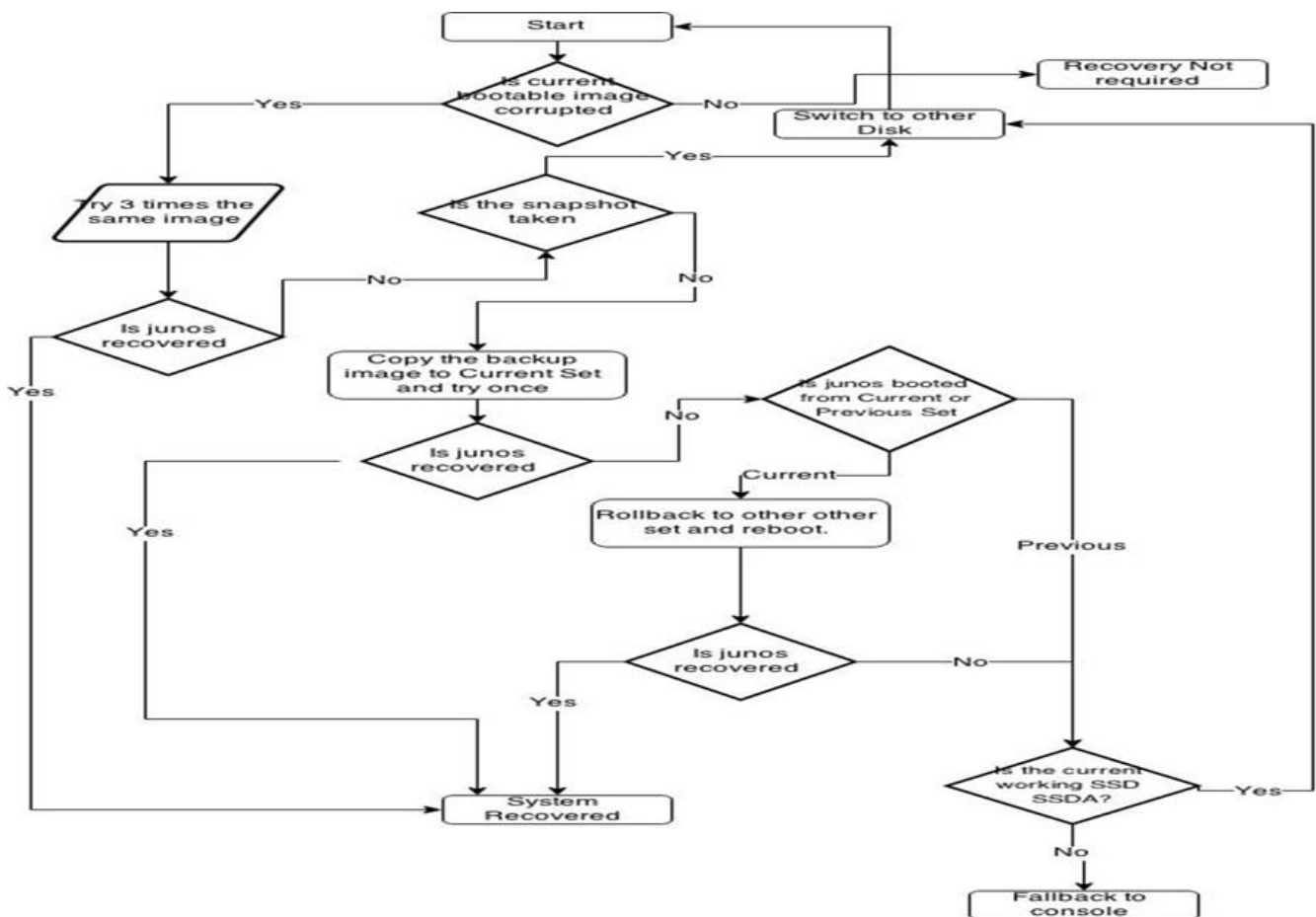


1.3.3 Junos OS VM life cycle management:

The Junos VM auto recovery is enabled by default. If the Junos OS VM launch is failed from the Current .img from the active set, then the host will try to launch the Junos OS VM from another .img available in the backup/other set/other disk directory. If the user does not want the host to recover the Junos OS VM automatically, then he can disable it using the following command:

```
#set vmhost no-auto-recovery
```

The following flow-chart explains recovery attempts made if there are any corruptions in the current boot image when the auto-recovery knob is enabled.



