JUNIPER NETWORKS | Engineering Simplicity

vSRX Deployment Guide for Private and Public Cloud Platforms

Published
2021-10-24

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

vSRX Deployment Guide for Private and Public Cloud Platforms

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

vSRX is the virtualized form of the Juniper Networks next-generation firewall. It is positioned for use in a virtualized or cloud environment where it can protect and secure east-west and north-south traffic. This guide provides you details on deployment of vSRX on various private and public cloud platforms.

1

PART

vSRX Deployment for KVM

CHAPTER 1

Overview

IN THIS CHAPTER

Understand vSRX with KVM

IN THIS SECTION

This section presents an overview of vSRX on KVM.

vSRX on KVM

The Linux kernel uses the kernel-based virtual machine (*KVM*) as a virtualization infrastructure. KVM is open source software that you can use to create multiple virtual machines (VMs) and to install security and networking appliances.

The basic components of KVM include:

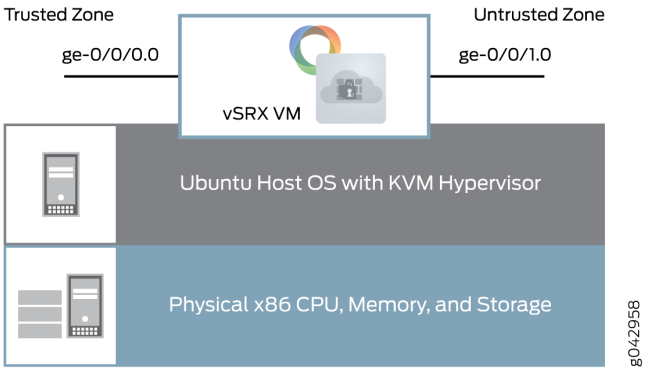
- A loadable kernel module included in the Linux kernel that provides the basic virtualization infrastructure
- A processor-specific module

When loaded into the Linux kernel, the KVM software acts as a *hypervisor*. KVM supports *multitenancy* and allows you to run multiple vSRX VMs on the *host* OS. KVM manages and shares the system resources between the host OS and the multiple vSRX VMs.

NOTE: vSRX requires you to enable hardware-based virtualization on a host OS that contains an Intel Virtualization Technology (VT) capable processor.

Figure 1 on page 3 illustrates the basic structure of a vSRX VM on an Ubuntu server.

Figure 1: vSRX VM on Ubuntu



vSRX Scale Up Performance

Table 1 on page 3 shows the vSRX scale up performance when deployed on KVM, based on the number of vCPUs and vRAM applied to a vSRX VM along with the Junos OS release in which a particular vSRX software specification was introduced.

Table 1: vSRX Scale Up Performance

vCPUs	vRAM	NICs	Release Introduced
2 vCPUs	4 GB	• Virtio • SR-IOV (Intel 82599, X520/540)	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1

Table 1: vSRX Scale Up Performance *(Continued)*

vCPUs	vRAM	NICs	Release Introduced
5 vCPUs	8 GB	• Virtio • SR-IOV (Intel 82599, X520/540)	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1
5 vCPUs	8 GB	• SR-IOV (Intel X710/ XL710)	Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1
1 vCPU	4 GB	SR-IOV on the Mellanox ConnectX-4 and ConnectX-5 family adapters.	Junos OS Release 21.2R1
4 vCPUs	8 GB	SR-IOV on the Mellanox ConnectX-4 and ConnectX-5 family adapters.	Junos OS Release 21.2R1
8 vCPUs	16GB	SR-IOV on the Mellanox ConnectX-4 and ConnectX-5 family adapters.	Junos OS Release 21.2R1
16 vCPUs	32 GB	SR-IOV on the Mellanox ConnectX-4 and ConnectX-5 family adapters.	Junos OS Release 21.2R1

You can scale the performance and capacity of a vSRX instance by increasing the number of vCPUs and the amount of vRAM allocated to the vSRX. The multi-core vSRX automatically selects the appropriate vCPUs and vRAM values at boot time, as well as the number of Receive Side Scaling (RSS) queues in the NIC. If the vCPU and vRAM settings allocated to a vSRX VM do not match what is currently available, the vSRX scales down to the closest supported value for the instance. For example, if a vSRX VM has 3 vCPUs and 8 GB of vRAM, vSRX boots to the smaller vCPU size, which requires a minimum of 2 vCPUs. You can scale up a vSRX instance to a higher number of vCPUs and amount of vRAM, but you cannot scale down an existing vSRX instance to a smaller setting.

NOTE: The number of RSS queues typically matches with the number of data plane vCPUs of a vSRX instance. For example, a vSRX with 4 data plane vCPUs should have 4 RSS queues.

vSRX Session Capacity Increase

vSRX solution is optimized to increase the session numbers by increasing the memory.

With the ability to increase the session numbers by increasing the memory, you can enable vSRX to:

- Provide highly scalable, flexible and high-performance security at strategic locations in the mobile network.
- Deliver the performance that service providers require to scale and protect their networks.

Run the `show security flow session summary | grep maximum` command to view the maximum number of sessions.

Starting in Junos OS Release 18.4R1, the number of flow sessions supported on a vSRX instance is increased based on the vRAM size used.

Starting in Junos OS Release 19.2R1, the number of flow sessions supported on a vSRX 3.0 instance is increased based on the vRAM size used.

NOTE: Maximum of 28M sessions are supported on vSRX 3.0. You can deploy vSRX 3.0 with more than 64G memory, but the maximum flow sessions can still be only 28M.

Table 2 on page 5 lists the flow session capacity.

Table 2: vSRX and vSRX 3.0 Flow Session Capacity Details

vCPUs	Memory	Flow Session Capacity
2	4 GB	0.5 M
2	6 GB	1 M
2/5	8 GB	2 M

Table 2: vSRX and vSRX 3.0 Flow Session Capacity Details *(Continued)*

vCPUs	Memory	Flow Session Capacity
2/5	10 GB	2 M
2/5	12 GB	2.5 M
2/5	14 GB	3 M
2/5/9	16 GB	4 M
2/5/9	20 GB	6 M
2/5/9	24 GB	8 M
2/5/9	28 GB	10 M
2/5/9/17	32 GB	12 M
2/5/9/17	40 GB	16 M
2/5/9/17	48 GB	20 M
2/5/9/17	56 GB	24 M
2/5/9/17	64 GB	28 M

Release History Table

Release	Description
19.2R1	Starting in Junos OS Release 19.2R1, the number of flow sessions supported on a vSRX 3.0 instance is increased based on the vRAM size used.
18.4R1	Starting in Junos OS Release 18.4R1, the number of flow sessions supported on a vSRX instance is increased based on the vRAM size used.

RELATED DOCUMENTATION

Requirements for vSRX on KVM | 7

Upgrade a Multi-core vSRX | 75

Install vSRX with KVM | 20

Requirements for vSRX on KVM

IN THIS SECTION

This section presents an overview of requirements for deploying a vSRX instance on KVM;

Software Specifications

Table 3 on page 8 lists the system software requirement specifications when deploying vSRX in a KVM environment. The table outlines the Junos OS release in which a particular software specification for deploying vSRX on KVM was introduced. You will need to download a specific Junos OS release to take advantage of certain features.

CAUTION: A Page Modification Logging (PML) issue related to the KVM host kernel might prevent the vSRX from successfully booting. If you experience this behavior with the vSRX, we recommend that you disable the PML at the host kernel level. See *Prepare Your Server for vSRX Installation* for details about disabling the PML as part of enabling nested virtualization.

Table 3: Specifications for vSRX

Component	Specification	Release Introduced
Linux KVM Hypervisor support	Ubuntu 14.04.5, 16.04, 16.10, and 18.04	Junos OS Release 18.4R1
	Red Hat Enterprise Linux (RHEL) 7.3	
	CentOS 7.2	
	CentOS 7.6 and 7.7	Junos OS Release 19.2R1
	Red Hat Enterprise Linux (RHEL) 7.6 and 7.7	Junos OS Release 19.2R1
	Red Hat Enterprise Linux (RHEL) 8.2	Junos OS Release 20.4R1
Memory	4 GB	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
	8 GB	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1
	16 GB	Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1
	32 GB	Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1
Disk space	16 GB IDE drive	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
vCPUs	2 vCPUs	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
	5 vCPUs	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1

Table 3: Specifications for vSRX *(Continued)*

Component	Specification	Release Introduced
	9 vCPUs	Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1
	17 vCPUs	Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1
vNICs	2-8 vNICs. • Virtio • SR-IOV (Intel 82599, X520/X540) For SR-IOV limitations, see the *Known Behavior* section of the *vSRX Release Notes*.	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
	• SR-IOV (X710, XL710, and XXV710)	Junos OS Release 15.1X49-D90
	• SR-IOV (Mellanox ConnectX-3/ ConnectX-3 Pro and Mellanox ConnectX-4 EN/ConnectX-4 Lx EN)	Junos OS Release 18.1R1
	Starting in Junos OS Release 19.4R1, DPDK version 18.11 is supported on vSRX. With this feature the Mellanox Connect Network Interface Card (NIC) on vSRX now supports OSPF Multicast and VLANs.	Junos OS Release 19.4R1

NOTE: A vSRX on KVM deployment requires you to enable hardware-based virtualization on a host OS that contains an Intel Virtualization Technology (VT) capable processor. You can verify CPU compatibility here: http://www.linux-kvm.org/page/Processor_support

Table 4 on page 10 lists the specifications on the vSRX VM.

Table 4: Specifications for vSRX 3.0

Component	Specification	Release Introduced
Linux KVM Hypervisor support	CentOS 7.2	Junos OS Release 18.4R1
	CentOS 7.6 and 7.7	Junos OS Release 19.2R1
	Red Hat Enterprise Linux (RHEL) 7.6 and 7.7	Junos OS Release 19.2R1
	Red Hat Enterprise Linux (RHEL) 8.2	Junos OS Release 20.4R1
Memory	4 GB	Junos OS Release 18.2R1
	8 GB	Junos OS Release 18.4R1
	16 GB	Junos OS Release 19.1R1
	32 GB	Junos OS Release 19.1R1
vDisk	20G	Junos OS Release 18.2R1
vCPUs	2	Junos OS Release 18.2R1
	5	Junos OS Release 18.4R1
	9	Junos OS Release 19.1R1
	17	Junos OS Release 19.1R1
vNICs	2-8	Junos OS Release 18.2R1
	2-8 vSRX supports VIRTIO on KVM hypervisor.	Junos OS Release 18.4R1

Table 4: Specifications for vSRX 3.0 *(Continued)*

Component	Specification	Release Introduced
	vSRX3.0 supports LiquidIO DPDK driver with KVM hypervisor. If you use the LiquidIO II smart NICs, then you can use vSRX3.0 by the VF of SR-IOV.	Junos OS Release 20.4R1
	SR-IOV 10-Gigabit High Availability on vSRX3.0. See [Configuring SR-IOV 10-Gigabit High Availability on vSRX 3.0]	Junos OS Release 20.4R1
	Mellanox ConnectX-4 and ConnectX-5 family adapters. For a summary of vSRX sizes (number of vCPU and amount of vRAM) that support the Mellanox ConnectX-4 and ConnectX-5 Family Adapters, see vSRX Scale Up Performance	Junos OS Release 21.2R1
DPDK	17.05	Junos OS Release 18.2R1
	17.5.02	Junos OS Release 19.1R1
	18.11	Junos OS Release 19.4R1
	Data Plane Development Kit (DPDK) version 20.11 on vSRX 3.0. The new version supports ICE Poll Mode Driver (PMD), which enables you to integrate with the physical Intel E810 series 100G NIC. Junos FreeBSD 12.Xis vSRX 3.0 VM's guest OS. The Routing Engine and Packet Forwarding Engine run on Junos FreeBSD OS as one VM, and the Packet Forwarding Engine utilizes DPDK technologies such as DPDK ICE PMD and single-root I/O virtualization (SR-IOV).	Junos OS Release 21.2R1

Starting in Junos OS Release 19.1R1, the vSRX instance supports guest OS using 9 or 17 vCPUs with single-root I/O virtualization over Intel X710/XL710 on Linux KVM hypervisor for improved scalability and performance.

KVM Kernel Recommendations for vSRX

Table 5 on page 12 lists the recommended Linux kernel version for your Linux host OS when deploying vSRX on KVM. The table outlines the Junos OS release in which support for a particular Linux kernel version was introduced.

Table 5: Kernel Recommendations for KVM

Linux Distribution	Linux Kernel Version	Supported Junos OS Release
CentOS	3.10.0.229 Upgrade the Linux kernel to capture the recommended version.	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1 or later release
Ubuntu	3.16	
	4.4	
RHEL	3.10	

Additional Linux Packages for vSRX on KVM

Table 6 on page 12 lists the additional packages you need on your Linux host OS to run vSRX on KVM. See your host OS documentation for how to install these packages if they are not present on your server.

Table 6: Additional Linux Packages for KVM

Package	Version	Download Link
libvirt	0.10.0	libvirt download
virt-manager (Recommended)	0.10.0	virt-manager download

Hardware Specifications

Table 7 on page 13 lists the hardware specifications for the host machine that runs the vSRX VM.

Table 7: Hardware Specifications for the Host Machine

Component	Specification
Host processor type	Intel x86_64 multi-core CPU **NOTE**: DPDK requires Intel Virtualization VT-x/VT-d support in the CPU. See *About Intel Virtualization Technology*.
Physical NIC support for vSRX and vSRX 3.0	• Virtio • SR-IOV (Intel X710/XL710, X520/540, 82599) • SR-IOV (Mellanox ConnectX-3/ConnectX-3 Pro and Mellanox ConnectX-4 EN/ ConnectX-4 Lx EN) **NOTE**: If using SR-IOV with either the Mellanox ConnectX-3 or ConnectX-4 Family Adapters, on the Linux host, if necessary, install the latest MLNX_OFED Linux driver. See Mellanox OpenFabrics Enterprise Distribution for Linux (MLNX_OFED). **NOTE**: You must enable the Intel VT-d extensions to provide hardware support for directly assigning physical devices per guest. See *Configure SR-IOV and PCI on KVM*.
Physical NIC support for vSRX 3.0	Support SR-IOV on Intel X710/XL710/XXV710 and Intel E810

Best Practices for Improving vSRX Performance

Review the following practices to improve vSRX performance.

NUMA Nodes

The x86 server architecture consists of multiple sockets and multiple cores within a socket. Each socket has memory that is used to store packets during I/O transfers from the NIC to the host. To efficiently read packets from memory, guest applications and associated peripherals (such as the NIC) should reside within a single socket. A penalty is associated with spanning CPU sockets for memory accesses, which might result in nondeterministic performance. For vSRX, we recommend that all vCPUs for the vSRX VM are in the same physical non-uniform memory access (NUMA) node for optimal performance.

CAUTION: The Packet Forwarding Engine (PFE) on the vSRX will become unresponsive if the NUMA nodes topology is configured in the hypervisor to spread the instance's vCPUs across multiple host NUMA nodes. vSRX requires that you ensure that all vCPUs reside on the same NUMA node.

We recommend that you bind the vSRX instance with a specific NUMA node by setting NUMA node affinity. NUMA node affinity constrains the vSRX VM resource scheduling to only the specified NUMA node.

Mapping Virtual Interfaces to a vSRX VM

To determine which virtual interfaces on your Linux host OS map to a vSRX VM:

1. Use the `virsh list` command on your Linux host OS to list the running VMs.

```
hostOS# virsh list
```

```
Id    Name                           State
----------------------------------------------------
9     centos1                        running
15    centos2                        running
16    centos3                        running
48    vsrx                           running
50    1117-2                         running
51    1117-3                         running
```

2. Use the `virsh domiflist` *`vsrx-name`* command to list the virtual interfaces on that vSRX VM.

```
hostOS# virsh domiflist vsrx
```

```
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet1      bridge     brem2      virtio      52:54:00:8f:75:a5
vnet2      bridge     br1        virtio      52:54:00:12:37:62
vnet3      bridge     brconnect  virtio      52:54:00:b2:cd:f4
```

NOTE: The first virtual interface maps to the fxp0 interface in Junos OS.

Interface Mapping for vSRX on KVM

Each network adapter defined for a vSRX is mapped to a specific interface, depending on whether the vSRX instance is a standalone VM or one of a cluster pair for high availability. The interface names and mappings in vSRX are shown in Table 8 on page 15 and Table 9 on page 16.

Note the following:

- In standalone mode:
  - fxp0 is the out-of-band management interface.
  - ge-0/0/0 is the first traffic (revenue) interface.
- In cluster mode:
  - fxp0 is the out-of-band management interface.
  - em0 is the cluster control link for both nodes.
  - Any of the traffic interfaces can be specified as the fabric links, such as ge-0/0/0 for fab0 on node 0 and ge-7/0/0 for fab1 on node 1.

Table 8 on page 15 shows the interface names and mappings for a standalone vSRX VM.

Table 8: Interface Names for a Standalone vSRX VM

Network Adapter	Interface Name in Junos OS for vSRX
1	fxp0
2	ge-0/0/0
3	ge-0/0/1
4	ge-0/0/2
5	ge-0/0/3

Table 8: Interface Names for a Standalone vSRX VM *(Continued)*

Network Adapter	Interface Name in Junos OS for vSRX
6	ge-0/0/4
7	ge-0/0/5
8	ge-0/0/6

Table 9 on page 16 shows the interface names and mappings for a pair of vSRX VMs in a cluster (node 0 and node 1).

Table 9: Interface Names for a vSRX Cluster Pair

Network Adapter	Interface Name in Junos OS for vSRX
1	fxp0 (node 0 and 1)
2	em0 (node 0 and 1)
3	ge-0/0/0 (node 0) ge-7/0/0 (node 1)
4	ge-0/0/1 (node 0) ge-7/0/1 (node 1)
5	ge-0/0/2 (node 0) ge-7/0/2 (node 1)
6	ge-0/0/3 (node 0) ge-7/0/3 (node 1)
7	ge-0/0/4 (node 0) ge-7/0/4 (node 1)

Table 9: Interface Names for a vSRX Cluster Pair *(Continued)*

Network Adapter	Interface Name in Junos OS for vSRX
8	ge-0/0/5 (node 0) ge-7/0/5 (node 1)

vSRX Default Settings on KVM

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.

Table 10 on page 17 lists the factory-default settings for security policies on the vSRX.

Table 10: Factory Default Settings for Security Policies

Source Zone	Destination Zone	Policy Action
trust	untrust	permit
trust	trust	permit
untrust	trust	deny

RELATED DOCUMENTATION

About Intel Virtualization Technology

DPDK Release Notes

CHAPTER 2

Install vSRX in KVM

IN THIS CHAPTER

Prepare Your Server for vSRX Installation

IN THIS SECTION

Enable Nested Virtualization

We recommend that you enable nested *virtualization* on your host OS or OpenStack compute node. Nested virtualization is enabled by default on Ubuntu but is disabled by default on *CentOS*.

Use the following command to determine if nested virtualization is enabled on your host OS. The result should be Y.

hostOS# **cat /sys/module/kvm_intel/parameters/nested**

hostOS# Y

NOTE: APIC virtualization (APICv) does not work well with nested VMs such as those used with KVM. On Intel CPUs that support APICv (typically v2 models, for example E5 v2 and E7 v2), you must disable APICv on the host server before deploying vSRX.

To enable nested virtualization on the host OS:

1. Depending on your host operating system, perform the following:
   - On CentOS, open the **/etc/modprobe.d/dist.conf** file in your default editor.

     ```
     hostOS# vi /etc/modprobe.d/dist.conf
     ```

   - On Ubuntu, open the **/etc/modprobe.d/qemu-system-x86.conf** file in your default editor.

     ```
     hostOS# vi /etc/modprobe.d/qemu-system-x86.conf
     ```

2. Add the following line to the file:

   ```
   hostOS# options kvm-intel nested=y enable_apicv=n
   ```

   NOTE: A Page Modification Logging (PML) issue related to the KVM host kernel might prevent the vSRX from successfully booting. We recommend that you add the following line to the file *instead* of the line listed above in Step 2:

   ```
   hostOS# options kvm-intel nested=y enable_apicv=n pml=n
   ```

3. Save the file and reboot the host OS.
4. (Optional) After the reboot, verify that nested virtualization is enabled.

   ```
   hostOS# cat /sys/module/kvm_intel/parameters/nested
   ```

   ```
   hostOS# Y
   ```

5. On Intel CPUs that support APICv (for example, E5 v2 and E7 v2), disable APICv on the host OS.

```
root@host# sudo rmmod kvm-intel
root@host# sudo sh -c “echo ’options kvm-intel enable_apicv=n’ >> /etc/modprobe.d/dist.conf”
root@host# sudo modprobe kvm-intel
```

6. Optionally, verify that APICv is now disabled.

```
root@host# cat /sys/module/kvm_intel/parameters/enable_apicv
```

```
N
```

Upgrade the Linux Kernel on Ubuntu

To upgrade to the latest stable Linux kernel on Ubuntu:

1. Get and install the available updated kernel.

   hostOS:$ **sudo apt-get install linux-image-generic-lts-utopic**
2. Reboot the host OS.

   hostOS:$ **reboot**
3. Optionally, type **uname -a** in a terminal on your host OS to verify that the host OS is using the latest kernel version.

   hostOS:$ **uname -a**

```
3.16.0-48-generic
```

Install vSRX with KVM

IN THIS SECTION

You use `virt-manager` or `virt-install` to install vSRX VMs. See your *host* OS documentation for complete details on these packages.