2. Prerequisite for Next Generation Routing Engine:

On MX960, MX480, MX240 and PTX5000, NG-RE will be supported from 15.1F3, 16.1R1 and higher.

On MX2K, NG-RE will be supported from 15.1F5, 16.1R2, 16.2R1 and higher.

2.1 Hardware Requirements:

RE-CB compatibility Matrix:

Router Model	CB Model	NG-RE	Legacy RE
MX240, MX480 and MX960	SCB	N	Y
	SCBE	N	Y
	SCBE2	Y	Y
PTX5000	OLD CB	N	Y
	NG-CB	Y	N

On MX240, 480 and 960, the NG-RE is supported only from SCBE2. The SCB is mechanically incompatible. The SCBE will be auto powered off when installed with NG-RE.

RE+CB on MX2010 and MX2020 are single entity. So, there will not be any hardware compatibility issue.

2.2 Software Image differentiation between current generation RE and NG-RE:

In the current generation of RE, there are two types of Junos OS available. They are 32-bit and 64-bit versions.

For the NG-RE, only 64-bit version is available and 32-bit version of Junos OS is not supported. There is a change in the domestic and export version naming convention for NG-RE. There is no domestic version in NG-RE, it's just a normal version. But the export version will be named as "limited".

`${package-prefix}-${media}-${platform}-${arch}-${abi}-${release}-${edition}].[tgz|img.gz]`

Where,

\${package-prefix} = “junos-vmhost-install” if it is a vmhost bundled image and “junos-install” or “junos-install-media” based on if it’s a media image or disk image for the Junos only image.

\${media} is present only when it’s a media image. It’s “usb” for images installed from USB drive and “net” for images installed over network.

\${platform} is either mx or ptx.

\${arch} is x86 – if non x86 platforms are supported, then they will have a different value for this field, such as “ppc”.

\${abi} is either “32” for 32-bit images or “64” for 64-bit images.

\${release} is the actual release name. For example, “15.1F3.9”.

\${edition} is an empty string “” for “domestic” images. For “export” (WW) images, **\${edition}** will be “limited” (See more details on this in following slides). Other images could have a different value in this field.

Note: As of now PTX images are valid for PTX5K platform only.

2.3 Supported Images on VMhost (x86 CPU based hardware):

The example of images supported on VMHost platforms are as follows:

On MX-series platforms:

junos-install-mx-x86-64-15.1F3.9.tgz ← in-image Junos upgrade for MX-series

junos-vmhost-install-mx-x86-64-15.1F3.9.tgz ← Regular VMHost upgrade image for MX-series

junos-vmhost-install-usb-mx-x86-64-15.1F3.9.img.gz ← USB installer image for MX-series

On PTX series Platforms:

junos-install-ptx-x86-64-15.1F3.9.tgz ← in-image Junos upgrade for PTX-series

junos-vmhost-install-ptx-x86-64-15.1F3.9.tgz ← Regular vmhost upgrade image for PTX-series

junos-vmhost-install-usb-ptx-x86-64-15.1F3.9.img.gz ← USB installer image for PTX-series

3. Upgrade/Downgrade Procedures:

3.1 VMHosts/Junos OS compatibility:

When the node upgrade happens via VMHost package and it is a full upgrade, by way of build components will be compatible with each other. However, a subcomponent/package upgrade subsequently e.g. in image Junos upgrade, it is important to check the installed base version and ensure the compatibility. For this, each relevant sub-package has the logic to fetch VMHost version and compare it with the supported version as specified during the build. A suitable warning is thrown and install is aborted if a version compatibility mismatch is found. Only a major version number is used for compatibility checking.

3.2 Install Options:

Install	Install Scenario	Package	Procedure
Fresh	Factory/Recovery/Clean	Recovery/Clean Install Media Package	install the host Operating System, tools and the Junos VMs using NET boot or USB disk
Regular	Field upgrade/downgrade	run time install package	The Junos VM will do the dependency check to identify which all contents need to be upgraded/downgraded to ensure compatibility

3.2.1 Fresh Install:

A fresh install scenario is one in which installation doesn't depend on the presence or absence of valid installation on the SSD. This procedure can be used for a fresh installation as well as for recovery from a corruption. A fresh installation can be done using the install media package that will install the host Operating System, tools and the Junos VMs using NET boot or USB disk.