NOTE: To upgrade an existing vSRX instance, see *Migration, Upgrade, and Downgrade* in the *vSRX Release Notes*.

Install vSRX with virt-manager

Ensure that sure you have already installed KVM, qemu, virt-manager, and libvirt on your host OS. You must also configure the required virtual networks and storage pool in the host OS for the vSRX VM. See your host OS documentation for details.

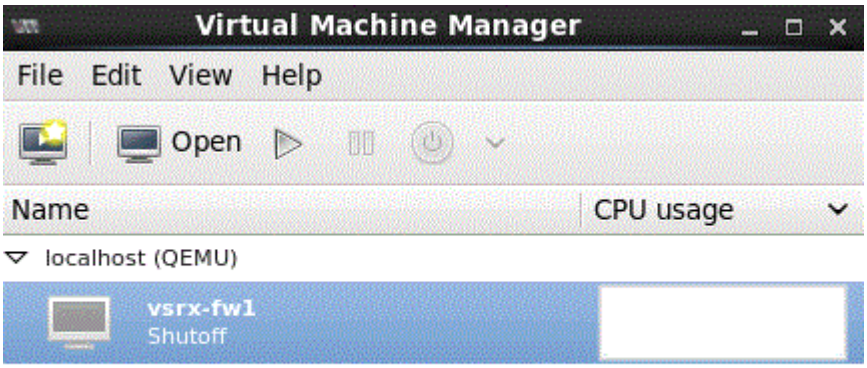
You can install and launch vSRX with the *KVM* `virt-manager` GUI package.

To install vSRX with `virt-manager`:

1. Download the vSRX QCOW2 image from the Juniper software download site.
2. On your host OS, type **virt-manager**. The Virtual Machine Manager appears. See Figure 2 on page 21.

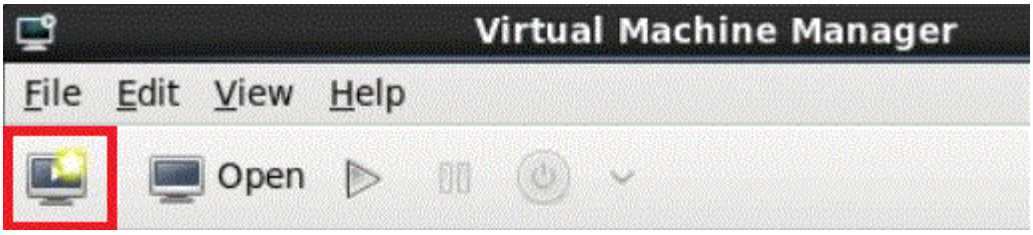
NOTE: You must have admin rights on the host OS to use `virt-manager`.

Figure 2: virt-manager



3. Click **Create a new virtual machine** as seen in Figure 3 on page 22. The New VM wizard appears .

Figure 3: Create a New Virtual Machine



4. Select **Import existing disk image**, and click **Forward**.
5. Browse to the location of the downloaded vSRX QCOW2 image and select the vSRX image.
6. Select **Linux** from the OS type list and select **Show all OS options** from the Version list.
7. Select **Red Hat Enterprise Linux 7** from the expanded Version list and click **Forward**.
8. Set the RAM to 4096 MB and set CPUs to 2. Click **Forward**.
9. Set the disk image size to 16 GB and click **Forward**.
10. Name the vSRX VM, and select **Customize this configuration before install** to change parameters before you create and launch the VM. Click **Finish**. The Configuration dialog box appears.
11. Select **Processor** and expand the **Configuration** list.
12. Select **Copy Host CPU Configuration**.
13. Set CPU Feature **invtsc** to disabled on CPUs that support that feature. Set **vmx** to require for optimal throughput. You can optionally set **aes** to require for improved cryptographic throughput

NOTE: If the CPU feature option is not present in your version of **virt-manager**, you need start and stop the VM once, and then edit the vSRX VM XML file, typically found in /etc/libvirt/qemu directory on your host OS. Use `virsh edit` to edit the VM XML file to configure <feature policy='require' name='vmx'/> under the <cpu mode> element. Also add <feature policy='disable' name='invtsc'/> if your host OS supports this CPU flag. Use the `virsh capabilities` command on your host OS to list the host OS and CPU virtualization capabilities. The following example shows the relevant portion of the vSRX XML file on a CentOS host:

```
<cpu mode='custom' match='exact'>
    <model fallback='allow'>SandyBridge</model>
    <vendor>Intel</vendor>
    <feature policy='require' name='pbe'/>
    <feature policy='require' name='tm2'/>
    <feature policy='require' name='est'/>
```

```
    <feature policy='require' name='vmx'/>
    <feature policy='require' name='osxsave'/>
    <feature policy='require' name='smx'/>
    <feature policy='require' name='ss'/>
    <feature policy='require' name='ds'/>
    <feature policy='require' name='vme'/>
    <feature policy='require' name='dtes64'/>
    <feature policy='require' name='monitor'/>
    <feature policy='require' name='ht'/>
    <feature policy='require' name='dca'/>
    <feature policy='require' name='pcid'/>
    <feature policy='require' name='tm'/>
    <feature policy='require' name='pdcm'/>
    <feature policy='require' name='pdpe1gb'/>
    <feature policy='require' name='ds_cpl'/>
    <feature policy='require' name='xtpr'/>
    <feature policy='require' name='acpi'/>
    <feature policy='disable' name='invtsc'/>
  </cpu>
```

14. Select the disk and expand **Advanced Options**.
15. Select **IDE** from the Disk bus list.
16. Select the NIC, and select **virtio** from the Device model field. This first NIC is the fpx0 (management) interface for vSRX.
17. Click **Add Hardware** to add more virtual networks, and select **virtio** from the Device model list.
18. Click **Apply**, and click **x** to close the dialog box.
19. Click **Begin Installation**. The VM manager creates and launches the vSRX VM.

NOTE: The default vSRX VM login ID is root with no password. By default, if a DHCP server is on the network, it assigns an IP address to the vSRX VM.

Install vSRX with virt-install

Ensure that sure you have already installed KVM, qemu, virt-install, and libvirt on your host OS. You must also configure the required virtual networks and storage pool in the host OS for the vSRX VM. See your host OS documentation for details.

NOTE: You must have root access on the host OS to use the `virt-install` command.

The `virt-install` and `virsh` tools are CLI alternatives to installing and managing vSRX VMs on a Linux host.

To install vSRX with `virt-install`:

1. Download the vSRX QCOW2 image from the Juniper software download site.
2. On your host OS, use the **virt-install** command with the mandatory options listed in Table 11 on page 24.

NOTE: See the official `virt-install` documentation for a complete description of available options.

Table 11: virt-install Options

Command Option	Description
--name *name*	Name the vSRX VM.
--ram *megabytes*	Allocate RAM for the VM, in megabytes.
--cpu *cpu-model*, *cpu-flags*	Enable the `vmx` feature for optimal throughput. You can also enable `aes` for improved cryptographic throughput. **NOTE**: CPU flag support depends on your host OS and CPU. Use `virsh capabilities` to list the virtualization capabilities of your host OS and CPU.
--vcpus *number*	Allocate the number of vCPUs for the vSRX VM.

Table 11: virt-install Options *(Continued)*

Command Option	Description
--disk *path*	Specify disk storage media and size for the VM. Include the following options: • size=*gigabytes* • device=disk • bus=ide • format=qcow2
--os-type *os-type* --os-variant *os-type*	Configure the guest OS type and variant.
--import	Create and boot the vSRX VM from an existing image.

The following example creates a vSRX VM with 4096 MB RAM, 2 vCPUs, and disk storage up to 16 GB:

```
hostOS# virt-install --name vSRXVM --ram 4096 --cpu SandyBridge,+vmx,-invtsc --vcpus=2 --
arch=x86_64 --disk path=/mnt/vsrx.qcow2,size=16,device=disk,bus=ide,format=qcow2 --os-type
linux --os-variant rhel7 --import
```

The following example shows the relevant portion of the vSRX XML file on a CentOS host:

```
<cpu mode='custom' match='exact'>
    <model fallback='allow'>SandyBridge</model>
    <vendor>Intel</vendor>
    <feature policy='require' name='pbe'/>
    <feature policy='require' name='tm2'/>
    <feature policy='require' name='est'/>
    <feature policy='require' name='vmx'/>
    <feature policy='require' name='osxsave'/>
    <feature policy='require' name='smx'/>
    <feature policy='require' name='ss'/>
    <feature policy='require' name='ds'/>
    <feature policy='require' name='vme'/>
```

```
    <feature policy='require' name='dtes64'/>
    <feature policy='require' name='monitor'/>
    <feature policy='require' name='ht'/>
    <feature policy='require' name='dca'/>
    <feature policy='require' name='pcid'/>
    <feature policy='require' name='tm'/>
    <feature policy='require' name='pdcm'/>
    <feature policy='require' name='pdpe1gb'/>
    <feature policy='require' name='ds_cpl'/>
    <feature policy='require' name='xtpr'/>
    <feature policy='require' name='acpi'/>
    <feature policy='disable' name='invtsc'/>
  </cpu>
```

NOTE: The default vSRX VM login ID is root with no password. By default, if a DHCP server is on the network, it assigns an IP address to the vSRX VM.

RELATED DOCUMENTATION

- Installing a virtual machine using virt-install
- Migration, Upgrade, and Downgrade
- Linux CPU Flags

Example: Install and Launch vSRX on Ubuntu

IN THIS SECTION

This example shows how to install and launch a vSRX instance on an Ubuntu server with KVM.

Requirements

This example uses the following hardware and software components:

- Generic x86 server
- Junos OS Release 15.1X49-D20 for vSRX
- Ubuntu version 14.04.2

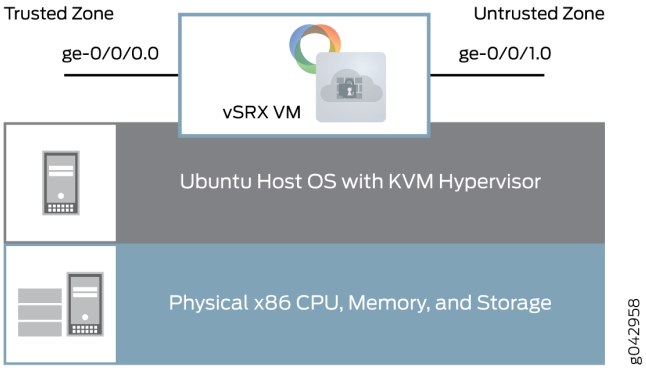
Before you begin:

- This example assumes a fresh install of the Ubuntu server software.
- Ensure that your host OS meets the requirements specified in *Requirements for vSRX on KVM*.

Overview

This example shows how to set up your Ubuntu host server and install and launch a vSRX VM. Figure 4 on page 27 shows the basic structure of a vSRX VM on an Ubuntu server.

Figure 4: vSRX VM on Ubuntu



NOTE: This example uses static IP addresses. If you are configuring the vSRX instance in an *NFV* environment, you should use DHCP.

Quick Configuration - Install and Launch a vSRX VM on Ubuntu

IN THIS SECTION

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and copy and paste the commands into the Ubuntu server terminal or vSRX console as specified.

Procedure

Step-by-Step Procedure

1. If the default virtual network does not already exist, copy the following commands and paste them into the Ubuntu server terminal to create the default virtual network.

```
cat <<EOF> /etc/libvirt/qemu/networks/default.xml
<network>
 <name>default</name>
 <forward mode='nat'/>
<nat>
 <port start ='1024' end='65535' />
 </nat>
<bridge name='virbr0' stp='on' delay='0' />
 <ip address='192.168.2.1' netmask='255.255.255.0'>
 <dhcp>
        <range start='192.168.2.2' end='192.168.2.254' />
 </dhcp>
 </ip>
</network>
 EOF
virsh net-define /etc/libvirt/qemu/networks/default.xml
virsh net-start default
```

```
virsh net-autostart default
```

2. Create the left, or trusted, virtual network on the Ubuntu server.

```
cat <<EOF> /etc/libvirt/qemu/networks/testleftnetwork.xml
<network>
 <name>TestLeft</name>
 <forward mode='route'/>
 <bridge name='virbr1' stp='on' delay='0' />
 <ip address='192.168.123.1' netmask='255.255.255.0'>
 <dhcp>
        <range start='192.168.123.100' end='192.168.123.250' />
 </dhcp>
 </ip>
</network>
 EOF
virsh net-define /etc/libvirt/qemu/networks/testleftnetwork.xml
virsh net-start TestLeft
virsh net-autostart TestLeft
```

3. Create the right, or untrusted, virtual network on the Ubuntu server.

```
cat <<EOF > /etc/libvirt/qemu/networks/testrightnetwork.xml
<network>
     <name>TestRight</name>
     <forward mode='nat'/>
<nat>
 <port start ='1024' end='65535' />
 </nat>
     <bridge name='virbr2' stp='on' delay='0' />
     <ip address='192.168.124.1' netmask='255.255.255.0'>
     <dhcp>
         <range start='192.168.124.100' end='192.168.124.250' />
    </dhcp>
 </ip>
</network>
EOF
virsh net-define /etc/libvirt/qemu/networks/testrightnetwork.xml
virsh net-start TestRight
```

```
virsh net-autostart TestRight
```

4. Download the vSRX KVM image from the Juniper Networks website at https://www.juniper.net/support/downloads/?p=vsrx#sw.

5. Copy the following commands and modify the `cpu` parameter and flags to match your Ubuntu server CPU. Paste the resulting commands into the Ubuntu server terminal to copy the image to a mount point and create the vSRX VM.

```
cp junos-vsrx-vmdisk-15.1X49-D20.2.qcow2 /mnt/vsrx20one.qcow2
virt-install  --name vSRX20One --ram 4096 --cpu SandyBridge,+vmx,-invtc, --vcpus=2 --
arch=x86_64 --disk path=/mnt/vsrx20one.qcow2,size=16,device=disk,bus=ide,format=qcow2 --os-
type linux --os-variant rhel7 --import --network=network:default,model=virtio --
network=network:TestLeft,model=virtio --network=network:TestRight,model=virtio
```

> **NOTE**: The CPU model and flags in the `virt-install` command might vary based on the CPU and features in the Ubuntu server.

6. To set the root password on the vSRX VM, copy and paste the command into the vSRX CLI at the `[edit]` hierarchy level.

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

7. To create a base configuration on the vSRX VM, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the following commands into the vSRX CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces fxp0 unit 0 family inet dhcp-client
set interfaces ge-0/0/0 unit 0 family inet address 192.168.123.254/24
set interfaces ge-0/0/1 unit 0 family inet dhcp-client
set security zones security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic system-
services all
set security zones security-zone untrust interfaces ge-0/0/1.0 host-inbound-traffic system-
services dhcp
set routing-instances CUSTOMER-VR instance-type virtual-router
set routing-instances CUSTOMER-VR interface ge-0/0/0.0
```

```
set routing-instances CUSTOMER-VR interface ge-0/0/1.0
set security nat source rule-set source-nat from zone trust
set security nat source rule-set source-nat to zone untrust
set security nat source rule-set source-nat  rule nat1 match source-address 0.0.0.0/0
set security nat source rule-set source-nat rule nat1 then source-nat interface
```

IN THIS SECTION

Step-by-Step Procedure

Step by Step Configuration

IN THIS SECTION

Use the following sections for a more detailed set of procedures to install and launch a vSRX VM.

Add Virtual Networks

Step-by-Step Procedure

You need to create virtual networks on the Ubuntu server to provide network connectivity to interfaces on the vSRX VM. Copy and paste these command into a terminal on the Ubuntu server.

This example uses three virtual networks:

- default— Connects the fxp0 management interface.

  NOTE: The default virtual network should already exist on the Ubuntu server. Use the `virsh net-list` command to verify that the default network is present and active.

- TestLeft— Connects the ge-0/0/0 interface to the trusted zone.
- TestRight— Connects the ge-0/0/1 interface to the untrusted zone.

1. If the default network does not exist, follow these steps:

   #### Step-by-Step Procedure

   a. Open a text editor on the Ubuntu server and create the default network XML (**default.xml**) file.

   ```
   emacs /etc/libvirt/qemu/networks/default.xml
   ```

   b. Set the forward mode to `nat`, configure an IP address and subnet mask, and a bridge interface, and configure DHCP to assign IP addresses to interfaces on this virtual network.

   NOTE: Use the XML format specified by libvirt.

   ```
   <network>
    <name>default</name>
    <forward mode='nat'/>
   <nat>
    <port start ='1024' end='65535' />
    </nat>
   <bridge name='virbr0' stp='on' delay='0' />
   ```

```
 <ip address='192.168.2.1' netmask='255.255.255.0'>
 <dhcp>
       <range start='192.168.2.2' end='192.168.2.254' />
 </dhcp>
 </ip>
</network>
```

c. Define and start the default virtual network, based on the **default.xml** file you created.

```
virsh net-define /etc/libvirt/qemu/networks/default.xml
virsh net-start default
virsh net-autostart default
```

2. Remove any previously configured TestLeft virtual network.

```
virsh net-destroy TestLeft
virsh net-undefine TestLeft
```

3. Remove any previously configured TestRight virtual network.

```
virsh net-destroy TestRight
virsh net-undefine TestRight
```

4. Open a text editor on the Ubuntu server and create the TestLeft network XML (**testleftnetwork.xml**) file.

```
emacs /etc/libvirt/qemu/networks/testleftnetwork.xml
```

5. Set the forward mode to `route`, configure an IP address and subnet mask, and a bridge interface, and configure DHCP to assign IP addresses to interfaces on this virtual network.

NOTE: Use the XML format specified by libvirt.

```
<network>
 <name>TestLeft</name>
 <forward mode='route'/>
 <bridge name='virbr1' stp='on' delay='0' />
 <ip address='192.168.123.1' netmask='255.255.255.0'>
 <dhcp>
        <range start='192.168.123.100' end='192.168.123.250' />
 </dhcp>
 </ip>
</network>
```

6. Open a text editor on the Ubuntu server and create the TestRight network XML (**testrightnetwork.xml**) file.

```
emacs /etc/libvirt/qemu/networks/testrightnetwork.xml
```

7. Set the forward mode to `nat`, configure an IP address and subnet mask, and a bridge interface, and configure DHCP to assign IP addresses to interfaces on this virtual network.

NOTE: Use the XML format specified by libvirt.

```
<network>
     <name>TestRight</name>
 <forward mode='nat'/>
<nat>
 <port start ='1024' end='65535' />
 </nat>
     <bridge name='virbr2' stp='on' delay='0' />
     <ip address='192.168.124.1' netmask='255.255.255.0'>
     <dhcp>
         <range start='192.168.124.100' end='192.168.124.250' />
    </dhcp>
 </ip>
</network>
```

8. Define and start the TestLeft virtual network, based on the **testleftnetwork.xml** file you created.

```
virsh net-define /etc/libvirt/qemu/networks/testleftnetwork.xml
virsh net-start TestLeft
virsh net-autostart TestLeft
```

9. Define and start the TestRight virtual network, based on the **testrightnetwork.xml** file you created.

```
virsh net-define /etc/libvirt/qemu/networks/testrightnetwork.xml
virsh net-start TestRight
virsh net-autostart TestRight
```

Verify the Virtual Networks

Purpose

Verify the new virtual network configuration on the Ubuntu server.

Action

Use the `virsh net-list` command on the Ubuntu server to verify that the new virtual interfaces are active and are set to autostart on reboot.

```
virsh net-list
```

```
 Name             State      Autostart     Persistent
----------------------------------------------------------
default                active      yes            yes
TestLeft               active      yes            yes
TestRight              active      yes            yes
```

Download and Installing the vSRX Image

Step-by-Step Procedure

To download and install the vSRX image on the Ubuntu server:

1. Download the vSRX KVM image from the Juniper Networks website: https://www.juniper.net/support/downloads/?p=vsrx#sw
2. Copy the vSRX image to an appropriate mount point.

```
hostOS# cp junos-vsrx-vmdisk-15.1X49-D20.2.qcow2 /mnt/vsrx20one.qcow2
```

3. Use the `virt-install` command to create a vSRX VM. Modify the `cpu` parameter and flags to match your Ubuntu server CPU.

```
hostOS# virt-install  --name vSRX20One --ram 4096 --cpu SandyBridge,+vmx,-invtc, --vcpus=2 --
arch=x86_64 --disk path=/mnt/vsrx20one.qcow2,size=16,device=disk,bus=ide,format=qcow2 --os-
type linux --os-variant rhel7 --import --network=network:default,model=virtio --
network=network:TestLeft,model=virtio --network=network:TestRight,model=virtio
```

NOTE: The CPU model and flags in the `virt-install` command might vary based on the CPU and features in the Ubuntu server.

Verify the vSRX Installation

Purpose

Verify the vSRX Installation.

Action

1. Use the `virsh console` command on the Ubuntu server to access the vSRX console and watch the progress of the installation. The installation can take several minutes to complete.

```
hostOS# virsh console vSRx200ne
```

```
Starting install...
ERROR    internal error: process exited while connecting to monitor: libust[11994/11994]:
Warning: HOME environment variable not set. Disabling LTTng-UST per-user tracing. (in
setup_local_apps() at lttng-ust-comm.c:305)
libust[11994/11995]: Error: Error opening shm /lttng-ust-wait-5 (in get_wait_shm() at lttng-
ust-comm.c:886)
libust[11994/11995]: Error: Error opening shm /lttng-ust-wait-5 (in get_wait_shm() at lttng-
ust-comm.c:886)

  Booting `Juniper Linux'

Loading Linux ...
Consoles: serial port
BIOS drive C: is disk0
BIOS drive D: is disk1
BIOS drive E: is disk2
BIOS drive F: is disk3
BIOS 639kB/999416kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.2
(builder@example.com, Thu Jul 30 23:20:10 UTC 2015)
Loading /boot/defaults/loader.conf
/kernel text=0xa3a2c0 data=0x6219c+0x11f8e0 syms=[0x4+0xb2ed0+0x4+0x1061bb]
/boot/modules/libmbpool.ko text=0xce8 data=0x114
/boot/modules/if_em_vsrx.ko text=0x184c4 data=0x7fc+0x20
/boot/modules/virtio.ko text=0x2168 data=0x208 syms=[0x4+0x7e0+0x4+0x972]
/boot/modules/virtio_pci.ko text=0x2de8 data=0x200+0x8 syms=[0x4+0x8f0+0x4+0xb22]
/boot/modules/virtio_blk.ko text=0x299c data=0x1dc+0xc syms=[0x4+0x960+0x4+0xa0f]
/boot/modules/if_vtnet.ko text=0x5ff0 data=0x360+0x10 syms=[0x4+0xdf0+0x4+0xf19]
/boot/modules/pci_hgcomm.ko text=0x12fc data=0x1a4+0x44 syms=[0x4+0x560+0x4+0x61d]
/boot/modules/chassis.ko text=0x9bc data=0x1d0+0x10 syms=[0x4+0x390+0x4+0x399]


Hit [Enter] to boot immediately, or space bar for command prompt.
```

```
Booting [/kernel]...
platform_early_bootinit: Early Boot Initialization
GDB: debug ports: sio
GDB: current port: sio
KDB: debugger backends: ddb gdb
KDB: current backend: ddb
Copyright (c) 1996-2015, Juniper Networks, Inc.
All rights reserved.
Copyright (c) 1992-2007 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
    The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
JUNOS 15.1X49-D15.4 #0: 2015-07-31 02:20:21 UTC

<output omitted>

The machine id is empty.
Cleaning up ...
Thu Aug 27 12:06:22 UTC 2015
Aug 27 12:06:22 init: exec_command: /usr/sbin/dhcpd (PID 1422) started
Aug 27 12:06:22 init: dhcp (PID 1422) started
Aug 27 12:06:23 init: exec_command: /usr/sbin/pppd (PID 1428) started

Amnesiac (ttyd0)

login:
```

2. On the vSRX console, log in and verify the vSRX version installed.

```
login: root
```

```
--- JUNOS 15.1X49-D15.4 built 2015-07-31 02:20:21 UTC

root@%
```

```
root@% cli
```

```
root>
```

```
root> show version
```

```
Model: vSRX
Junos: 15.1X49-D15.4
JUNOS Software Release [15.1X49-D15.4]
```

Create a Base Configuration on the vSRX Instance

Step-by-Step Procedure

To configure a base setup on the vSRX instance, enter the following steps in `edit` mode:

1. Create a root password.

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

2. Set the IP address family for the management interface, and enable the DHCP client for this interface.

```
set interfaces fxp0 unit 0 family inet dhcp-client
```

3. Set the IP address for the ge-0/0/0.0 interface.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.123.254/24
```

4. Set the IP address family for the ge-0/0/1.0 interface, and enable the DHCP client for this interface.

```
set interfaces ge-0/0/1 unit 0 family inet dhcp-client
```

5. Add the ge-0/0/0.0 interface to the trust security zone and allow all system services from inbound traffic on that interface.

```
set security zones security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic system-
services all
```

6. Add the ge-0/0/1.0 interface to the untrust security zone and allow only DHCP system services from inbound traffic on that interface.

```
set security zones security-zone untrust interfaces ge-0/0/1.0 host-inbound-traffic system-
services dhcp
```

7. Create a virtual router routing instance and add the two interfaces to that routing instance.

```
set routing-instances CUSTOMER-VR instance-type virtual-router
set routing-instances CUSTOMER-VR interface ge-0/0/0.0
set routing-instances CUSTOMER-VR interface ge-0/0/1.0
```

8. Create a source NAT rule set.

```
set security nat source rule-set source-nat from zone trust
set security nat source rule-set source-nat to zone untrust
```

9. Configure a rule that matches packets and translates the source address to the address of the egress interface.

```
set security nat source rule-set source-nat  rule nat1 match source-address 0.0.0.0/0
set security nat source rule-set source-nat rule nat1 then source-nat interface
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
show interfaces
```

From configuration mode, confirm your security policies by entering the `show security policies` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
show security policies
```

```
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone trust to-zone untrust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
```

```
        }
        then {
            permit;
        }
    }
}
from-zone untrust to-zone trust {
    policy default-deny {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
}
```

If you are done configuring the device, enter `commit` from configuration mode.

NOTE: As a final step, exit configuration mode and use the `request system reboot` command to reboot the vSRX VM. You can use the `virsh console` command on the Ubuntu server to reconnect to the vSRX after reboot.

Verify the Basic Configuration on the vSRX Instance

Purpose

Verify the basic configuration on the vSRX instance.

Action

Verify that the ge-0/0/0.0 interface has an assigned IP address from the TestLeft network DHCP address range, and that the ge-0/0/1.0 has an assigned IP address from the TestRight network DHCP address range.

```
root> show interfaces terse
```

```
Interface               Admin Link Proto    Local                 Remote
ge-0/0/0                up    up
ge-0/0/0.0              up    up   inet     192.168.123.254/24
gr-0/0/0                up    up
ip-0/0/0                up    up
lsq-0/0/0               up    up
lt-0/0/0                up    up
mt-0/0/0                up    up
sp-0/0/0                up    up
sp-0/0/0.0              up    up   inet
                                   inet6
sp-0/0/0.16383          up    up   inet
ge-0/0/1                up    up
ge-0/0/1.0              up    up   inet     192.168.124.238/24
dsc                     up    up
em0                     up    up
em0.0                   up    up   inet     128.0.0.1/2
em1                     up    up
em1.32768               up    up   inet     192.168.1.2/24
em2                     up    up
fxp0                    up    up
fxp0.0                  up    up   inet     192.168.2.1/24
ipip                    up    up
irb                     up    up
lo0                     up    up
lo0.16384               up    up   inet     127.0.0.1           --> 0/0
lo0.16385               up    up   inet     10.0.0.1            --> 0/0
                                            10.0.0.16           --> 0/0
                                            128.0.0.1           --> 0/0
                                            128.0.0.4           --> 0/0
                                            128.0.1.16          --> 0/0
lo0.32768               up    up
lsi                     up    up
```

```
mtun                    up    up
pimd                    up    up
pime                    up    up
pp0                     up    up
ppd0                    up    up
ppe0                    up    up
st0                     up    up
tap                     up    up
vlan                    up    down
```

RELATED DOCUMENTATION

- libvirt Network XML Format
- libvirt Command Reference

Load an Initial Configuration on a vSRX with KVM

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, you can use a mounted ISO image to pass the initial startup Junos OS configuration to a vSRX VM. This ISO image contains a file in the root directory called juniper.conf. This file uses the standard Junos OS command syntax to define configuration details, such as root password, management IP address, default gateway, and other configuration statements.

The process to bootstrap a vSRX VM with an ISO configuration image is as follows:

1. Create the **juniper.conf** configuration file with your Junos OS configuration.
2. Create an ISO image that includes the **juniper.conf** file.
3. Mount the ISO image to the vSRX VM.

4. Boot or reboot the vSRX VM. vSRX will boot using the **juniper.conf** file included in the mounted ISO image.
5. Unmount the ISO image from the vSRX VM.

> **NOTE**: If you do not unmount the ISO image after the initial boot or reboot, all subsequent configuration changes to the vSRX are overwritten by the ISO image on the next reboot.

Create a vSRX Bootstrap ISO Image

This task uses a Linux system to create the ISO image.

To create a vSRX bootstrap ISO image:

1. Create a configuration file in plaintext with the Junos OS command syntax and save in a file called **juniper.conf**.
2. Create a new directory.

```
hostOS$ mkdir iso_dir
```

3. Copy **juniper.conf** to the new ISO directory.

```
hostOS$ cp juniper.conf iso_dir
```

> **NOTE**: The **juniper.conf** file must contain the full vSRX configuration. The ISO bootstrap process overwrites any existing vSRX configuration.

4. Use the Linux `mkisofs` command to create the ISO image.

```
hostOS$ mkisofs -l -o test.iso iso_dir
```

```
I: -input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
```

```
Total rockridge attributes bytes: 0
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
175 extents written (0 MB)
```

NOTE: The `-l` option allows for a long filename.

Provision vSRX with an ISO Bootstrap Image on KVM

To provision a vSRX VM from an ISO bootstrap image:

1. Use the `virsh edit` command on the KVM host server where the vSRX VM resides to add the bootstrap ISO image as a disk device.

```
<disk type='file' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <source file='/home/test.iso'/>
      <target dev='hdc' bus='ide'/>
      <readonly/>
      <address type='drive' controller='0' bus='1' target='0' unit='0'/>
    </disk>
```

2. Boot or reboot the vSRX VM.

```
user@host# virsh start ixvSRX
```

```
Connected to domain ixvSRX
```

3. Optionally, use the `virsh domblklist` Linux command to verify that the bootstrap ISO image is part of the VM.

```
hostOS# virsh domblklist ixvSRX
```

```
Target     Source
------------------------------------------------
```

```
hda        /home/test/vsrx209.qcow2
hdc        /home/test/test.iso
```

4. Verify the configuration, then power down the vSRX VM to remove the ISO image.
5. Use the `virsh edit` command on the KVM host server to remove the ISO image xml statements added in step 1, and then reboot the vSRX VM.

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, you can use a mounted ISO image to pass the initial startup Junos OS configuration to a vSRX VM. This ISO image contains a file in the root directory called juniper.conf. This file uses the standard Junos OS command syntax to define configuration details, such as root password, management IP address, default gateway, and other configuration statements.

RELATED DOCUMENTATION

- Linux mkisofs command

Use Cloud-Init in an OpenStack Environment to Automate the Initialization of vSRX Instances

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX image to help simplify configuring new vSRX instances operating in an OpenStack environment according to a specified user-data file. Cloud-init is performed during the first-time boot of a vSRX instance.

Cloud-init is an OpenStack software package for automating the initialization of a cloud instance at boot-up. It is available in Ubuntu and most major Linux and FreeBSD operating systems. Cloud-init is designed to support multiple different cloud providers so that the same virtual machine (VM) image can

be directly used in multiple hypervisors and cloud instances without any modification. Cloud-init support in a VM instance runs at boot time (first-time boot) and initializes the VM instance according to the specified user-data file.

A user-data file is a special key in the metadata service that contains a file that cloud-aware applications in the VM instance can access upon a first-time boot. In this case, it is the validated Junos OS configuration file that you intend to upload to a vSRX instance as the active configuration. This file uses the standard Junos OS command syntax to define configuration details, such as root password, management IP address, default gateway, and other configuration statements.

When you create a vSRX instance, you can use cloud-init with a validated Junos OS configuration file (**juniper.conf**) to automate the initialization of new vSRX instances. The user-data file uses the standard Junos OS syntax to define all the configuration details for your vSRX instance. The default Junos OS configuration is replaced during the vSRX instance launch with a validated Junos OS configuration that you supply in the form of a user-data file.

NOTE: If using a release *earlier* than Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, the user-data configuration file cannot exceed 16 KB. If your user-data file exceeds this limit, you must compress the file using gzip and use the compressed file. For example, the gzip junos.conf command results in the junos.conf.gz file.

Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, if using a configuration drive data source in an OpenStack environment, the user-data configuration file size can be up to 64 MB.

The configuration must be validated and include details for the fxp0 interface, login, and authentication. It must also have a default route for traffic on fxp0. If any of this information is missing or incorrect, the instance is inaccessible and you must launch a new one.

WARNING: Ensure that the user-data configuration file is not configured to perform autoinstallation on interfaces using Dynamic Host Configuration Protocol (DHCP) to assign an IP address to the vSRX. Autoinstallation with DHCP will result in a "commit fail" for the user-data configuration file.

Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, the cloud-init functionality in vSRX has been extended to support the use of a configuration drive data source in an OpenStack environment. The configuration drive uses the user-data attribute to pass a validated Junos OS configuration file to the vSRX instance. The user-data can be plain text or MIME file type text/plain. The configuration drive is typically used in conjunction with the Compute service, and is present to the instance as a disk partition labeled `config-2`. The configuration drive has a maximum size of 64 MB, and must be formatted with either the vfat or ISO 9660 filesystem.

The configuration drive data source also provides the flexibility to add more than one file that can be used for configuration. A typical use case would be to add a Day0 configuration file and a license file. In this case, there are two methods that can be employed to use a configuration drive data source with a vSRX instance:

- User-data (Junos OS Configuration File) alone—This approach uses the user-data attribute to pass the Junos OS configuration file to each vSRX instance. The user-data can be plain text or MIME file type text/plain.
- Junos OS configuration file and license file—This approach uses the configuration drive data source to send the Junos OS configuration and license file(s) to each vSRX instance.

> **NOTE**: If a license file is to be configured in vSRX, it is recommended to use the `-file` option rather than the `user-data` option to provide the flexibility to configure files larger than the 16 KB limit of user-data.

To use a configuration drive data source to send Junos OS configuration and license file(s) to a vSRX instance, the files needs to be sent in a specific folder structure. In this application, the folder structure of the configuration drive data source in vSRX is as follows:

```
-   OpenStack
        -   latest
              -   junos-config
                      -  configuration.txt
              -   junos-license
                      -  License_file_name.lic
                      -  License_file_name.lic
```

`//OpenStack//latest/junos-config/configuration.txt`

`//OpenStack//latest/junos-license/license.lic`

Before you begin:

- Create a configuration file with the Junos OS command syntax and save it. The configuration file can be plain text or MIME file type text/plain. The string `#junos-config` must be the first line of the user-data configuration file before the Junos OS configuration.

> **NOTE**: The `#junos-config` string is mandatory in the user-data configuration file; if it is not included, the configuration will not be applied to the vSRX instance as the active configuration.

- Determine the name for the vSRX instance you want to initialize with a validated Junos OS configuration file.
- Determine the flavor for your vSRX instance, which defines the compute, memory, and storage capacity of the vSRX instance.
- Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, if using a configuration drive, ensure the following criteria is met to enable cloud-init support for a configuration drive in OpenStack:
  - The configuration drive must be formatted with either the `vfat` or `iso9660` filesystem.

    NOTE: The default format of a configuration drive is an ISO 9660 file system. To explicitly specify the ISO 9660/vfat format, add the `config_drive_format=iso9660/vfat` line to the `nova.conf` file.

  - The configuration drive must have a filesystem label of `config-2`.
  - The folder size must be no greater than 64 MB.

Depending on your OpenStack environment, you can use either an OpenStack command-line interface (such as `nova boot` or `openstack server create`) or the OpenStack Dashboard ("Horizon") to launch and initialize a vSRX instance.

Perform Automatic Setup of a vSRX Instance Using an OpenStack Command-Line Interface

You can launch and manage a vSRX instance using either the `nova boot` or `openstack server create` commands, which includes the use of a validated Junos OS configuration user-data file from your local directory to initialize the active configuration of the target vSRX instance.

To initiate the automatic setup of a vSRX instance from an OpenStack command-line client:

1. If you have not done so already, create a configuration file with the Junos OS command syntax and save the file. The configuration file can be plain text or MIME file type text/plain.

   The user-data configuration file must contain the full vSRX configuration that is to be used as the active configuration on each vSRX instance, and the string `#junos-config` must be the first line of the user-data configuration file before the Junos OS configuration.

   NOTE: The `#junos-config` string is mandatory in the user-data configuration file; if it is not included, the configuration will not be applied to the vSRX instance as the active configuration.

2. Copy the Junos OS configuration file to an accessible location from where it can be retrieved to launch the vSRX instance.
3. Depending on your OpenStack environment, use the `nova boot` or `openstack server create` command to launch the vSRX instance with a validated Junos OS configuration file as the specified user-data.

> **NOTE**: You can also use the `nova boot` equivalent in an Orchestration service such as HEAT.

For example:

- `nova boot -user-data` **`</path/to/vsrx_configuration.txt>`** `--image` **`vSRX_image`** `--flavor` **`vSRX_flavor_instance`**
- `openstack server create -user-data` **`</path/to/vsrx_configuration.txt>`** `--image` **`vSRX_image`** `--flavor` **`vSRX_flavor_instance`**

  Where:

  `-user-data` **`</path/to/vsrx_configuration.txt>`** specifies the location of the Junos OS configuration file. The user-data configuration file size is limited to approximately 16,384 bytes.

  `--image` **`vSRX_image`** identifies the name of a unique vSRX image.

  `--flavor` **`vSRX_flavor_instance`** identifies the vSRX flavor (ID or name).

Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, to enable the use of a configuration drive for a specific request in the OpenStack compute environment, include the `-config-drive true` parameter in the `nova boot` or `openstack server create` command.

> **NOTE**: It is possible to enable the configuration drive automatically on all instances by configuring the OpenStack Compute service to always create a configuration drive. To do this, specify the `force_config_drive=True` option in the `nova.conf` file.

For example, to use the user-data attribute to pass the Junos OS configuration to each vSRX instance:

`nova boot -config-drive true -flavor` **`vSRX_flavor_instance`** `-image` **`vSRX_image`** `-user-data` **`</path/to/vsrx_configuration.txt>`**

Where:

`-user-data` **`</path/to/vsrx_configuration.txt>`** specifies the location of the Junos OS configuration file. The user-data configuration file size is limited to approximately 64 MB.

`-image` **`vSRX_image`** identifies the name of a unique vSRX image.

`-flavor` **`vSRX_flavor_instance`** identifies the vSRX flavor (ID or name).

For example, to specify the configuration drive with multiple files (Junos OS configuration file and license file):

`nova boot -config-drive true -flavor` **`vSRX_flavor_instance`** `-image` **`vSRX_image`** `[-file` **`/junos-config/configuration.txt=/path/to/file`**`] [-file` **`/junos-license/license.lic=path/to/license`**`]`

Where:

`[-file` **`/junos-config/configuration.txt=/path/to/file`**`]` specifies the location of the Junos OS configuration file.

`[-file` **`/junos-license/license.lic=path/to/license`**`]` specifies the location of the Junos OS configuration file.

`-image` **`vSRX_image`** identifies the name of a unique vSRX image.

`-flavor` **`vSRX_flavor_instance`** identifies the vSRX flavor (ID or name).

4. Boot or reboot the vSRX instance. During the initial boot-up sequence, the vSRX instance processes the cloud-init request.

   NOTE: The boot time for the vSRX instance might increase with the use of the cloud-init package. This additional time in the initial boot sequence is due to the operations performed by the cloud-init package. During this operation, the cloud-init package halts the boot sequence and performs a lookup for the configuration data in each data source identified in the cloud.cfg. The time required to look up and populate the cloud data is directly proportional to the number of data sources defined. In the absence of a data source, the lookup process continues until it reaches a predefined timeout of 30 seconds for each data source.

5. When the initial boot-up sequence resumes, the user-data file replaces the original factory-default Junos OS configuration loaded on the vSRX instance. If the commit succeeds, the factory-default configuration will be permanently replaced. If the configuration is not supported or cannot be applied to the vSRX instance, the vSRX will boot using the default Junos OS configuration.

SEE ALSO

- Cloud-Init Documentation
- OpenStack command-line clients
- Compute service (nova) command-line client
- Openstack Server Create

Enabling the configuration drive (configdrive)

Instances

Perform Automatic Setup of a vSRX Instance from the OpenStack Dashboard (Horizon)

Horizon is the canonical implementation of the OpenStack Dashboard. It provides a Web-based user interface to OpenStack services including Nova, Swift, Keystone, and so on. You can launch and manage a vSRX instance from the OpenStack Dashboard, which includes the use of a validated Junos OS configuration user-data file from your local directory to initialize the active configuration of the target vSRX instance.

To initiate the automatic setup of a vSRX instance from the OpenStack Dashboard:

1. If you have not done so already, create a configuration file with the Junos OS command syntax and save the file. The configuration file can be plain text or MIME file type text/plain.

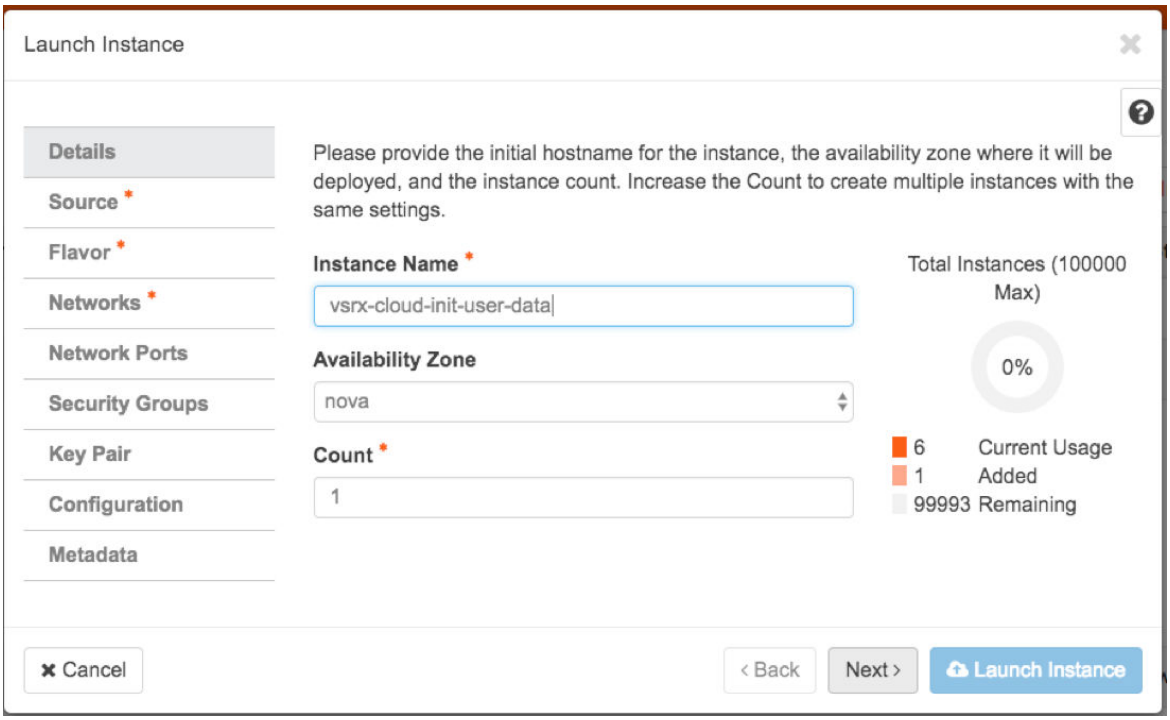
   The user-data configuration file must contain the full vSRX configuration that is to be used as the active configuration on each vSRX instance, and the string `#junos-config` must be the first line of the user-data configuration file before the Junos OS configuration.

   > **NOTE**: The `#junos-config` string is mandatory in the user-data configuration file; if it is not included, the configuration will not be applied to the vSRX instance as the active configuration.

2. Copy the Junos OS configuration file to an accessible location from where it can be retrieved to launch the vSRX instance.
3. Log in to the OpenStack Dashboard using your login credentials and then select the appropriate project from the drop-down menu at the top left.
4. On the Project tab, click the **Compute** tab and select **Instances**. The dashboard shows the various instances with its image name, its private and floating IP addresses, size, status, availability zone, task, power state, and so on.
5. Click **Launch Instance**. The Launch Instance dialog box appears.

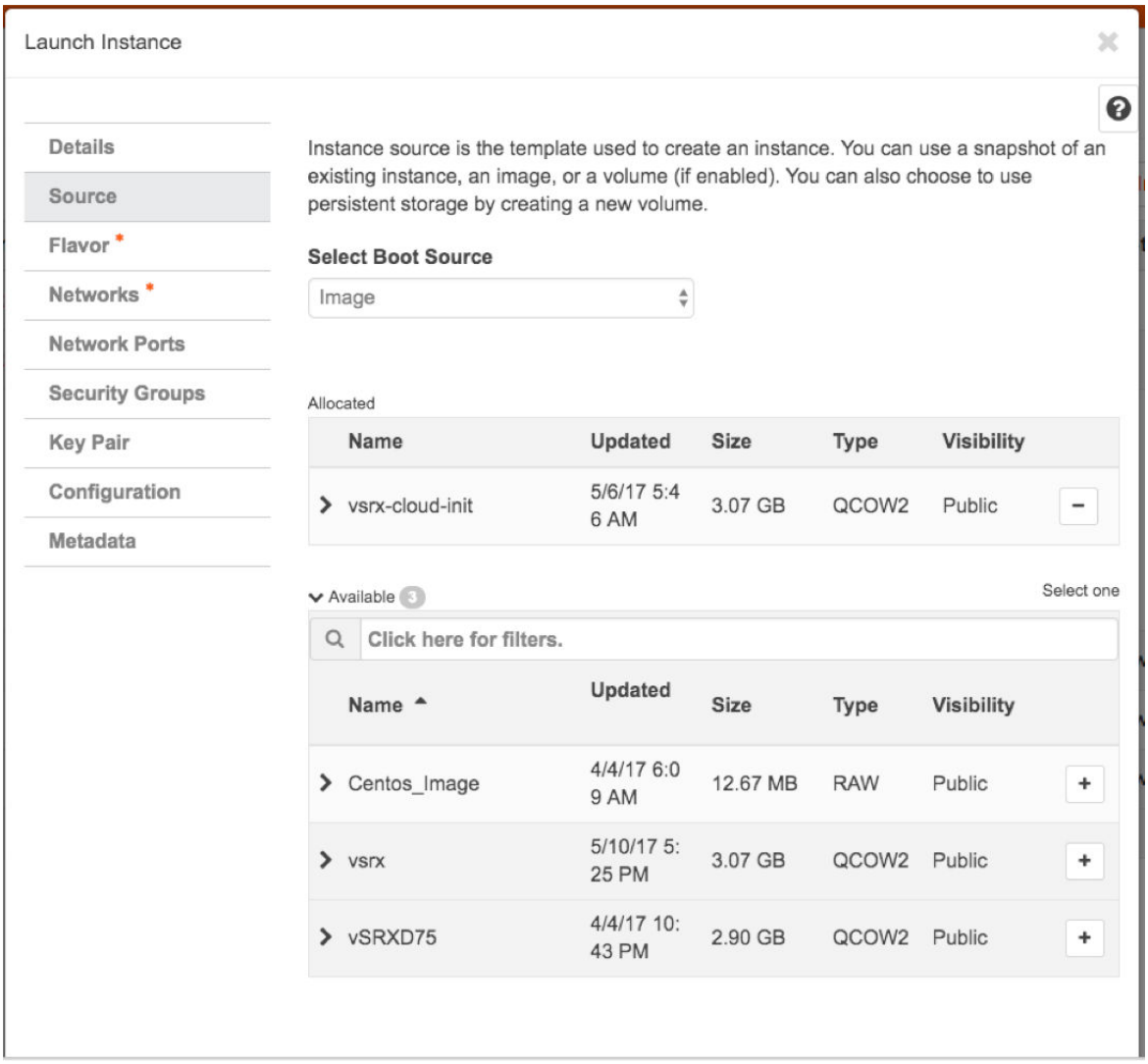
6. From the Details tab (see Figure 5 on page 54), enter an instance name for the vSRX VM along with the associated availability zone (for example, Nova) and then click **Next**. We recommend that you keep this name the same as the hostname assigned to the vSRX VM.

Figure 5: Launch Instance Details Tab



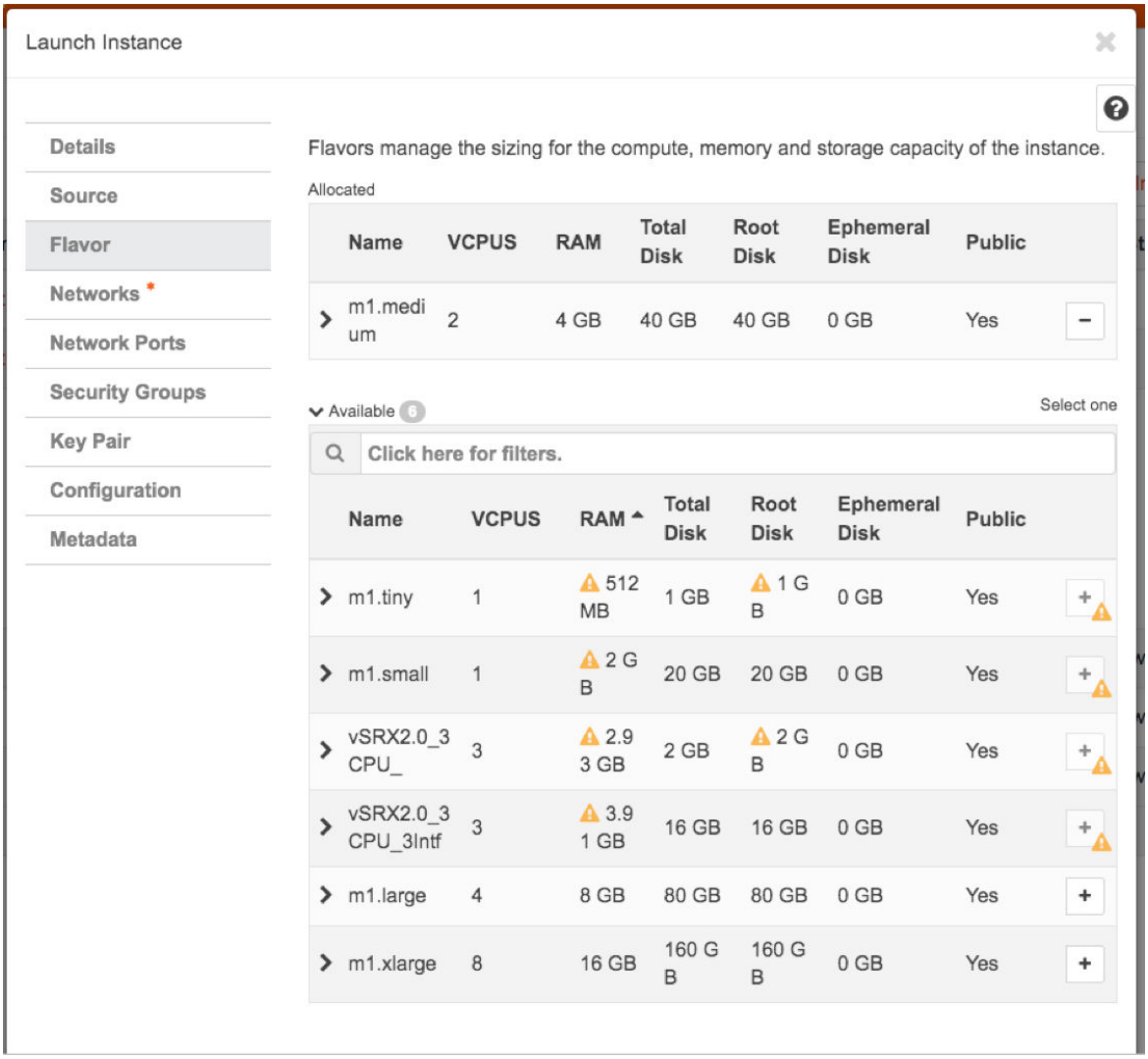
7. From the Source tab (see Figure 6 on page 55), select a vSRX VM image source file from the Available list and then click **+(Plus)**. The selected vSRX image appears under Allocated. Click **Next**.

Figure 6: Launch Instance Source Tab



8. From the Flavor tab (see Figure 7 on page 56), select a vSRX instance with a specific compute, memory, and storage capacity from the Available list and then click **+(plus sign)**. The selected vSRX flavor appears under Allocated. Click **Next**.

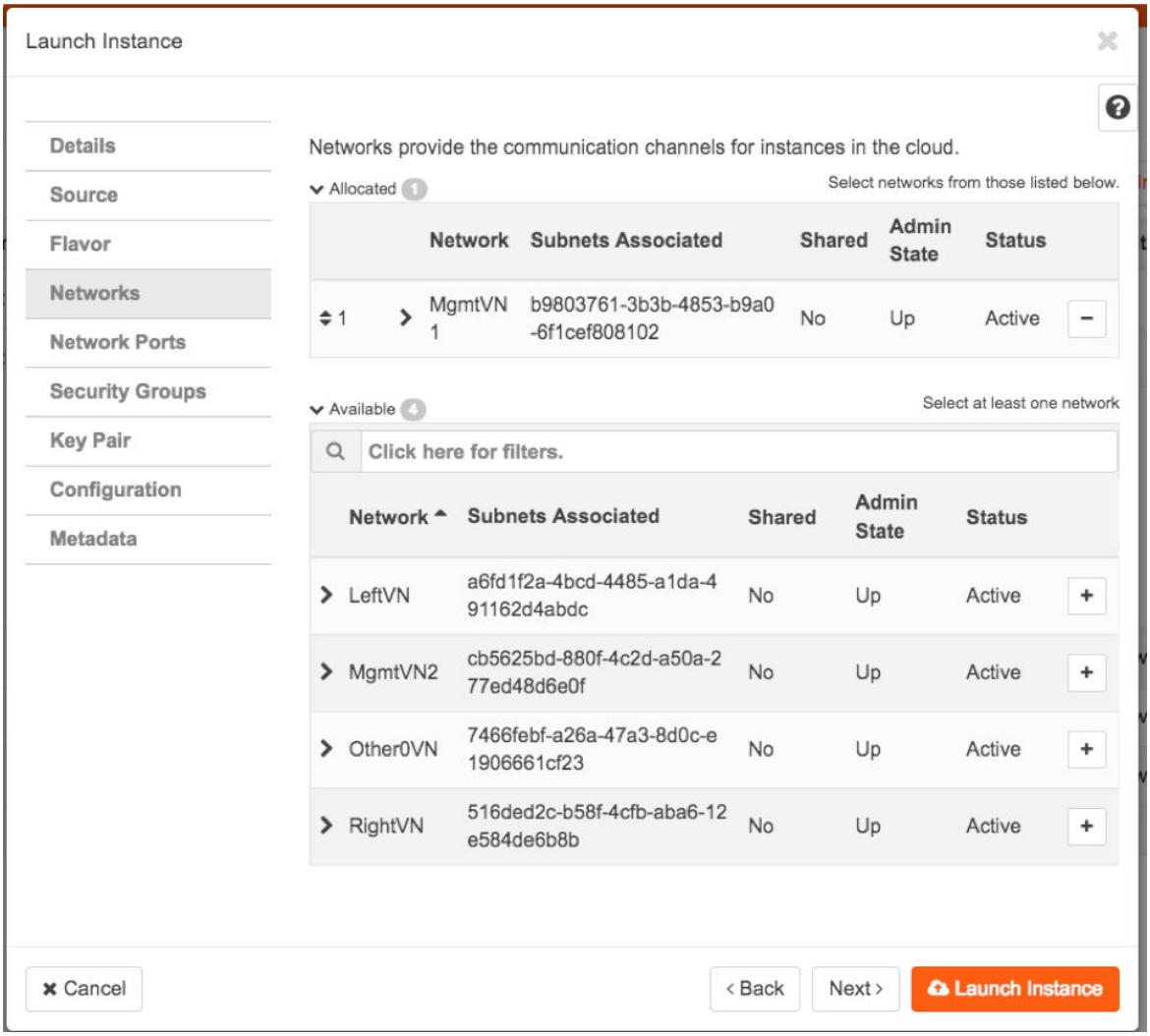
Figure 7: Launch Instance Flavor Tab



9. From the Networks tab (see Figure 8 on page 57), select the specific network of the vSRX instance from the Available list and then click **+(plus sign)**. The selected network appears under Allocated. Click **Next**.

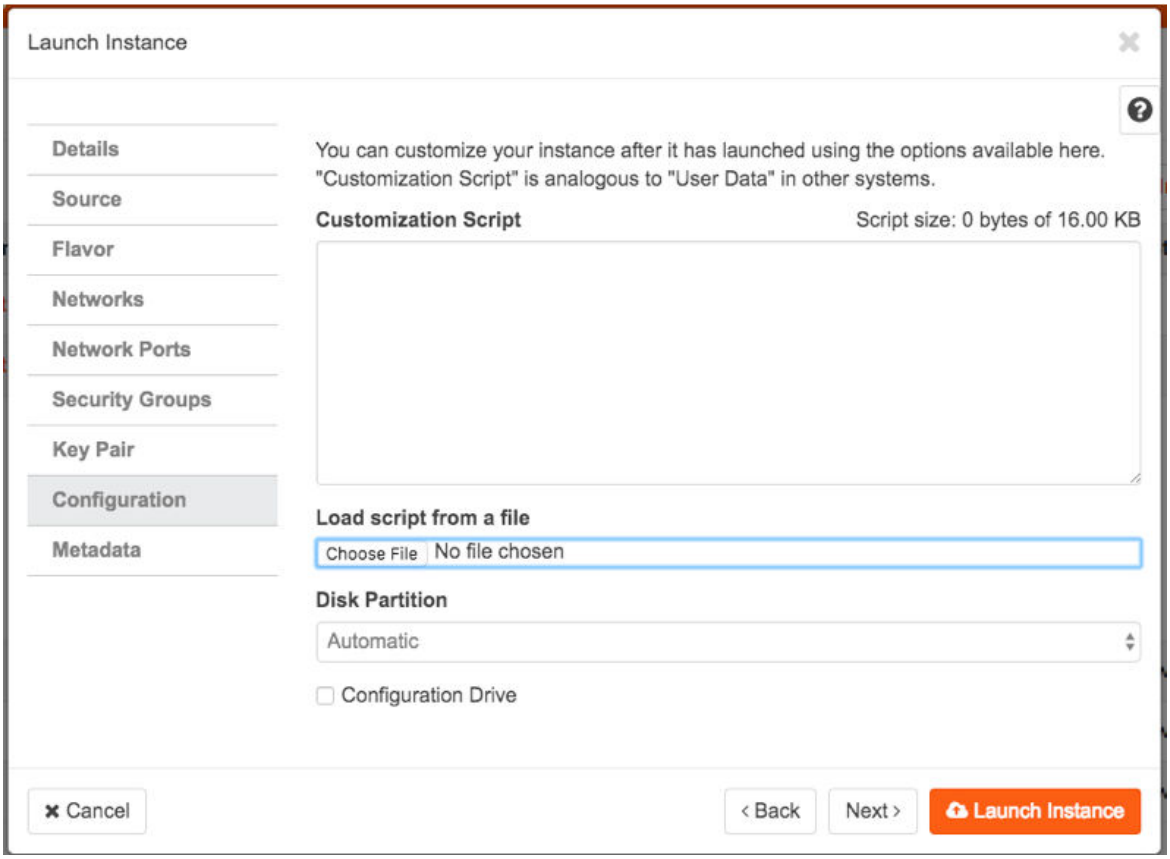
NOTE: Do not update any parameters in the Network Ports, Security Groups, or Key Pair tabs in the Launch Instance dialog box.

Figure 8: Launch Instance Networks Tab



10. From the Configuration tab (see Figure 9 on page 58), click **Browse** and navigate to the location of the validated Junos OS configuration file from your local directory that you want to use as the user-data file. Click **Next**.

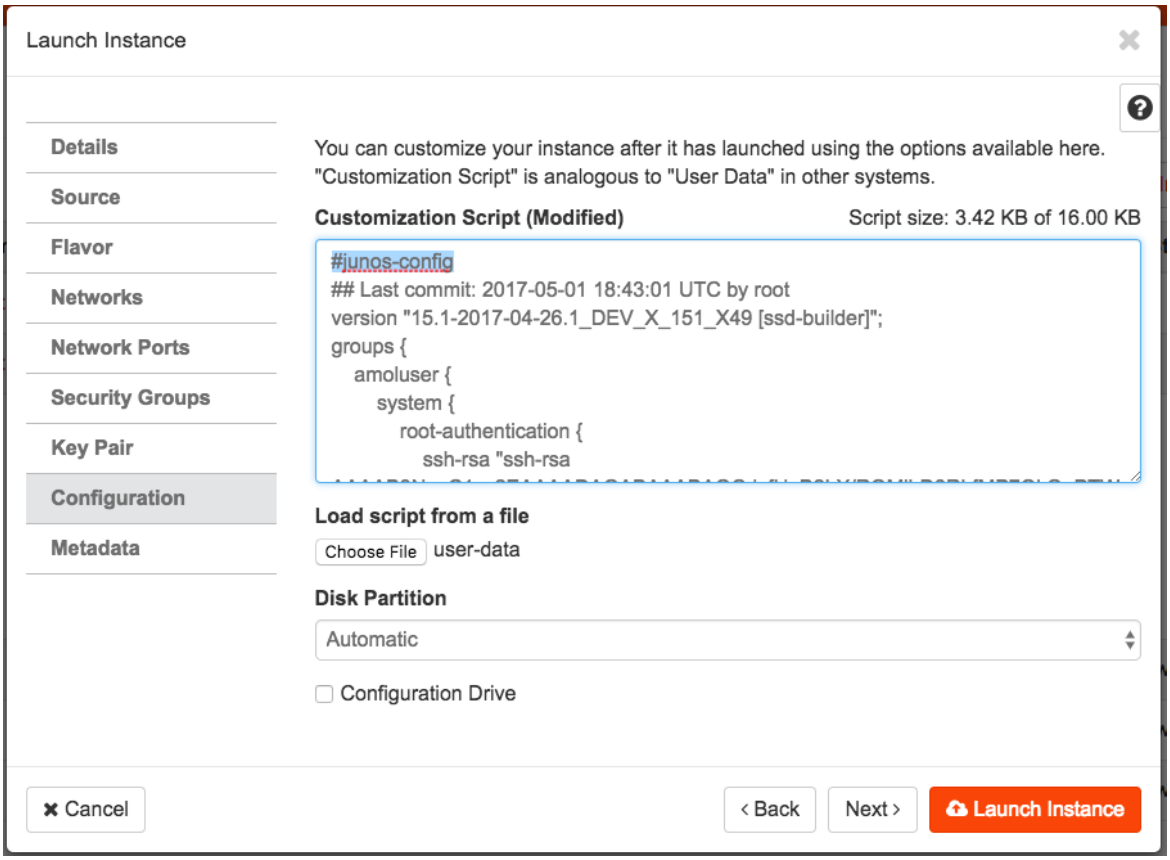
Figure 9: Launch Instance Configuration Tab



11. Confirm that the loaded Junos OS configuration contains the `#junos-config` string in the first line of the user-data configuration file (see Figure 10 on page 59) and then click **Next**.

NOTE: Do not update any parameters in the Metadata tab of the Launch Instance dialog box.

Figure 10: Launch Instance Configuration Tab with Loaded Junos OS Configuration



12. Click **Launch Instance**. During the initial boot-up sequence, the vSRX instance processes the cloud-init request.

NOTE: The boot time for the vSRX instance might increase with the use of the cloud-init package. This additional time in the initial boot sequence is due to the operations performed by the cloud-init package. During this operation, the cloud-init package halts the boot sequence and performs a lookup for the configuration data in each data source identified in the cloud.cfg. The time required to look up and populate the cloud data is directly proportional to the number of data sources defined. In the absence of a data source, the lookup process continues until it reaches a predefined timeout of 30 seconds for each data source.

13. When the initial boot-up sequence resumes, the user-data file replaces the original factory-default Junos OS configuration loaded on the vSRX instance. If the commit succeeds, the factory-default configuration will be permanently replaced. If the configuration is not supported or cannot be applied to the vSRX instance, the vSRX will boot using the default Junos OS configuration.

SEE ALSO

- Cloud-Init Documentation
- OpenStack Dashboard
- Launch and Manage Instances
- Horizon: The OpenStack Dashboard Project

Release History Table

Release	Description
15.1X49-D130	Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, the cloud-init functionality in vSRX has been extended to support the use of a configuration drive data source in an OpenStack environment. The configuration drive uses the user-data attribute to pass a validated Junos OS configuration file to the vSRX instance.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX image to help simplify configuring new vSRX instances operating in an OpenStack environment according to a specified user-data file. Cloud-init is performed during the first-time boot of a vSRX instance.

CHAPTER 3

vSRX VM Management with KVM

IN THIS CHAPTER

Configure vSRX Using the CLI

To configure the vSRX instance using the CLI:

1. Verify that the vSRX is powered on.
2. Log in as the root user. There is no password.
3. Start the CLI.