A fresh install will prompt for install continuation before modifying internal boot media. This is to avoid unintentional formatting of the boot media. On timeout or explicit "no" by the user, it will fall to a shell.

3.2.2 Regular Install:

A regular installation uses regular install package on the currently running Junos VM to upgrade/downgrade relevant components.

The following Installations will be supported:

3.2.2.1 Junos upgrade:

Junos upgrade can be achieved via regular **junos-install-x.tgz** image upgrade as on existing REs. This will be more common and straight forward.

Junos OS upgrade needs a regular package to be specified via the "request system software add" command. If the host/hypervisor's software does not meet the minimum version as specified in the regular upgrade package, then the host upgrade would be needed as well.

3.2.2.2 Host Upgrade:

Host upgrade may be achieved via regular **junos-vmhost-install-x.tgz** image upgrade as on existing REs. Host upgrade needs regular package to be specified via the "**request vmhost**

software add” command. This is the recommended mode of upgrade as this will install the host image along with the compatible Junos OS.

A complete snapshot of the Host Image along with the Junos OS can be taken using the following command:

```
> request vmhost snapshot
```

If the device is booted from SSD1, on executing the above command snapshot will be created on the SSD2 for backup purpose.

The following command output shows the RE has booted from primary disk (disk1):

```
labroot@lab> show vmhost version
Current root details,          Device sda, Label: jrootp_P,
Partition: sda3
Current boot disk: Primary
Current root set: p
UEFI      Version: NGRE_v00.53.00.0
```

The following are details of the snapshot getting created on the secondary disk (disk2):

```
labroot@lab> request vmhost snapshot partition
warning: All existing data on the target media will be lost
Proceed ? [yes,no] (no) yes

warning: Proceeding with vmhost snapshot partition
Current root details,          Device sda, Label: jrootp_P, Partition: sda3
sh /etc/install/mk-mtre-partition.sh S
Estimated raw partition sizes:
ESP/SPARE/ROOTP/ROOTB/PV/SWAP: 954, 1908, 4770, 4770, 34346, 477
Aligned, Estimated raw partition sizes:
ESP/SPARE/ROOTP/ROOTB/PV/SWAP: 952, 1904, 4768, 4768, 34344, 472
Estimated LVM partition sizes:
PV/LVMROOT/LVMJUNOS/LVMVM/LVMSPARE: 42968, 3437, 34374, 6445, 0
Aligned, Estimated LVM partition sizes:
PV/LVMROOT/LVMJUNOS/LVMVM/LVMSPARE: 42968, 3432, 34368, 6440, 0
Aligned, adjusted raw and LVM partition sizes:
ESP/SPARE/ROOTP/ROOTB/PV/SWAP: 255, 255, 2048, 2048, 42968, 128
PV/LVMROOT/LVMJUNOS/LVMVM/LVMSPARE: 42968, 3432, 32784, 6440, 304
*****
Device partition information:
  ESP partition size :    255 MiB (/dev/sdb1) (p1) (efi_S)
  JSPARE partition size :  255 MiB (/dev/sdb2) (p2) (jspare_S)
  JROOTP partition size : 2048 MiB (/dev/sdb3) (p3) (jrootp_S)
```

```

JROOTB partition size :   2048 MiB (/dev/sdb4) (p4) (jrootb_S)
JPV partition size :   42968 MiB (/dev/sdb5) (p5) (jpv_S)
Swap partition size :    128 MiB (/dev/sdb6) (p6) (swap_S)
*****
Removing volume group jvg_S
  Logical volume "jlvrootrw" successfully removed
  Logical volume "jlvmjunos" successfully removed
  Logical volume "jlvmmvm" successfully removed
  Logical volume "jlvmspare" successfully removed
  Volume group "jvg_S" successfully removed
Existing partition table on /dev/sdb ...
2+0 records in
2+0 records out
1024 bytes (1.0 kB) copiedModel: ATA StorFly VSF202CC (scsi)
Disk /dev/sdb: 47703MiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags

Deleting partition table on /dev/sdb ...
Initiating Secure Erase on /dev/sdb ...
Setting temporary password on /dev/sdb ...
, 0.000326747 s, 3.1 MB/s
security_password="PaSsWoRd"

/dev/sdb:
  Issuing SECURITY_SET_PASS command, password="PaSsWoRd", user=user, mode=high
security_password="PaSsWoRd"

/dev/sdb:
  Issuing SECURITY_ERASE command, password="PaSsWoRd", user=user

real    0m4.830s
user    0m0.000s
sys     0m0.001s
Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Information: You may need to update /etc/fstab.

Creating new partition table (MSDOS) on /dev/sdb ...
Creating boot partition on /dev/sdb1
Enabling boot flag on /dev/sdb1
Creating JSPARE partition on /dev/sdb2