```
root#cli
root@>
```

4. Enter configuration mode.

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

5. Set the root authentication password by entering a *cleartext* password, an encrypted password, or an SSH public key string (*DSA* or *RSA*).

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

6. Configure the hostname.

```
[edit]
root@# set system host-name host-name
```

7. Configure the management interface.

```
[edit]
root@# set interfaces fxp0 unit 0 family inet dhcp-client
```

8. Configure the traffic interfaces.

```
[edit]
root@# set interfaces ge-0/0/0 unit 0 family inet dhcp-client
```

9. Configure basic security zones and bind them to traffic interfaces.

```
[edit]
root@# set security zones security-zone trust interfaces ge-0/0/0.0
```

10. Verify the configuration.

```
[edit]
root@# commit check
configuration check succeeds
```

11. Commit the configuration to activate it on the vSRX instance.

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

12. Optionally, use the `show` command to display the configuration to verify that it is correct.

> **NOTE**: Certain Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds to each licensed feature. To conform to software feature licensing requirements, you must purchase one license per feature per instance. The presence of the appropriate software unlocking key on your virtual instance allows you to configure and use the licensed feature. See Managing Licenses for vSRX for details.

RELATED DOCUMENTATION

- CLI User Guide

Connect to the vSRX Management Console on KVM

Ensure that you have the `virt-manager` package or `virsh` installed on your *host* OS.

To connect to the vSRX management console using `virt-manager`:

1. Launch `virt-manager`.
2. Highlight the vSRX VM you want to connect to from the list of VMs displayed.
3. Click **Open**.
4. Select **View>Text Consoles>Serial 1**. The vSRX console appears.

To connect to the vSRX VM with `virsh`:

1. Use the `virsh console` command on the Linux host OS.

```
user@host# virsh console vSRX-kvm-2
```

```
Connected to domain vSRX-kvm-2
```

2. The vSRX console appears.

RELATED DOCUMENTATION

virt-tools

Add a Virtual Network to a vSRX VM with KVM

You can extend an existing vSRX *VM* to use additional virtual networks.

To create a *virtual network* with `virt-manager`:

1. Launch `virt-manager` and select **Edit>Connection Details**. The Connection details dialog box appears.
2. Select **Virtual Networks**. The list of existing virtual networks appears.
3. Click **+** to create a new virtual network for the control link. The Create a new virtual network wizard appears.
4. Set the subnet for this virtual network and click **Forward**.
5. Optionally, select **Enable DHCP** and click **Forward**.
6. Select the network type from the list and click **Forward**.
7. Verify the settings and click **Finish** to create the virtual network.

To create a virtual network with `virsh`:

1. Use the `virsh net-define` command on the *host* OS to create an XML file that defines the new virtual network. Include the XML fields described in Table 12 on page 65 to define this network.

> **NOTE**: See the official `virsh` documentation for a complete description of available options, including how to configure IPv6 networks.

Table 12: virsh net-define XML Fields

Field	Description
<network>...</network>	Use this XML wrapper element to define a virtual network.
<name>*net-name*</name>	Specify the virtual network name.
<bridge name="*bridge-name*" />	Specify the name of the host bridge used for this virtual network.
<forward mode="*forward-option*" />	Specify routed or nat. Do not use the <forward> element for isolated mode.
<ip address="*ip-address*" netmask="*net-mask*" <dhcp range start="*start*" end="*end*" </dhcp> </ip>	Specify the IP address and subnet mask used by this virtual network, along with the DHCP address range.

The following example shows a sample XML file that defines a new virtual network.

```
<network>
 <name>mgmt</name>
 <bridge name="vbr1" />
 <forward mode="nat" />
 <ip address="10.10.10.1" netmask="255.255.255.0" >
  <dhcp>
<range start="10.10.10.2" end="10.10.10.99" />
  </dhcp>
 </ip>
</network>
```

2. Use the `virsh net-start` command in the host OS to start the new virtual network.

   hostOS# **virsh net-start mgmt**

3. Use the `virsh net-autostart` command in the host OS to automatically start the new virtual network when the host OS boots.

   hostOS# **virsh net-autostart mgmt**

4. Optionally, use the `virsh net-list -all` command in the host OS to verify the new virtual network.

```
HostOS# # virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 mgmt                 active     yes           yes
 default              active     yes           yes
```

RELATED DOCUMENTATION

virt tools

Add a Virtio Virtual Interface to a vSRX VM with KVM

You can add additional *virtio* virtual interfaces to an existing vSRX *VM* with KVM.

To add additional virtio virtual interfaces to a vSRX VM using `virt-manager`:

1. In `virt-manager`, double-click the vSRX VM and select **View>Details**. The vSRX Virtual Machine details dialog box appears.
2. Click **Add Hardware**. The Add Hardware dialog box appears.
3. Select **Network** from the left navigation panel.
4. Select the host device or virtual network on which you want this new virtual interface from the Network source list.
5. Select **virtio** from the Device model list and click **Finish**.
6. From the vSRX console, reboot the vSRX instance.

   `vsrx#` **request system reboot**.

   vSRX reboots both Junos OS and the vSRX guest VM.

To add additional virtio virtual interfaces to a vSRX VM using `virsh`:

1. Use the `virsh attach-interface` command on the host OS with the mandatory options listed in Table 13 on page 67.

   NOTE: See the official `virsh` documentation for a complete description of available options.

Table 13: virsh attach-interface Options

Command Option	Description
--domain *name*	Specify the name of the guest VM.
--type	Specify the host OS connection type as `bridge` or `network`.
--source *interface*	Specify the physical or logical interface on the host OS to associate with this vNIC.
--target *vnic*	Specify the name for the new vNIC.
--model	Specify the vNIC model.

The following example creates a new virtio vNIC from the host OS virbr0 bridge.

```
user@host# virsh attach-interface --domain vsrxVM --type bridge --source virbr0 --target vsrx-mgmt --model virtio
```

```
Interface attached successfully
```

```
user@host# virsh dumpxml vsrxVM
```

```
<output omitted>

 <interface type='bridge'>
      <mac address='00:00:5e:00:53:e8'/>
      <source bridge='virbr0'/>
      <target dev='vsrx-mgmt'/>
      <model type='virtio'/>
      <alias name='net1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
    </interface>
```

2. From the vSRX console, reboot the vSRX instance.

   `vsrx#` **request system reboot**.

   vSRX reboots both Junos OS and the VSRX guest VM.

RELATED DOCUMENTATION

virt tools

Configure SR-IOV and PCI on KVM

IN THIS SECTION

This section includes the following topics on SR-IOV for a vSRX instance deployed on KVM:

SR-IOV Overview

vSRX on KVM supports single-root I/O virtualization (*SR-IOV*) interface types. SR-IOV is a standard that allows a single physical NIC to present itself as multiple vNICs, or virtual functions (VFs), that a *virtual machine* (VM) can attach to. SR-IOV combines with other virtualization technologies, such as Intel VT-d, to improve the I/O performance of the VM. SR-IOV allows each VM to have direct access to packets queued up for the VFs attached to the VM. You use SR-IOV when you need I/O performance that approaches that of the physical bare metal interfaces.

NOTE: SR-IOV in KVM does not remap interface numbers. The interface sequence in the vSRX VM XML file matches the interface sequence shown in the Junos OS CLI on the vSRX instance.

SR-IOV uses two PCI functions:

- Physical Functions (PFs)—Full PCIe devices that include SR-IOV capabilities. Physical Functions are discovered, managed, and configured as normal PCI devices. Physical Functions configure and

manage the SR-IOV functionality by assigning Virtual Functions. When SR-IOV is disabled, the host creates a single PF on one physical NIC.

- Virtual Functions (VFs)—Simple PCIe functions that only process I/O. Each Virtual Function is derived from a Physical Function. The number of Virtual Functions a device may have is limited by the device hardware. A single Ethernet port, the Physical Device, may map to many Virtual Functions that can be shared to guests. When SR-IOV is enabled, the host creates a single PF and multiple VFs on one physical NIC. The number of VFs depends on the configuration and driver support.

SR-IOV HA Support with Trust Mode Disabled (KVM only)

IN THIS SECTION

Understand SR-IOV HA Support with Trust Mode Disabled (KVM only)

A Redundant Ethernet Interface (RETH) is a virtual interface consisting of equal number of member interfaces from each participating node of an SRX cluster. All logical configurations such as IP address, QoS, zones, and VPNs are bound to this interface. Physical properties are applied to the member or child interfaces. A RETH interface has a virtual MAC address which is calculated using the cluster id. RETH has been implemented as an aggregated interface/LAG in Junos OS. For a LAG, the parent (logical) IFDs MAC address is copied to each of the child interfaces. When you configure the child interface under the RETH interface, the RETH interface's virtual MAC gets overwritten on the **current MAC address** field of the child physical interface. This also requires the virtual MAC address to be programmed on the corresponding NIC.

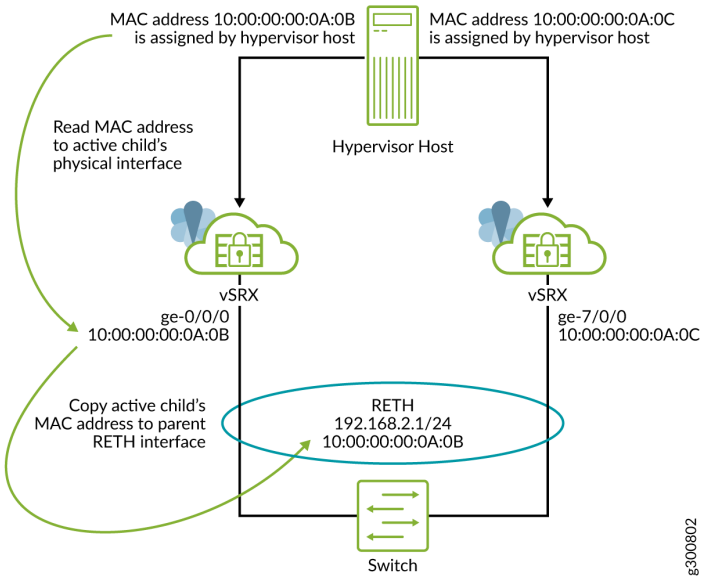
Junos OS runs as a VM on vSRX. Junos OS does not have direct access to the NIC and only has a virtual NIC access provided by the hypervisor which might be shared with other VMs running on the same host machine. This virtual access comes with certain restrictions such as a special mode called trust mode, which is required to program a virtual MAC address on the NIC. During deployments, providing the trust mode access might not be feasible because of possible security issues. To enable RETH model to work in such environments, MAC rewrite behavior is modified. Instead of copying the parent virtual MAC address to the children, we keep the children's physical MAC address intact and copy the physical MAC address of the child belonging to the active node of the cluster to the *current MAC* of the reth interface. This way, MAC rewrite access is not required when trust mode is disabled.

In case of vSRX, the DPDK reads the physical MAC address provided by the hypervisor and shares it with the Junos OS control plane. In standalone mode, this physical MAC address is programmed on the physical IFDs. But the support for the same is unavailable in cluster mode, because of which the MAC address for the physical interface is taken from the Juniper reserved MAC pool. In an environment where trust mode is not feasible, the hypervisor is unable to provide the physical MAC address.

To overcome this problem, we have added support to use the hypervisor provided physical MAC address instead of allocating it from the reserved MAC pool. See "Configure SR-IOV support with Trust Mode Disabled (KVM only)" on page 70.

Configure SR-IOV support with Trust Mode Disabled (KVM only)

Figure 11: Copying MAC address from active child interface to parent RETH



Starting in Junos OS Release 19.4R1, SR-IOV HA is supported with trust mode disabled. You can enable this mode by configuring the `use-active-child-mac-on-reth` and `use-actual-mac-on-physical-interfaces` configuration statements at the `[edit chassis cluster]` hierarchy level. If you configure commands in a cluster, the hypervisor assigns the child physical interface's MAC address and the parent RETH interface's MAC address is overwritten by the active child physical interface's MAC address

You need to reboot the vSRX instance to enable this mode. Both the nodes in the cluster need to be rebooted for the commands to take effect.

You need to configure the commands `use-active-child-mac-on-reth` and `use-actual-mac-on-physical-interfaces` together to enable this feature.

SEE ALSO

> use-active-child-mac-on-reth
>
> use-actual-mac-on-physical-interfaces

Limitations

SR-IOV HA support with trust mode disabled on KVM has the following limitations:

- SR-IOV HA support with trust mode disabled is only supported on KVM based systems.
- A reth interface can have maximum one port as a member on each vSRX cluster node.
- You cannot use `security nat proxy-arp` feature for NAT pools because no G-ARP is sent out on failover for the IP addresses in NAT pools. Instead, one can set the routes to the NAT pool range in the upstream router to point to the vSRX reth interface's IP address as the next-hop. Or, if directly connected hosts need to access the NAT pool addresses, these NAT pool addresses can be configured for proxy ARP under the reth interface.
- If the reth interface is configured with many VLANs, it might take some time to send all the G-ARPs on a failover. This might lead to a noticeable interruption in traffic.
- A dataplane failover will result in a change of the MAC address of the reth interface. Therefore the failover is not transparent to directly connected neighboring Layer 3 devices (routers or servers). The vSRX reth IP address must be mapped to a new MAC address in the ARP table on the neighboring devices. vSRX will send out a G-ARP which will help these devices. In case these neighboring devices do not act on the G-ARP received from the vSRX or show a slow response, the traffic might be interrupted until that device updates it's ARP table correctly.

Configure an SR-IOV Interface on KVM

If you have a physical NIC that supports SR-IOV, you can attach SR-IOV-enabled vNICs or virtual functions (VFs) to the vSRX instance to improve performance. We recommend that if you use SR-IOV, all revenue ports are configured as SR-IOV.

Note the following about SR-IOV support for vSRX on KVM:

- Starting in Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1, a vSRX instance deployed on KVM supports SR-IOV on an Intel X710/XL710 NIC in addition to Intel 82599 or X520/540.
- Starting in Junos OS Release 18.1R1, a vSRX instance deployed on KVM supports SR-IOV on the Mellanox ConnectX-3 and ConnectX-4 Family Adapters.

NOTE: See the *vSRX Performance Scale Up* discussion in *Understand vSRX with KVM* for the vSRX scale up performance when deployed on KVM, based on vNIC and the number of vCPUs and vRAM applied to a vSRX VM.

Before you can attach an SR-IOV enabled VF to the vSRX instance, you must complete the following tasks:

- Insert an SR-IOV-capable physical network adapter in the host server.
- Enable the Intel VT-d CPU virtualization extensions in BIOS on your host server. The Intel VT-d extensions provides hardware support for directly assigning a physical devices to guest. Verify the process with the vendor because different systems have different methods to enable VT-d.
- Ensure that SR-IOV is enabled at the system/server BIOS level by going into the BIOS settings during the host server boot-up sequence to confirm the SR-IOV setting. Different server manufacturers have different naming conventions for the BIOS parameter used to enable SR-IOV at the BIOS level. For example, for a Dell server ensure that the **SR-IOV Global Enable** option is set to **Enabled**.

NOTE: We recommend that you use `virt-manager` to configure SR-IOV interfaces. See the `virsh attach-device` command documentation if you want to learn how to add a PCI host device to a VM with the `virsh` CLI commands.

To add an SR-IOV VF to a vSRX VM using the `virt-manager` graphical interface:

1. In the Junos OS CLI, shut down the vSRX VM if it is running.

```
vsrx> request system power-off
```

2. In `virt-manager`, double-click the vSRX VM and select **View>Details**. The vSRX Virtual Machine details dialog box appears.
3. Select the Hardware tab, then click **Add Hardware**. The Add Hardware dialog box appears.
4. Select **PCI Host Device** from the Hardware list on the left.
5. Select the SR-IOV VF for this new virtual interface from the host device list.
6. Click **Finish** to add the new device. The setup is complete and the vSRX VM now has direct access to the device.
7. From the virt-manager icon bar at the upper-left side of the window, click the Power On arrow. The vSRX VM starts. Once the vSRX is powered on the Running status will display in the window.

   You can connect to the management console to watch the boot-up sequence.

NOTE: After the boot starts, you need to select **View>Text Consoles>Serial 1** in `virt-manager` to connect to the vSRX console.

To add an SR-IOV VF to a vSRX VM using `virsh` CLI commands:

1. Define four virtual functions for eno2 interface, update the sriov_numvfs file with number 4.

```
root@LabHost:~# echo 4 > /sys/class/net/eno2/device/sriov_numvfs
root@LabHost:~# more /sys/class/net/eno2/device/sriov_numvfs
```

2. Identify the device.

   Identify the PCI device designated for device assignment to the virtual machine. Use the `lspci` command to list the available PCI devices. You can refine the output of `lspci` with `grep`.

   Use command **lspci** to check the VF number according to the VF ID.

```
root@ kvmsrv:~# lspci | grep Ether
```

```
.....
83:00.0 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev
02)  -  Physical Function
83:00.1 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev
02)  -  Physical Function
83:02.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:02.1 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:02.2 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:02.3 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:02.4 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:02.5 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:02.6 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:02.7 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:0a.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:0a.1 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:0a.2 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:0a.3 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:0a.4 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:0a.5 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
83:0a.6 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
```

```
83:0a.7 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev 02)
.....
```

3. Add SR-IOV device assignment from a vSRX XML profile on KVM and review device information.

   The driver could use either vfio or kvm, depends on KVM server OS/kernel version and drivers for virtualization support. The address type references the unique PCI slot number for each SR-IOV VF (Virtual Function).

   Information on the domain, bus, and function are available from output of the `virsh nodedev-dumpxml` command.

```
<interface type="hostdev" managed="yes">
<driver name="vfio"/>
<source>
<address type="pci" domain="0x0000" bus="0x83" slot="0x02" function="0x3"/>
</source>
<address type="pci" domain="0x0000" bus="0x00" slot="0x05" function="0x0"/>
</interface>
```

4. Add PCI device in edit setting and select VF according to the VF number.

   NOTE: This operation should be done when VM is powered off. Also, do not clone VMs with PCI devices which might lead to VF or MAC conflict.

5. Start the VM using the `# virsh start` *name of virtual machine* command.

Release History Table

Release	Description
18.1R1	Starting in Junos OS Release 18.1R1, a vSRX instance deployed on KVM supports SR-IOV on the Mellanox ConnectX-3 and ConnectX-4 Family Adapters.
15.1X49-D90	Starting in Junos OS Release 15.1X49-D90 and Junos OS Release 17.3R1, a vSRX instance deployed on KVM supports SR-IOV on an Intel X710/XL710 NIC in addition to Intel 82599 or X520/540.

RELATED DOCUMENTATION

Requirements for vSRX on KVM | 7

- Intel SR-IOV Explanation
- PCI-SIG SR-IOV Primer
- SR-IOV
- Intel - SR-IOV Configuration Guide
- Red Hat - SR–IOV - PCI Devices

Upgrade a Multi-core vSRX

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, you can use `virt-manager` to scale the performance and capacity of a vSRX instance by increasing the number of vCPUs or the amount of vRAM allocated to the vSRX. See *Requirements for vSRX on KVM* for the software requirement specifications for a vSRX VM.

See your *host* OS documentation for complete details on the `virt-manager` package

NOTE: You cannot scale down the number of vCPUs or decrease the amount of vRAM for an existing vSRX VM.

Configure the Queue Value for vSRX VM with KVM

Before you plan to scale up vSRX performance, modify the vSRX VM XML file to configure network multi-queuing as a means to support an increased number of dataplane vCPUs for the vSRX VM. This setting updates the libvirt driver to enable multi-queue virtio-net so that network performance can scale as the number of dataplane vCPUs increases. Multi-queue virtio is an approach that enables the processing of packet sending and receiving to be scaled to the number of available virtual CPUs (vCPUs) of a guest, through the use of multiple queues.

The configuration of multi-queue virtio-net, however, can only be performed in the XML file. OpenStack does not support multi-queue.

To update the queue, at the `<driver name='vhost' queues='x'/>` line in the vSRX VM XML file, match the number of queues with number of dataplane vCPUs you plan to configure for the vSRX VM. The default is 4 dataplane vCPUs, but you can scale that number to 4, 8, or 16 vCPUs.

The following XML file example configures 8 queues for a vSRX VM with 8 dataplane vCPUs:

```
<output omitted>

 <interface type='network'>
        <source network='net2'/>
      <model type='virtio'/>
      <driver name='vhost' queues='8'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </interface>
```

Shutdown the vSRX Instance with virt-manager

In situations where you want to edit and modify the vSRX VM XML file, you need to completely shut down vSRX and the associated VM.

To gracefully shutdown the vSRX instance with `virt-manager`:

1. Launch `virt-manager`.
2. Check the vSRX instance you want to power off.
3. Select **Open** to open a console window to the vSRX instance.
4. From the vSRX console, reboot the vSRX instance.

   `vsrx#` **request system power-off**.
5. From `virt-manager`, select **Shut Down** to completely shutdown the VM so you can edit the XML file.

NOTE: Do not use **Force Reset** or **Force Off** on any active VM as it may create file corruptions.

Upgrade vSRX with virt-manager

You must shut down the vSRX VM before you can update vCPU or vRAM values for the VM.

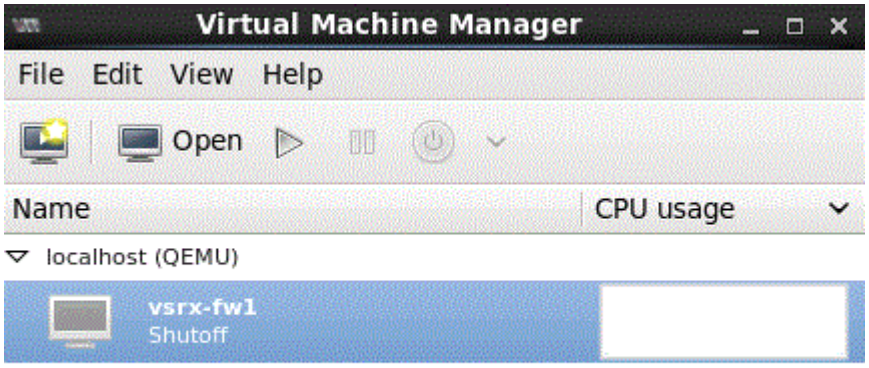
You can upgrade and launch vSRX with the *KVM* `virt-manager` GUI package.

To scale up a vSRX VM with `virt-manager` to a higher number of vCPUs or to an increased amount of vRAM:

1. On your host OS, type **virt-manager**. The Virtual Machine Manager appears. See Figure 12 on page 77.

> **NOTE**: You must have admin rights on the host OS to use `virt-manager`.

Figure 12: virt-manager



2. Select **Open** to open the powered down vSRX VM and select **Edit Hardware Details** to open the virtual machine details window.
3. Select **Processor** and set the number of vCPUs. Click **Apply**.
4. Select **Memory** and set the vRAM to the desired size. Click **Apply**.
5. Click **Power On**. The VM manager launches the vSRX VM with the new vCPU and vRAM settings.

> **NOTE**: vSRX scales down to the closest supported value if the vCPU or vRAM settings do not match what is currently available.

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, you can use virt-manager to scale the performance and capacity of a vSRX instance by increasing the number of vCPUs or the amount of vRAM allocated to the vSRX

RELATED DOCUMENTATION

- Understand vSRX with KVM | 2
- Requirements for vSRX on KVM | 7
- Installing a virtual machine using virt-install

Monitor the vSRX VM in KVM

You can monitor the overall state of the vSRX *VM* with `virt-manager` or `virsh`.

To monitor the vSRX VM with `virt-manager`:

1. From the virt-manager GUI, select the vSRX VM you want to monitor.
2. Select **View>Graph** and select the statistics you want to monitor. Options include CPU, memory, disk I/O, and network interface statistics.

   The window updates with thumbnail graphs for the statistics you selected.
3. Optionally, double-click on the thumbnail graph to expand the view.

To monitor the vSRX VM with `virsh`, use the commands listed in Table 14 on page 78.

Table 14: virsh Monitor Commands

Command	Description
`virsh cpu-stats` *vm-name*	Lists the CPU statistics for the VM.
`virsh domifstat` *vm-name interface-name*	Displays the vNIC statistics for the VM.
`virsh dommemstat` *vm-name*	Displays memory statistics for the VM.
`virsh vcpuinfo` *vm-name*	Displays vCPU details for the VM.
`virsh nodecpustats`	Displays CPU statistics for the host OS.

RELATED DOCUMENTATION

- virt tools

Manage the vSRX Instance on KVM

IN THIS SECTION

Each vSRX instance is an independent *VM* that you can power on, pause, or shut down. You can manage the vSRX VM with multiple tools, including `virt-manager` and `virsh`.

Power On the vSRX Instance with virt-manager

To power on the vSRX instance with `virt-manager`:

1. Launch `virt-manager`.
2. Check the vSRX instance you want to power on.
3. From the icon bar, select the power on arrow. The vSRX VM starts. You can connect to the management console to watch the boot-up sequence.

> **NOTE**: After the boot starts, you need to select **View>Text Consoles>Serial 1** in `virt-manager` to connect to the vSRX console.

Power On the vSRX Instance with virsh

To power on the vSRX instance with `virsh`:

Use the `virsh start` command on the *host* OS to start a vSRX VM.

```
user@host# virsh start vSRX-kvm-2
```

```
Domain vSRX-kvm-2 started
```

Pause the vSRX Instance with virt-manager

To pause the vSRX instance with `virt-manager`:

1. Launch `virt-manager`.
2. Check the vSRX instance you want to pause.
3. From the icon bar, select the power on pause icon. The vSRX VM pauses.

Pause the vSRX Instance with virsh

To pause the vSRX instance with `virsh`:

Use the `virsh suspend` command on the host OS to pause a vSRX VM.

```
user@host# virsh suspend vSRX-kvm-2
```

```
Domain vSRX-kvm-2 suspended
```

Rebooting the vSRX Instance with virt-manager

To reboot the vSRX instance with `virt-manager`:

1. Launch `virt-manager`.
2. Check the vSRX instance you want to reboot.
3. Select **Open** to open a console window to the vSRX instance.
4. From the vSRX console, reboot the vSRX instance.

   `vsrx#` **request system reboot**.

   vSRX reboots both Junos OS and the VSRX guest VM.

Reboot the vSRX Instance with virsh

To reboot the vSRX VM with `virsh`:

1. Use the `virsh console` command on the host OS to connect to the vSRX VM.
2. On the vSRX console, use the `request system reboot` command to reboot Junos OS and the vSRX VM.

```
user@host# virsh console vSRX-kvm-2
```

```
Connected to domain vSRX-kvm-2
```

```
vsrx# request system reboot
```

Power Off the vSRX Instance with virt-manager

To power off the vSRX instance with `virt-manager`:

1. Launch `virt-manager`.
2. Check the vSRX instance you want to power off.
3. Select **Open** to open a console window to the vSRX instance.
4. From the vSRX console, power off the vSRX instance.

```
vsrx> request system power-off
```

vSRX powers off both Junos OS and the guest VM.

Power Off the vSRX Instance with virsh

To power off the vSRX instance with `virsh`:

1. Use the `virsh console` command on the host OS to connect to the vSRX VM.

2. On the vSRX console, use the `request system power-off` command to power off Junos OS and the vSRX VM.

```
user@host# virsh console vSRX-kvm-2
```

```
Connected to domain vSRX-kvm-2
```

```
vsrx# request system power-off
```

Shutdown the vSRX Instance with virt-manager

In situations where you want to edit and modify the vSRX VM XML file, you need to completely shut down vSRX and the associated VM.

To gracefully shutdown the vSRX instance with `virt-manager`:

1. Launch `virt-manager`.
2. Check the vSRX instance you want to power off.
3. Select **Open** to open a console window to the vSRX instance.
4. From the vSRX console, reboot the vSRX instance.

   `vsrx#` **request system power-off**.
5. From `virt-manager`, select **Shut Down** to completely shutdown the VM so you can edit the XML file.

> **NOTE**: Do not use **Force Reset** or **Force Off** on any active VM as it may create file corruptions.

Shutdown the vSRX Instance with virsh

In situations where you want to modify the vSRX VM XML file, you need to completely shut down vSRX and the associated VM.

To gracefully shutdown the vSRX instance with `virsh`:

1. Use the `virsh console` command on the host OS to connect to the vSRX VM.
2. On the vSRX console, use the `request system power-off` command to power off Junos OS and the vSRX VM.

3. On the host OS, use the `virsh shutdown` command to shut down the VM after vSRX has powered off.

```
user@host# virsh console vSRX-kvm-2
```

```
Connected to domain vSRX-kvm-2
```

```
vsrx# request system power-off
user@host# virsh shutdown vSRX-kvm-2
```

NOTE: Do not use the `virsh destroy` command on any active VM as it may create file corruptions.

Remove the vSRX Instance with virsh

In situations where you want to completely remove the vSRX instance, you need to destroy the vSRX VM and undefine the associated XML file.

To completely remove the vSRX instance with `virsh`:

1. On the host OS, use the `virsh destroy` command to destroy the vSRX VM.
2. On the host OS, use the `virsh undefine` command to undefine the vSRX XML file.

```
user@host# virsh destroy vSRX-kvm-2
user@host# virsh undefine vSRX-kvm-2
```

RELATED DOCUMENTATION

virt tools

Recover the Root Password for vSRX in a KVM Environment

If you forget the root password for a vSRX instance deployed in a KVM environment, use this password recovery procedure to reset the root password. (KB article 31790).

NOTE: You need console access to recover the root password

To recover the root password for a vSRX instance:

1. Reboot the vSRX instance by entering the `virsh reboot` command, specifying either *domain-id* or *domain-name*.
2. Immediately attempt to login to the vSRX instance by entering the `virsh console` *domain-name* command to access the vSRX console.

   NOTE: We recommend that you specify *domain-name* when attempting the vSRX instance login. It is possible that *domain-id* might change after the vSRX instance reboot.

3. After login you will see a prompt similar to `Escape character is ^]`. Press **Enter** two or three times until the boot process begins. Continue with the boot process until you see the following prompt:

```
Hit [Enter] to boot immediately, or space bar for command prompt. Booting [kernel] in 9
seconds...
```

4. Press the space bar two or three times to stop the boot sequence, and then enter `boot -s` to login to single-user mode.
5. Enter `recovery` to start the root password recovery procedure.

```
Enter full pathname of shell or 'recovery' for root password recovery  or RETURN for /bin/
sh: recovery
```

   Once the script terminates you will be in vSRX operational mode.

6. Enter configuration mode in the CLI.
7. Set the root password.

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

8. Enter the new root password.

```
New password: xxxxxxxx
Retype new password:
```

9. At the second prompt, reenter the new root password.
10. If you are finished configuring the new root password for the vSRX instance, commit the configuration.

```
root@host# commit
commit complete
```

11. Exit from configuration mode.
12. Exit from operational mode.
13. Enter `y` to reboot the device.

```
Reboot the system? [y/n] y
```

The start up messages display on the screen.

14. Once again, press the space bar two or three times to access the bootstrap loader prompt.
15. The vSRX instance starts up again and prompts you to enter a user name and password. Enter the newly configured password.

```
login: root
Password: xxxxxxxx
```

CHAPTER 4

Configure vSRX Chassis Clusters on KVM

IN THIS CHAPTER

Configure a vSRX Chassis Cluster in Junos OS

IN THIS SECTION

Chassis Cluster Overview

Chassis cluster groups a pair of the same kind of vSRX instances into a cluster to provide network node redundancy. The vSRX instances in a chassis cluster must be running the same Junos OS release, and each instance becomes a node in the chassis cluster. You connect the control virtual interfaces on the respective nodes to form a *control plane* that synchronizes the configuration and Junos OS kernel state on both nodes in the cluster. The control link (a *virtual network* or *vSwitch*) facilitates the redundancy of interfaces and services. Similarly, you connect the *data plane* on the respective nodes over the fabric virtual interfaces to form a unified data plane. The fabric link (a virtual network or vSwitch) allows for the management of cross-node flow processing and for the management of session redundancy.

The control plane software operates in active/passive mode. When configured as a chassis cluster, one node acts as the primary and the other as the secondary to ensure stateful failover of processes and

services in the event of a system or hardware failure on the primary . If the primary fails, the secondary takes over processing of control plane traffic.

NOTE: If you configure a chassis cluster across two hosts, disable igmp-snooping on the bridge that each host physical interface belongs to and that the control virtual NICs (vNICs) use. This ensures that the control link heartbeat is received by both nodes in the chassis cluster.

The chassis cluster data plane operates in active/active mode. In a chassis cluster, the data plane updates session information as traffic traverses either node, and it transmits information between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, traffic can enter the cluster on one node and exit from the other node.

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple *Packet Forwarding Engine*s. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.
- Support for generic routing encapsulation (*GRE*) and IP-over-IP (IP-IP) tunnels used to route encapsulated IPv4 or *IPv6* traffic by means of two internal interfaces, gr-0/0/0 and ip-0/0/0, respectively. Junos OS creates these interfaces at system startup and uses these interfaces only for processing GRE and IP-IP tunnels.

At any given instant, a cluster node can be in one of the following states: hold, primary, secondary-hold, secondary, ineligible, or disabled. Multiple event types, such as interface monitoring, Services Processing Unit (SPU) monitoring, failures, and manual failovers, can trigger a state transition.

Enable Chassis Cluster Formation

You create two vSRX instances to form a chassis cluster, and then you set the cluster ID and node ID on each instance to join the cluster. When a vSRX instance joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

You can deploy up to 255 chassis clusters in a *Layer 2* domain. Clusters and nodes are identified in the following ways:

- The *cluster ID* (a number from 1 to 255) identifies the cluster.
- The *node ID* (a number from 0 to 1) identifies the cluster node.

Generally, on SRX Series devices, the cluster ID and node ID are written into EEPROM. On the vSRX instance, vSRX stores and reads the IDs from **boot/loader.conf** and uses the IDs to initialize the chassis cluster during startup.

Prerequisites

Ensure that your vSRX instances comply with the following prerequisites before you enable chassis clustering:

- You have committed a basic configuration to both vSRX instances that form the chassis cluster. See *Configure vSRX Using the CLI*.
- Use `show version` in Junos OS to ensure that both vSRX instances have the same software version.
- Use `show system license` in Junos OS to ensure that both vSRX instances have the same licenses installed.

You must set the same chassis cluster ID on each vSRX node and reboot the vSRX VM to enable chassis cluster formation.

1. In operational command mode, set the chassis cluster ID and node number on vSRX node 0.

```
user@vsrx0>set chassis cluster cluster-id number node 0 reboot
```

2. In operational command mode, set the chassis cluster ID and node number on vSRX node 1.