```



```
JLVMSPAREFS partition size :    304 MiB (/dev/jvg_S/jlvmspare) (jlvmspare)
(jlvmspare)
*****
Creating Physical Volume on /dev/sdb5 ...
   Writing physical volume data to disk "/dev/sdb5"
   Physical volume "/dev/sdb5" successfully created
Creating Volume Group jvg_S ...
   Volume group "jvg_S" successfully created
Creating JLVMROOTFS jlvrootrw ...
   Logical volume "jlvrootrw" created
Creating JLVMJUNOSFS jlvmjunos ...
   Logical volume "jlvmjunos" created
Creating JLVMMVMSFS jlvmmvm ...
   Logical volume "jlvmmvm" created
Creating JLVMSPAREFS jlvmspare ...
mke2fs 1.42.8 (20-Jun-2013)
   Logical volume "jlvmspare" created
Formatting as ext4...
Discarding device blocks:
4096/878592\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08
\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08done
Filesystem label=jlvrootrw
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
219888 inodes, 878592 blocks
43929 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=901775360
27 block groups
32768 blocks per group, 32768 fragments per group
8144 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200

Allocating group tables: 0/27\x08\x08\x08\x08\x08      \x08\x08\x08\x08\x08done
Writing inode tables: 0/27\x08\x08\x08\x08\x08      \x08\x08\x08\x08\x08done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information:
0/27\x08\x08\x08\x08\x08      \x08\x08\x08\x08\x08done

mke2fs 1.42.8 (20-Jun-2013)
Formatting as ext4...
Discarding device blocks:
4096/8392704\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08
\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08\x08done
Filesystem label=jlvmjunos
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2101232 inodes, 8392704 blocks
```



```

Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
77824 inodes, 311296 blocks
15564 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67633152
38 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: 0/38\x08\x08\x08\x08\x08      \x08\x08\x08\x08\x08done
Writing inode tables: 0/38\x08\x08\x08\x08\x08      \x08\x08\x08\x08\x08done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information:
0/38\x08\x08\x08\x08\x08      \x08\x08\x08\x08\x08done

Block device tree dump ...
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0      0 46.6G  0 disk
|-sda1               8:1      0 254M  0 part
|-sda2               8:2      0 255M  0 part
|-sda3               8:3      0   2G  0 part /.old
|-sda4               8:4      0   2G  0 part
|-sda5               8:5      0 42G   0 part
| |-jvg_P-jlvmrootrw 252:4      0 3.4G  0 lvm  /.old/tdev/.union/.s
| |-jvg_P-jlvmjunos  252:5      0 32G   0 lvm  /junos
| |-jvg_P-jlvmvm      252:6      0 6.3G  0 lvm  /vm
| |-jvg_P-jlvm spare  252:7      0 304M  0 lvm  /spare
`-sda6               8:6      0 129M  0 part
sdb                  8:16     0 46.6G  0 disk
|-sdb1               8:17     0 254M  0 part
|-sdb2               8:18     0 255M  0 part
|-sdb3               8:19     0   2G  0 part
|-sdb4               8:20     0   2G  0 part
|-sdb5               8:21     0 42G   0 part
| |-jvg_S-jlvmrootrw 252:0      0 3.4G  0 lvm
| |-jvg_S-jlvmjunos  252:1      0 32G   0 lvm
| |-jvg_S-jlvmvm      252:2      0 6.3G  0 lvm
| |-jvg_S-jlvm spare  252:3      0 304M  0 lvm
`-sdb6               8:22     0 129M  0 part
Snapshot admin context from current boot disk to target disk ...
Proceeding with snapshot on secondary disk
Mounting device in preparation for snapshot...
Cleaning up target disk for snapshot ...
Creating snapshot on target disk from current boot disk ...
Snapshot created on secondary disk.
Software snapshot done