```
user@vsrx1>set chassis cluster cluster-id number node 1 reboot
```

NOTE: The vSRX interface naming and mapping to vNICs changes when you enable chassis clustering. See *Requirements for vSRX on KVM* for a summary of interface names and mappings for a pair of vSRX VMs in a cluster (node 0 and node 1).

Chassis Cluster Quick Setup with J-Web

To configure chassis cluster from *J-Web*:

1. Enter the vSRX node 0 interface IP address in a Web browser.
2. Enter the vSRX username and password, and click **Log In**. The J-Web dashboard appears.
3. Click **Configuration Wizards> Cluster (HA) Setup** from the left panel. The Chassis Cluster Setup Wizard appears. Follow the steps in the setup wizard to configure the cluster ID and the two nodes in the cluster, and to verify connectivity.

> **NOTE**: Use the built-in Help icon in J-Web for further details on the Chassis Cluster Setup wizard.

> **NOTE**: Navigate to **Configure>Device Settings>Cluster (HA) Setup** from Junos OS release 18.1 and later to configure the chassis cluster setup.

4. Configure the secondary node Node1 by selecting **Yes, this is the secondary unit to be setup (Node 1)** using radio button.
5. Click **Next**.
6. Specify the settings such as **Enter password**, **Re-enter password**, **Node 0 FXP0 IP**, and **Node 1 FXP0 IP** for secondary node access.
7. Click **Next**.
8. Select the secondary unit's Control Port and Fabric Port.
9. Click **Next**.
10. (Optional) Select **Save a backup file before proceeding with shutdown** using check box to re-configure it for chassis cluster.
11. Click **Next**.
12. Click **Shutdown and continue** to connect to other unit.
13. Click **Refresh Browser**.
14. Configure the primary node Node0 by selecting **No, this is the primary unit to be setup (Node 0)** to configure primary unit and establish a chassis cluster configuration.
15. Click **Next**.
16. Specify the settings such as **Enter password**, **Re-enter password**, **Node 0 FXP0 IP**, and **Node 1 FXP0 IP** for primary node access.
17. Click **Next** to restart the primary unit.
18. (Optional) Select **Save a backup file before proceeding with shutdown** to save a backup file of current settings before proceeding.
19. Click **Reboot and continue**. After completing the reboot, power on the secondary unit to establish the chassis cluster connection.
20. Login to the device console and add static route to get the J-Web access.
21. Login to the J-Web and click **Configuration Wizards> Cluster (HA) Setup** from the left panel. The Chassis Cluster Setup Wizard appears.
22. Click **Next** to get the primary unit connected.
23. Configure the basic settings **DHCP Client**, **IP address**, **Default gateway**, **Member interface Node 0**, **Member interface Node 1**.
24. Click **Next** to complete the chassis cluster configuration.

25. Click **Finish** to exit the wizard. You can access the primary node using J-Web.

Manually Configure a Chassis Cluster with J-Web

You can use the *J-Web* interface to configure the primary node 0 vSRX instance in the cluster. Once you have set the cluster and node IDs and rebooted each vSRX, the following configuration will automatically be synced to the secondary node 1 vSRX instance.

Select **Configure>Chassis Cluster>Cluster Configuration**. The Chassis Cluster configuration page appears.

> **NOTE**: Navigate to **Configure>Device Settings>Cluster (HA) Setup** from Junos OS release 18.1 and later to configure the HA cluster setup.

Table 15 on page 90 explains the contents of the HA Cluster Settings tab.

Table 16 on page 92 explains how to edit the Node Settings tab.

Table 17 on page 92 explains how to add or edit the HA Cluster Interfaces table.

Table 18 on page 94 explains how to add or edit the HA Cluster Redundancy Groups table.

Table 15: Chassis Cluster Configuration Page

Field	Function
Node Settings	
Node ID	Displays the node ID.
Cluster ID	Displays the cluster ID configured for the node.
Host Name	Displays the name of the node.
Backup Router	Displays the router used as a gateway while the Routing Engine is in secondary state for redundancy-group 0 in a chassis cluster.
Management Interface	Displays the management interface of the node.

Table 15: Chassis Cluster Configuration Page *(Continued)*

Field	Function
IP Address	Displays the management IP address of the node.
Status	Displays the state of the redundancy group. • **Primary**–Redundancy group is active. • **Secondary**–Redundancy group is passive.
Chassis Cluster>HA Cluster Settings>Interfaces	
Name	Displays the physical interface name.
Member Interfaces/IP Address	Displays the member interface name or IP address configured for an interface.
Redundancy Group	Displays the redundancy group.
Chassis Cluster>HA Cluster Settings>Redundancy Group	
Group	Displays the redundancy group identification number.
Preempt	Displays the selected preempt option. • **True**–Primary Role can be preempted based on priority. • **False**–Primary Role cannot be preempted based on priority.
Gratuitous ARP Count	Displays the number of gratuitous Address Resolution Protocol (*ARP*) requests that a newly elected primary device in a chassis cluster sends out to announce its presence to the other network devices.
Node Priority	Displays the assigned priority for the redundancy group on that node. The eligible node with the highest priority is elected as primary for the redundant group.

Table 16: Edit Node Setting Configuration Details

Field	Function	Action
Node Settings		
Host Name	Specifies the name of the host.	Enter the name of the host.
Backup Router	Displays the device used as a gateway while the Routing Engine is in the secondary state for redundancy-group 0 in a chassis cluster.	Enter the IP address of the backup router.
Destination		
IP	Adds the destination address.	Click **Add**.
Delete	Deletes the destination address.	Click **Delete**.
Interface		
Interface	Specifies the interfaces available for the router. **NOTE**: Allows you to add and edit two interfaces for each fabric link.	Select an option.
IP	Specifies the interface IP address.	Enter the interface IP address.
Add	Adds the interface.	Click **Add**.
Delete	Deletes the interface.	Click **Delete**.

Table 17: Add HA Cluster Interface Configuration Details

Field	Function	Action
Fabric Link > Fabric Link 0 (fab0)		

Table 17: Add HA Cluster Interface Configuration Details *(Continued)*

Field	Function	Action
Interface	Specifies fabric link 0.	Enter the interface IP fabric link 0.
Add	Adds fabric interface 0.	Click **Add**.
Delete	Deletes fabric interface 0.	Click **Delete**.
Fabric Link > Fabric Link 1 (fab1)		
Interface	Specifies fabric link 1.	Enter the interface IP for fabric link 1.
Add	Adds fabric interface 1.	Click **Add**.
Delete	Deletes fabric interface 1.	Click **Delete**.
Redundant Ethernet		
Interface	Specifies a logical interface consisting of two physical Ethernet interfaces, one on each chassis.	Enter the logical interface.
IP	Specifies a redundant Ethernet IP address.	Enter a redundant Ethernet IP address.
Redundancy Group	Specifies the redundancy group ID number in the chassis cluster.	Select a redundancy group from the list.
Add	Adds a redundant Ethernet IP address.	Click **Add**.
Delete	Deletes a redundant Ethernet IP address.	Click **Delete**.

Table 18: Add Redundancy Groups Configuration Details

Field	Function	Action
Redundancy Group	Specifies the redundancy group name.	Enter the redundancy group name.
Allow preemption of primaryship	Allows a node with a better priority to initiate a failover for a redundancy group. **NOTE**: By default, this feature is disabled. When disabled, a node with a better priority does not initiate a redundancy group failover (unless some other factor, such as faulty network connectivity identified for monitored interfaces, causes a failover).	–
Gratuitous ARP Count	Specifies the number of gratuitous Address Resolution Protocol requests that a newly elected primary sends out on the active redundant Ethernet interface child links to notify network devices of a change in primary role on the redundant Ethernet interface links.	Enter a value from 1 to 16. The default is 4.
node0 priority	Specifies the priority value of node0 for a redundancy group.	Enter the node priority number as 0.
node1 priority	Specifies the priority value of node1 for a redundancy group.	Select the node priority number as 1.
Interface Monitor		
Interface	Specifies the number of redundant Ethernet interfaces to be created for the cluster.	Select an interface from the list.
Weight	Specifies the weight for the interface to be monitored.	Enter a value from 1 to 125.
Add	Adds interfaces to be monitored by the redundancy group along with their respective weights.	Click **Add**.

Table 18: Add Redundancy Groups Configuration Details *(Continued)*

Field	Function	Action
Delete	Deletes interfaces to be monitored by the redundancy group along with their respective weights.	Select the interface from the configured list and click **Delete**.
IP Monitoring		
Weight	Specifies the global weight for IP monitoring.	Enter a value from 0 to 255.
Threshold	Specifies the global threshold for IP monitoring.	Enter a value from 0 to 255.
Retry Count	Specifies the number of retries needed to declare reachability failure.	Enter a value from 5 to 15.
Retry Interval	Specifies the time interval in seconds between retries.	Enter a value from 1 to 30.
IPV4 Addresses to Be Monitored		
IP	Specifies the IPv4 addresses to be monitored for reachability.	Enter the IPv4 addresses.
Weight	Specifies the weight for the redundancy group interface to be monitored.	Enter the weight.
Interface	Specifies the logical interface through which to monitor this IP address.	Enter the logical interface address.
Secondary IP address	Specifies the source address for monitoring packets on a secondary link.	Enter the secondary IP address.
Add	Adds the IPv4 address to be monitored.	Click **Add**.
Delete	Deletes the IPv4 address to be monitored.	Select the IPv4 address from the list and click **Delete**.

SEE ALSO

Chassis Cluster Feature Guide for Security Devices

vSRX Cluster Staging and Provisioning for KVM

IN THIS SECTION

You can provision the vSRX VMs and virtual networks to configure chassis clustering.

The staging and provisioning of the vSRX *chassis cluster* includes the following tasks:

Chassis Cluster Provisioning on vSRX

Chassis cluster requires the following direct connections between the two vSRX instances:

- Control link, or *virtual network*, which acts in active/passive mode for the control plane traffic between the two vSRX instances
- Fabric link, or virtual network, which acts in active/active mode for the data traffic between the two vSRX instances

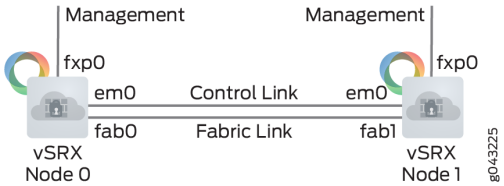
> **NOTE**: You can optionally create two fabric links for more redundancy.

The vSRX cluster uses the following interfaces:

- Out-of-band Management interface (fxp0)
- Cluster control interface (em0)
- Cluster fabric interface (fab0 on node0, fab1 on node1)

NOTE: The control interface must be the second *vNIC*. You can optionally configure a second fabric link for increased redundancy.

Figure 13: vSRX Chassis Cluster



vSRX supports chassis cluster using the virtio driver and interfaces, with the following considerations:

- When you enable chassis cluster, you must also enable jumbo frames (MTU size = 9000) to support the fabric link on the virtio network interface.
- If you configure a chassis cluster across two physical hosts, disable igmp-snooping on each host physical interface that the vSRX control link uses to ensure that the control link heartbeat is received by both nodes in the chassis cluster.

```
hostOS# echo 0 > /sys/devices/virtual/net/<bridge-name>/bridge/multicast_snooping
```

- After you enable chassis cluster, the vSRX instance maps the second vNIC to the control link, em0. You can map any other vNICs to the fabric link.

NOTE: For virtio interfaces, link status update is not supported. The link status of virtio interfaces is always reported as Up. For this reason, a vSRX instance using virtio and chassis cluster cannot receive link up and link down messages from virtio interfaces.

The virtual network MAC aging time determines the amount of time that an entry remains in the MAC table. We recommend that you reduce the MAC aging time on the virtual networks to minimize the downtime during failover.

For example, you can use the `brctl setageing bridge 1` command to set aging to 1 second for the Linux bridge.

You configure the virtual networks for the control and fabric links, then create and connect the control interface to the control virtual network and the fabric interface to the fabric virtual network.

Creating the Chassis Cluster Virtual Networks with virt-manager

In KVM, you create two virtual networks (control and fabric) to which you can connect each vSRX instance for chassis clustering.

To create a virtual network with `virt-manager`:

1. Launch `virt-manager` and select **Edit>Connection Details**. The Connection details dialog box appears.
2. Select **Virtual Networks**. The list of existing virtual networks appears.
3. Click **+** to create a new virtual network for the control link. The Create a new virtual network wizard appears.
4. Set the subnet for this virtual network and click **Forward**.
5. Select **Enable DHCP** and click **Forward**.
6. Select **Isolated virtual network** and click **forward**.
7. Verify the settings and click **Finish** to create the virtual network.

Creating the Chassis Cluster Virtual Networks with virsh

In KVM, you create two virtual networks (control and fabric) to which you can connect each vSRX for chassis clustering.

To create the control network with `virsh`:

1. Use the `virsh net-define` command on the *host* OS to create an XML file that defines the new virtual network. Include the XML fields described in Table 19 on page 98 to define this network.

> **NOTE**: See the official `virsh` documentation for a complete description of available options.

Table 19: virsh net-define XML Fields

Field	Description
<network>...</network>	Use this XML wrapper element to define a virtual network.
<name>*net-name*</name>	Specify the virtual network name.
<bridge name="*bridge-name*" />	Specify the name of the host bridge used for this virtual network.

Table 19: virsh net-define XML Fields *(Continued)*

Field	Description
<forward mode="*forward-option*" />	Specify routed or nat. Do not use the <forward> element for isolated mode.
<ip address="*ip-address*" netmask="*net-mask*" <dhcp range start="*start*" end="*end*" </dhcp> </ip>	Specify the IP address and subnet mask used by this virtual network, along with the DHCP address range.

The following example shows a sample XML file that defines a control virtual network.

```
<network>
 <name>control</name>
 <bridge name="controlvbr0" />
 <ip address="10.10.10.1" netmask="255.255.255.0" >
  <dhcp>
<range start="10.10.10.2" end="10.10.10.99" />
  </dhcp>
 </ip>
</network>
```

2. Use the `virsh net-start` command to start the new virtual network.

   hostOS# **virsh net-start control**

3. Use the `virsh net-autostart` command to automatically start the new virtual network when the host OS boots.

   hostOS# **virsh net-autostart control**

4. Optionally, use the `virsh net-list -all` command in the host OS to verify the new virtual network.

```
hostOS# # virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 control              active     yes           yes
 default              active     yes           yes
```

5. Repeat this procedure to create the fabric virtual network.

Configuring the Control and Fabric Interfaces with virt-manager

To configure the control and fabric interfaces for chassis clustering with `virt-manager`:

1. In `virt-manager`, double-click the vSRX VM and select **View>Details**. The vSRX Virtual Machine details dialog box appears.
2. Select the second *vNIC* and select the control *virtual network* from the Source device list.
3. Select **virtio** from the Device model list and click **Apply**.
4. Select a subsequent vNIC, and select the fabric virtual network from the Source device list.
5. Select **virtio** from the Device model list and click **Apply**.
6. For the fabric interface, use the `ifconfig` command on the host OS to set the MTU to 9000.

   hostOS# **ifconfig vnet1 mtu 9000**

Configuring the Control and Fabric Interfaces with virsh

To configure control and fabric interfaces to a vSRX VM with `virsh`:

1. Type **virsh attach-interface --domain *vsrx-vm-name* --type network --source *control-vnetwork* --target control --model virtio** on the host OS.

   This command creates a virtual interface called control and connects it to the control virtual network.
2. Type **virsh attach-interface --domain *vsrx-vm-name* --type network --source *fabric-vnetwork* --target fabric --model virtio** on the host OS.

   This command creates a virtual interface called fabric and connects it to the fabric virtual network.
3. For the fabric interface, use the `ifconfig` command on the host OS to set the MTU to 9000.

   hostOS# **ifconfig vnet1 mtu 9000**

Configuring Chassis Cluster Fabric Ports

After the chassis cluster is formed, you must configure the interfaces that make up the fabric (data) ports.

Ensure that you have configured the following:

- Set the chassis cluster IDs on both vSRX instances and rebooted the vSRX instances.
- Configured the control and fabric links.

1. On the vSRX node 0 console in configuration mode, configure the fabric (data) ports of the cluster that are used to pass real-time objects (RTOs). The configuration will be synchronized directly through the control port to vSRX node 1.

NOTE: A fabric port can be any unused revenue interface.

```
user@vsrx0# set interfaces fab0 fabric-options member-interfaces ge-0/0/0
user@vsrx0# set interfaces fab1 fabric-options member-interfaces ge-7/0/0
user@vsrx0# set chassis cluster reth-count 2
user@vsrx0# set chassis cluster redundancy-group 0 node 0 priority 100
user@vsrx0# set chassis cluster redundancy-group 0 node 1 priority 10
user@vsrx0# set chassis cluster redundancy-group 1 node 0 priority 100
user@vsrx0# set chassis cluster redundancy-group 1 node 1 priority 10
user@vsrx0# commit
```

2. Reboot vSRX node 0.

RELATED DOCUMENTATION

virt tools

Verify the Chassis Cluster Configuration

IN THIS SECTION

Purpose

Verify that the chassis cluster is operational after you set up the vSRX instances for clustering and set the cluster ID and the node ID.

Action

After reboot, the two nodes are reachable on interface fxp0 with SSH. If the configuration is operational, the `show chassis cluster status` command displays output similar to that shown in the following sample output.

vsrx> `show chassis cluster status`

```
Cluster ID: 1
Node                  Priority          Status    Preempt  Manual Monitor-failures

Redundancy group: 0 , Failover count: 1
    node0                   100         secondary      no       no     None
    node1                   10          primary        no       no     None