```

The following output shows the details of a snapshot created on the secondary disk:

```
labroot@lab> show vmhost snapshot
UEFI      Version: NGRE_v00.53.00.01
```

Secondary Disk, Snapshot Time: Thu Feb 25 05:23:03 UTC 2016

```
Version: set p
VMHost Version: 2.932
VMHost Root: vmhost-x86_64-15.1I20160130_0011_rbu-builder
VMHost Core: vmhost-core-x86_64-15.1I20160130_0011_rbu-builder
kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
Junos Disk: junos-install-x86-64-15.1F-20160131.0
```

```
Version: set b
VMHost Version: 2.912
VMHost Root: vmhost-x86_64-15.1I20151223_0011_rbu-builder
VMHost Core: vmhost-core-x86_64-15.1I20151223_0011_rbu-builder
kernel: 3.10.79-ovp-rt74-WR6.0.0.20_preempt-rt
Junos Disk: junos-install-x86-64-15.1-20151224_mfg_15q1_gld.1
VMHost Diag: vmhost-diag-x86_64-15.1I20151223_0011_rbu-builder
```

The following command can be used to boot the RE from a snapshot created on the secondary disk (disk2).

```
labroot@lab> request vmhost reboot disk2 re0
```

The following command confirms that VMHost was booted from the secondary disk:

```
labroot@lab> show vmhost version
Current root details,      Device sdb, Label: jrootp_S, Partition: sdb3
Current boot disk: Secondary
Current root set: p
UEFI      Version: NGRE_v00.53.00.01
```

The Following alarm also indicates the RE booted from the secondary disk (disk2):

```
labroot@lab> show chassis alarms
1 alarms currently active
Alarm time      Class  Description
2016-02-25 23:18:29 PST  Minor  VMHost 0 Boot from alternate disk
```

The router will be booted from the alternate set in the following conditions:

- When the attempt to launch Junos VM using the active set image failed and rollback is attempted to boot from the alternate set.
- When the active set fails and root rollover is done during boot.

```
labroot@lab> show vmhost version
Current root details,          Device sda, Label: jrootb_P, Partition: sda4
Current boot disk: Primary
Current root set: b
UEFI      Version: NGRE_v0.43
```

The following alarm also indicates the RE booted from the secondary set:

```
labroot@lab> show chassis alarms
1 alarms currently active
Alarm time          Class  Description
2016-02-26 02:09:19 PST  Minor  VMHost 0 Boot from alternate set
```

Note: When the device is booted from SSD2, the following must be used to take a complete VMhost snapshot along with partition details to SSD1. This option would be used to recover SSD1 having a corrupted image.

```
> request vmhost snapshot recovery [partition]
```

Note: When a regular install is attempted while the RE is booted from SSD2 (disk2), the image upgrade is done on the SSD1, not on the other set in the same disk.

It's always recommended to take the complete snapshot of the Host along with the Junos OS before the upgrade.

3.3 Multiple node upgrade:

The regular software upgrade command supports option `re0 | re 1` to enable the upgrade software on both REs. This option will be leveraged to carry out required software upgrade on the other RE. The procedure would be the same as local RE upgrade, except the package will be pushed from another RE.

3.4 VMhost upgrade, snapshot and rollback:

Booted From	Upgrade (request vmhost software add)	Rollback (request vmhost software rollback)	Snapshot (request vmhost snapshot [partition])	Snapshot (request vmhost snapshot recovery [partition])
Primary Disk Set p	Primary Disk Sep b	Primary Disk Set b	Secondary Disk (set p and b)	N/A
Primary disk set b	Primary Disk Sep p	Primary Disk Set p	Secondary Disk (set p and b)	N/A
Backup Disk (set p or b)	Primary Disk Sep p and b (format and install)	N/A	N/A	Primary Disk (set p and b)

3.5 MX-VC upgrade procedure:

Not supported at FRS.

3.6 Procedure for creating an USB based bootable for NG-RE:

NG-RE USB installation will install both Linux and Junos OS. This installation will erase all files in Linux and Junos. Create back up files if needed before proceeding with installation.

NOTE: For NG-RE USB installation, you need a USB with at least 8GB.

1. Copy `junos-vmhost-install-usb-x86-64-15.1-xxxx_rainier_paradise.0.img.gz` to RE.
2. Unzip the copied file in RE.
3. Write the above image to USB via the `dd` command
(`dd if=/path/to/downloaded.img of=/dev/devicenode bs=4M`)
from BSD shell prompt as root user [Device node can be identified by giving `ls /dev/da*`. Mostly it is `/dev/da0`].
4. Insert USB on NG-RE and reboot via CLE `"request vmhost reboot"` or power cycle RE.
5. Press Y to install image from USB to SSDs. After installation is completed, Press Y again to reboot the router.
6. For any reason, if the user key is in a different character, then it will fall to shell. The user is requested to do an exit from the shell in order for the installation to proceed. When prompted to install, press Y to continue. Once completed, press Y again to reboot the router.
7. After successful Junos OS boot, remove the USB from RE. Don't remove USB in-between, as it might corrupt file systems of USB.

3.7 ISSU:

It's not supported at FRS.

4. New CLI commands related to Vmhost:

4.1 Request Support Information (RSI) extension:

Command	Description
show vmhost status	Information whether Host to Guest communication is fine
show vmhost bridge	Interfaces used for communication between Host and Guest
show vmhost netstat	Netstat of Host
show vmhost information	Information like Host uptime
show vmhost processes	Host process information
show vmhost support-info	This command internally execute below linux commands which help in debug <ul style="list-style-type: none">• Ethtool -I <interface>• Df -h• Uptime• Dmesg• Uname -a• ps -aefT• top -n1 -b• cat /proc/cpuinfo• cat /proc/meminfo• mpstat -P ALL• virsh dumpxml vjunos• virsh vcpuinfo vjunos• cat /var/log/daemon.log• cat /var/log/syslog• cat /var/log/auth.log• cat /var/log/audit/audit.log• ls /var/crash

4.2 VMHOST show CLI commands:

CLI	Purpose	Junos equivalent commands	Comment
Show vmhost bridge	Provide information of bridge Table	NA	
Show vmhost crash	List crashes happened in host	Show system core-dumps	
Show vmhost hardware	Show RE hardware	Show chassis hardware	
Show vmhost information	Provide information about host IP, version, model, etc	NA	
Show vmhost logs	List logs from host		
Show vmhost netstat	List host netstat		
Show vmhost process	List host process	Show system process	
Show vmhost resource-usage	Show RE vmhost resource usage	NA	
Show vmhost snapshot	Show vmhost snapshot information		
Show vmhost status	Inform host status (Offline/Online)		
Show vmhost uptime	Show RE vmhost uptime and reboot reason	Show system uptime	
Show vmhost version	Show vmhost version information	Show version	

4.3 VMHOST request CLI commands:

CLI	Purpose	Junos equivalent commands	Comment
Request vmhost cleanup	Intention is to cleanup logs, /var/tmp and /var/crash contained from host	Request system storage cleanup	
Request vmhost file-copy	Intention is to copy crash or logs from host to junos	NA	
Request vmhost halt	Intention is to halt the RE, which is useful in upgrades.	Request system halt	Junos command depreciated, vmhost command currently behave similar to power-off
Request vmhost power-off	Power off the junos as well as host	Request system power-off	Junos command depreciated
Request vmhost power-on	Power on the junos as well as host	Request system power-on	Junos command depreciated
Request vmhost reboot	Reboot RE vmhost	Request system reboot	This command reboots the Junos OS VM alone.
Request vmhost snapshot	Create a vmhost recovery snapshot	Request system snapshot	

Request vmhost software	Upgrade junos as well as host package	Request system software	
Request vmhost hard-disk-test	Run smartd self-tests on hard disks		
Request vmhost zeroize	Erase all data, including configuration and log files	Request system zeroize	