Redundancy group: 1 , Failover count: 2
    node0                   100         secondary      no       no     None
    node1                   10          primary        no       no     None
```

A cluster is healthy when both the primary and the secondary nodes are present and when both have a priority greater than 0.

2

PART

vSRX Deployment for VMware

CHAPTER 5

Overview

IN THIS CHAPTER

Understand vSRX with VMware

IN THIS SECTION

This section presents an overview of vSRX on VMware

vSRX Overview

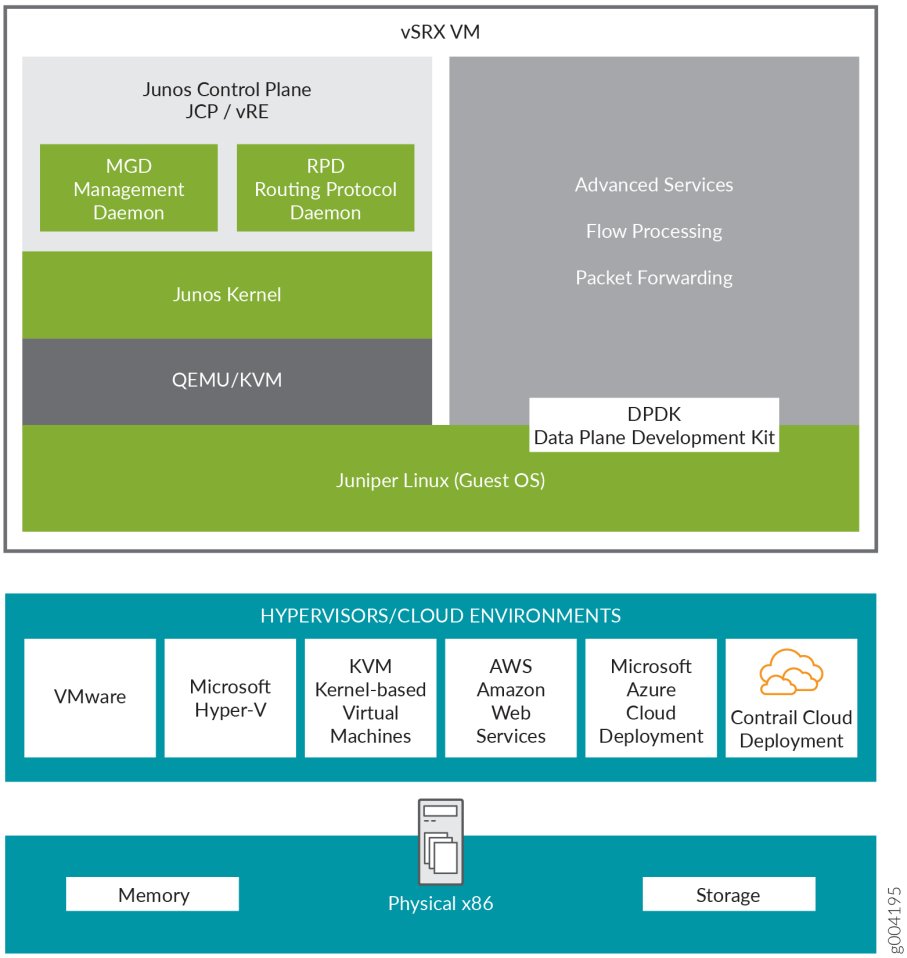
vSRX is a virtual security appliance that provides security and networking services at the perimeter or edge in virtualized private or public *cloud* environments. vSRX runs as a virtual machine (*VM*) on a standard x86 server. vSRX is built on the Junos operating system (Junos OS) and delivers networking and security features similar to those available on the software releases for the SRX Series Services Gateways.

The vSRX provides you with a complete Next-Generation Firewall (NGFW) solution, including core firewall, VPN, NAT, advanced Layer 4 through Layer 7 security services such as Application Security, intrusion detection and prevention (IPS), and UTM features including Enhanced Web Filtering and Anti-

Virus. Combined with Sky ATP, the vSRX offers a cloud-based advanced anti-malware service with dynamic analysis to protect against sophisticated malware, and provides built-in machine learning to improve verdict efficacy and decrease time to remediation.

Figure 14 on page 105 shows the high-level architecture.

Figure 14: vSRX Architecture



vSRX includes the Junos control plane (JCP) and the packet forwarding engine (PFE) components that make up the data plane. vSRX uses one virtual CPU (vCPU) for the JCP and at least one vCPU for the PFE. Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, multi-core vSRX supports scaling vCPUs and GB virtual RAM (vRAM). Additional vCPUs are applied to the data plane to increase performance.

Junos OS Release 18.4R1 supports a new software architecture vSRX 3.0 that removes dual OS and nested virtualization requirement of existing vSRX architecture.

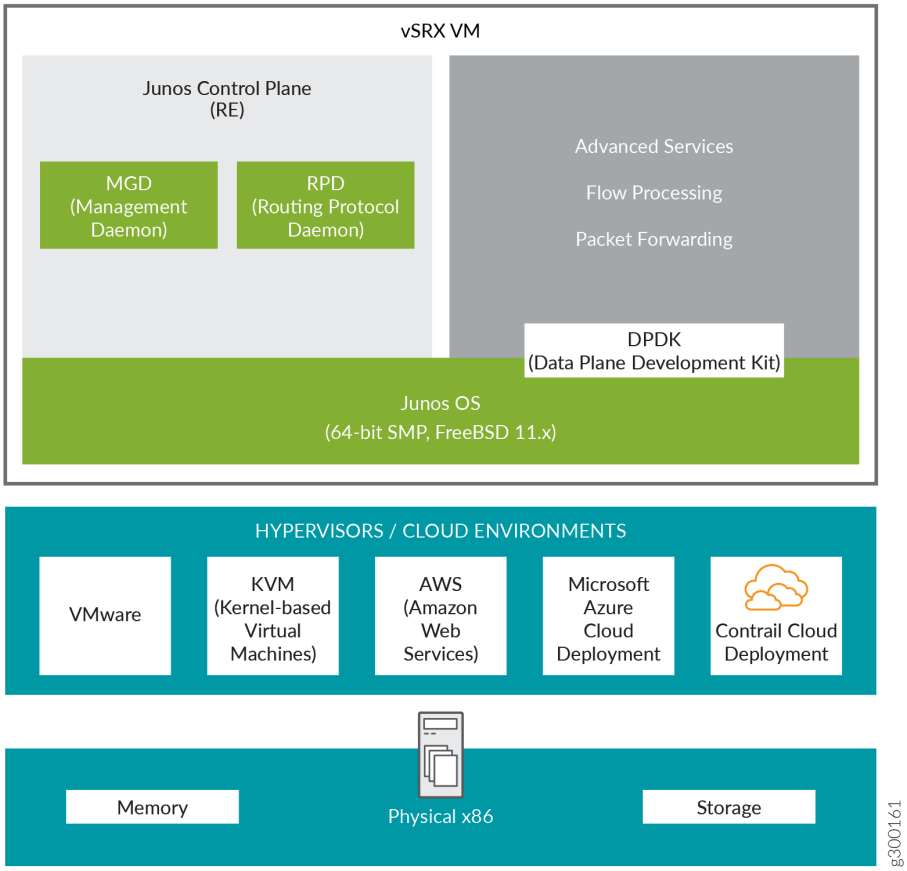
In vSRX 3.0 architecture, FreeBSD 11.x is used as the guest OS and the Routing Engine and Packet Forwarding Engine runs on FreeBSD 11.x as single virtual machine for improved performance and scalability. vSRX 3.0 uses DPDK to process the data packets in the data plane. A direct Junos upgrade from vSRX to vSRX 3.0 software is not supported.

vSRX 3.0 has the following enhancements compared to vSRX:

- Removed the restriction of requiring nested VM support in hypervisors.
- Removed the restriction of requiring ports connected to control plane to have Promiscuous mode enabled.
- Improved boot time and enhanced responsiveness of the control plane during management operations.
- Improved live migration.

Figure 15 on page 107 shows the high-level software architecture for vSRX 3.0

Figure 15: vSRX 3.0 Architecture



vSRX Benefits and Use Cases

vSRX on standard x86 servers enables you to quickly introduce new services, deliver customized services to customers, and scale security services based on dynamic needs. vSRX is ideal for public, private, and hybrid cloud environments.

Some of the key benefits of vSRX in a virtualized private or public cloud multitenant environment include:

- *Stateful firewall* protection at the tenant edge
- Faster deployment of virtual firewalls into new sites
- Ability to run on top of various hypervisors and public cloud infrastructures

- Full routing, *VPN*, core security, and networking capabilities
- Application security features (including IPS and App-Secure)
- Content security features (including Anti Virus, Web Filtering, Anti Spam, and Content Filtering)
- Centralized management with Junos Space Security Director and local management with J-Web Interface
- Juniper Networks Sky Advanced Threat Prevention (Sky ATP) integration

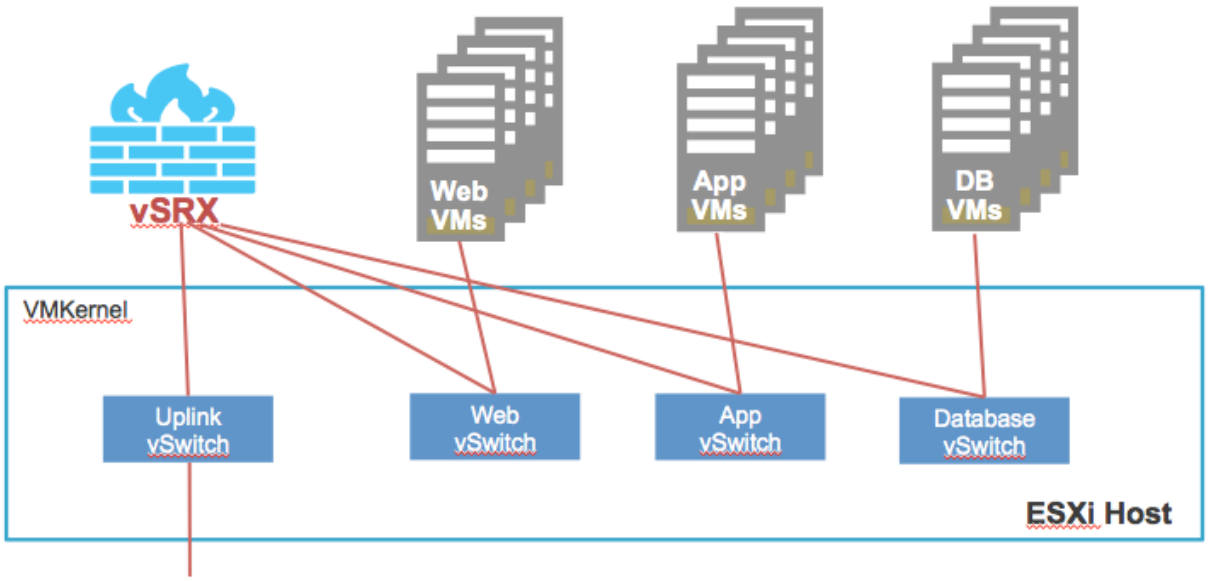
vSRX on VMWare ESXi deployment

VMware vSphere is a virtualization environment for systems supporting the x86 architecture. VMware ESXi® is the hypervisor used to create and run virtual machines (VMs) and virtual appliances on a host machine. The VMware vCenter Server® is a service that manages the resources of multiple ESXi hosts.

The VMware vSphere Web Client is used to deploy the vSRX VM.

Figure 16 on page 108 shows an example of how vSRX can be deployed to provide security for applications running on one or more virtual machines. The vSRX virtual switch has a connection to a physical adapter (the uplink) so that all application traffic flows through the vSRX VM to the external network.

Figure 16: Example of vSRX Deployment



vSRX Scale Up Performance

Table 20 on page 109 shows the vSRX scale up performance based on the number of vCPUs and vRAM applied to a vSRX VM. The table outlines the Junos OS release in which a particular software specification for deploying vSRX on VMware was introduced. You will need to download a specific Junos OS release to take advantage of certain scale up performance features.

Table 20: vSRX Scale Up Performance

vCPUs	vRAM	NICs	Junos OS Release Introduced
2 vCPUs	4 GB	• SR-IOV (Intel 82599, X520/X540) • VMNET3	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
5 vCPUs	8 GB	• SR-IOV (Intel 82599, X520/X540) • VMNET3	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1
9 vCPUs	16 GB	• SR-IOV (Mellanox ConnectX-3/ConnectX-3 Pro and Mellanox ConnectX-4 EN/ConnectX-4 Lx EN) **NOTE**: SR-IOV (Mellanox ConnectX-3/ConnectX-3 Pro and Mellanox ConnectX-4 EN/ConnectX-4 Lx EN) is required if you intend to scale the performance and capacity of a vSRX to 9 vCPUs and 16 GB vRAM.	Junos OS Release 18.4R1
17 vCPUs	32 GB	• SR-IOV (Mellanox ConnectX-3/ConnectX-3 Pro and Mellanox ConnectX-4 EN/ConnectX-4 Lx EN) **NOTE**: SR-IOV (Mellanox ConnectX-3/ConnectX-3 Pro and Mellanox ConnectX-4 EN/ConnectX-4 Lx EN) is required if you intend to scale the performance and capacity of a vSRX to 17 vCPUs and 32 GB vRAM.	Junos OS Release 18.4R1
1 vCPU	4 GB	SR-IOV on the Mellanox ConnectX-3 and ConnectX-4 family adapters.	Junos OS Release 21.2R1
4 vCPUs	8 GB	SR-IOV on the Mellanox ConnectX-3 and ConnectX-4 family adapters.	Junos OS Release 21.2R1

Table 20: vSRX Scale Up Performance *(Continued)*

vCPUs	vRAM	NICs	Junos OS Release Introduced
8 vCPUs	16GB	SR-IOV on the Mellanox ConnectX-3 and ConnectX-4 family adapters.	Junos OS Release 21.2R1
16 vCPUs	32 GB	SR-IOV on the Mellanox ConnectX-3 and ConnectX-4 family adapters.	Junos OS Release 21.2R1

You can scale the performance and capacity of a vSRX instance by increasing the number of vCPUs and the amount of vRAM allocated to the vSRX. The multi-core vSRX automatically selects the appropriate vCPUs and vRAM values at boot time, as well as the number of Receive Side Scaling (RSS) queues in the NIC. If the vCPU and vRAM settings allocated to a vSRX VM do not match what is currently available, the vSRX scales down to the closest supported value for the instance. For example, if a vSRX VM has 3 vCPUs and 8 GB of vRAM, vSRX boots to the smaller vCPU size, which requires a minimum of 2 vCPUs. You can scale up a vSRX instance to a higher number of vCPUs and amount of vRAM, but you cannot scale down an existing vSRX instance to a smaller setting.

NOTE: The number of RSS queues typically matches with the number of data plane vCPUs of a vSRX instance. For example, a vSRX with 4 data plane vCPUs should have 4 RSS queues.

vSRX Session Capacity Increase

vSRX solution is optimized to increase the session numbers by increasing the memory.

With the ability to increase the session numbers by increasing the memory, you can enable vSRX to:

- Provide highly scalable, flexible and high-performance security at strategic locations in the mobile network.
- Deliver the performance that service providers require to scale and protect their networks.

Run the `show security flow session summary | grep maximum` command to view the maximum number of sessions.

Starting in Junos OS Release 18.4R1, the number of flow sessions supported on a vSRX instance is increased based on the vRAM size used.

Starting in Junos OS Release 19.2R1, the number of flow sessions supported on a vSRX 3.0 instance is increased based on the vRAM size used.

Table 21 on page 111 lists the flow session capacity.

Table 21: vSRX and vSRX 3.0 Flow Session Capacity Details

vCPUs	Memory	Flow Session Capacity
2	4 GB	0.5 M
2	6 GB	1 M
2/5	8 GB	2 M
2/5	10 GB	2 M
2/5	12 GB	2.5 M
2/5	14 GB	3 M
2/5/9	16 GB	4 M
2/5/9	20 GB	6 M
2/5/9	24 GB	8 M
2/5/9	28 GB	10 M
2/5/9/17	32 GB	12 M
2/5/9/17	40 GB	16 M
2/5/9/17	48 GB	20 M
2/5/9/17	56 GB	24 M

Table 21: vSRX and vSRX 3.0 Flow Session Capacity Details *(Continued)*

vCPUs	Memory	Flow Session Capacity
2/5/9/17	64 GB	28 M

Release History Table

Release	Description
19.2R1	Starting in Junos OS Release 19.2R1, the number of flow sessions supported on a vSRX 3.0 instance is increased based on the vRAM size used.
18.4R1	Starting in Junos OS Release 18.4R1, the number of flow sessions supported on a vSRX instance is increased based on the vRAM size used.
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, multi-core vSRX supports scaling vCPUs and GB virtual RAM (vRAM). Additional vCPUs are applied to the data plane to increase performance.

RELATED DOCUMENTATION

- VMware vSphere
- RSS: Receive Side Scaling

Requirements for vSRX on VMware

IN THIS SECTION

Software Specifications

Table 22 on page 113 lists the system software requirement specifications when deploying vSRX on VMware. The table outlines the Junos OS release in which a particular software specification for deploying vSRX on VMware was introduced. You must need to download a specific Junos OS release to take advantage of certain features.

Table 22: Specifications for vSRX and vSRX 3.0 on VMware

Component	Specification	Junos OS Release Introduced
Hypervisor support	VMware ESXi 5.1, 5.5, or 6.0	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
	VMware ESXi 6.5	Junos OS Release 18.4R1
	VMware ESXi 6.5 and 6.7 (vSRX 3.0 only)	Junos OS Release 19.3R1
	VMware ESXi 7.0 (For vSRX 3.0 only)	Junos OS Release 20.1R2
Memory	4 GB	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
	8GB	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1
	16 GB	Junos OS Release 18.4R1
	32 GB	Junos OS Release 18.4R1
Disk space	16 GB (IDE or SCSI drives)	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1
vCPUs	2 vCPUs	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1

Table 22: Specifications for vSRX and vSRX 3.0 on VMware *(Continued)*

Component	Specification	Junos OS Release Introduced
	5 vCPUs	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1
	9 vCPUs	Junos OS Release 18.4R1
	17 vCPUs	Junos OS Release 18.4R1
vNICs	Up to 10 vNICs • SR-IOV **NOTE**: We recommend the Intel X520/X540/X710 10 G physical NICs for SR-IOV support on vSRX. For SR-IOV limitations, see the *Known Behavior* section of the *vSRX Release Notes*. • VMNET3 **NOTE**: The Intel DPDK drivers use polling mode for all vNICs, so the NAPI and interrupt mode features in VMXNET3 are not currently supported. **NOTE**: Starting in Junos OS Release 15.1X49-D20, in vSRX deployments using VMware ESX, changing the default speed (1000 Mbps) or the default link mode (full duplex) is not supported on VMXNET3 vNICs.	Junos OS Release 15.1X49-D15 and Junos OS Release 17.3R1

Table 22: Specifications for vSRX and vSRX 3.0 on VMware *(Continued)*

Component	Specification	Junos OS Release Introduced
	Starting in Junos OS Release 18.4R1: • SR-IOV (Mellanox ConnectX-3/ ConnectX-3 Pro and Mellanox ConnectX-4 EN/ConnectX-4 Lx EN) is required if you intend to scale the performance and capacity of a vSRX VM to 9 or 17 vCPUs and 16 or 32 GB vRAM. **NOTE**: Mellanox NIC (any ConnectX) cards are not support on VMWare. • The DPDK version has been upgraded from 17.02 to 17.11.2 to support the Mellanox Family Adapters.	Junos OS Release 18.4R1
	Starting in Junos OS Release 19.4R1, DPDK version 18.11 is supported on vSRX. With this feature the Mellanox Connect Network Interface Card (NIC) on vSRX now supports OSPF Multicast and VLANs.	Junos OS Release 19.4R1

Table 23 on page 115 lists the specifications on the vSRX 3.0 virtual machine (VM).

Table 23: Specifications for vSRX 3.0 on VMware

vCPU	vRAM	DPDK	Hugepage	vNICs	vDisk	Junos OS Release Introduced
2	4G	17.05	2G	2-10	20G	Junos OS Release 18.2R1

Table 23: Specifications for vSRX 3.0 on VMware *(Continued)*

vCPU	vRAM	DPDK	Hugepage	vNICs	vDisk	Junos OS Release Introduced
5	8G	17.05	6G	2–10 vSRX on VMWare supports VMXNET3 through DPDK and PMD, and SR-IOV (82599). A maximum number of eight interfaces are supported. DPDK uses HugePage for improved performance.	20G	Junos OS Release 18.4R1
9 or 17 vCPUs	32G	18.11		With single-root I/O virtualization over Intel X710/XL710 for improved scalability and performance.		Starting in Junos OS Release 19.1R1

Hardware Specifications

Table 24 on page 116 lists the hardware specifications for the host machine that runs the vSRX VM.

Table 24: Hardware Specifications for the Host Machine

Component	Specification
Host processor type	Intel x86_64 multicore CPU **NOTE**: DPDK requires Intel Virtualization VT-x/VT-d support in the CPU. See *About Intel Virtualization Technology.*
Virtual network adapter	VMXNet3 device or VMware Virtual NIC **NOTE**: Virtual Machine Communication Interface (VMCI) communication channel is internal to the ESXi hypervisor and the vSRX VM.

Table 24: Hardware Specifications for the Host Machine *(Continued)*

Component	Specification
Physical NIC support on vSRX 3.0	Support SR-IOV on X710/XL710/XXV710 vSRX3.0 SR-IOV HA on I40E (X710,X740,X722 and so on) are not supported on VMware. Starting in Junos OS Release 21.2R1, a vSRX 3.0 instance deployed on VMware supports SR-IOV on the Mellanox ConnectX-4 and ConnectX-5 family adapters. For a summary of vSRX sizes (number of vCPU and amount of vRAM) that support the Mellanox ConnectX-4 and ConnectX-5 Family Adapters, see vSRX Scale Up Performance Chassis Cluster is not supported with SR-IOV interface adapters.

Best Practices for Improving vSRX Performance

Review the following practices to improve vSRX performance.

NUMA Nodes

The x86 server architecture consists of multiple sockets and multiple cores within a socket. Each socket also has memory that is used to store packets during I/O transfers from the NIC to the host. To efficiently read packets from memory, guest applications and associated peripherals (such as the NIC) should reside within a single socket. A penalty is associated with spanning CPU sockets for memory accesses, which might result in nondeterministic performance. For vSRX, we recommend that all vCPUs for the vSRX VM are in the same physical non-uniform memory access (NUMA) node for optimal performance.

CAUTION: The Packet Forwarding Engine (PFE) on the vSRX will become unresponsive if the NUMA nodes topology is configured in the hypervisor to spread the instance's vCPUs across multiple host NUMA nodes. vSRX requires that you ensure that all vCPUs reside on the same NUMA node.

We recommend that you bind the vSRX instance with a specific NUMA node by setting NUMA node affinity. NUMA node affinity constrains the vSRX VM resource scheduling to only the specified NUMA node.

PCI NIC-to-VM Mapping

If the node on which vSRX is running is different from the node to which the Intel PCI NIC is connected, then packets will have to traverse an additional hop in the QPI link, and this will reduce overall

throughput. Use the `esxtop` command to view information about relative physical NIC locations. On some servers where this information is not available, refer to the hardware documentation for the slot-to-NUMA node topology.

Interface Mapping for vSRX on VMware

Each network adapter defined for a vSRX is mapped to a specific interface, depending on whether the vSRX instance is a standalone VM or one of a cluster pair for high availability. The interface names and mappings in vSRX are shown in Table 25 on page 118 and Table 26 on page 119.

Note the following:

- In standalone mode:
  - fxp0 is the out-of-band management interface.
  - ge-0/0/0 is the first traffic (revenue) interface.
- In cluster mode:
  - fxp0 is the out-of-band management interface.
  - em0 is the cluster control link for both nodes.
  - Any of the traffic interfaces can be specified as the fabric links, such as ge-0/0/0 for fab0 on node 0 and ge-7/0/0 for fab1 on node 1.

Table 25 on page 118 shows the interface names and mappings for a standalone vSRX VM.

Table 25: Interface Names for a Standalone vSRX VM

Network Adapter	Interface Name in Junos OS
1	fxp0
2	ge-0/0/0
3	ge-0/0/1
4	ge-0/0/2
5	ge-0/0/3

Table 25: Interface Names for a Standalone vSRX VM *(Continued)*

Network Adapter	Interface Name in Junos OS
6	ge-0/0/4
7	ge-0/0/5
8	ge-0/0/6

Table 26 on page 119 shows the interface names and mappings for a pair of vSRX VMs in a cluster (node 0 and node 1).

Table 26: Interface Names for a vSRX Cluster Pair

Network Adapter	Interface Name in Junos OS
1	fxp0 (node 0 and 1)
2	em0 (node 0 and 1)
3	ge-0/0/0 (node 0) ge-7/0/0 (node 1)
4	ge-0/0/1 (node 0) ge-7/0/1 (node 1)
5	ge-0/0/2 (node 0) ge-7/0/2 (node 1)
6	ge-0/0/3 (node 0) ge-7/0/3 (node 1)
7	ge-0/0/4 (node 0) ge-7/0/4 (node 1)

Table 26: Interface Names for a vSRX Cluster Pair *(Continued)*

Network Adapter	Interface Name in Junos OS
8	ge-0/0/5 (node 0) ge-7/0/5 (node 1)

vSRX Default Settings on VMware

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.

NOTE: For the management interface, fxp0, VMware uses the VMXNET 3 vNIC and requires promiscuous mode on the vSwitch.

Table 27 on page 120 lists the factory default settings for the vSRX security policies.

Table 27: Factory Default Settings for Security Policies

Source Zone	Destination Zone	Policy Action
trust	untrust	permit
trust	trust	permit
untrust	trust	deny

RELATED DOCUMENTATION

About Intel Virtualization Technology

DPDK Release Notes

CHAPTER 6

Install vSRX in VMware

IN THIS CHAPTER

Install vSRX with VMware vSphere Web Client

The following procedure describes how to install vSRX and connect vSRX interfaces to the virtual switches for the appropriate applications. Only the vSRX virtual switch has a connection to a physical adapter (the uplink) so that all application traffic flows through the vSRX VM to the external network.

To install vSRX with the VMware vSphere Web Client:

> **NOTE**: To upgrade an existing vSRX instance, see *Migration, Upgrade, and Downgrade* in the *vSRX Release Notes*.

1. Download the vSRX software package for VMware from the Juniper Networks website.

   > **NOTE**: Do not change the filename of the downloaded software image or the installation will fail.

2. Validate the vSRX .ova file if required. For more information, see *Validate the vSRX .ova File for VMware*.

3. Enter the vCenter server hostname or address in your browser (https://<*ipaddress*>:9443) to access the vSphere Web Client, and log in to the vCenter server with your credentials.

4. Select a host or other valid parent for a virtual machine and click **Actions > All vCenter Actions > Deploy OVF Template**.

NOTE: The Client Integration Plug-in must be installed before you can deploy OVF templates (see your VMware documentation).

5. Click **Browse** to locate the vSRX software package, and then click **Next**.
6. Click **Next** in the OVF Template Details window.
7. Click **Accept** in the End User License Agreement window, and then click **Next**.
8. Change the default vSRX VM name in the Name box and click **Next**. It is advisable to keep this name the same as the hostname you intend to give to the VM.
9. In the Datastore window, do not change the default settings for:
   - Datastore
   - Available Space

   Table 28 on page 122 lists the disk formats available to store the virtual disk. You can choose one of the three options listed.

   NOTE: For detailed information on the disk formats, see Virtual Disk Provisioning.

   Table 28: Disk Formats for Virtual Disk Storage

Disk Format	Description
Thick Provision Lazy Zeroed	Allocates disk space to the virtual disk without erasing the previously stored data. The previous data is erased when the VM is used for the first time.
Thick Provision Eager Zeroed	Erases the previously stored data completely and then allocates the disk space to the virtual disk. Creation of disks in this format is time consuming.
Thin Provision	Allocates only as much datastore space as the disk needs for its initial operations. Use this format to save storage space.

10. Select a datastore to store the configuration file and virtual disk files in OVF template, and then click **Next**.

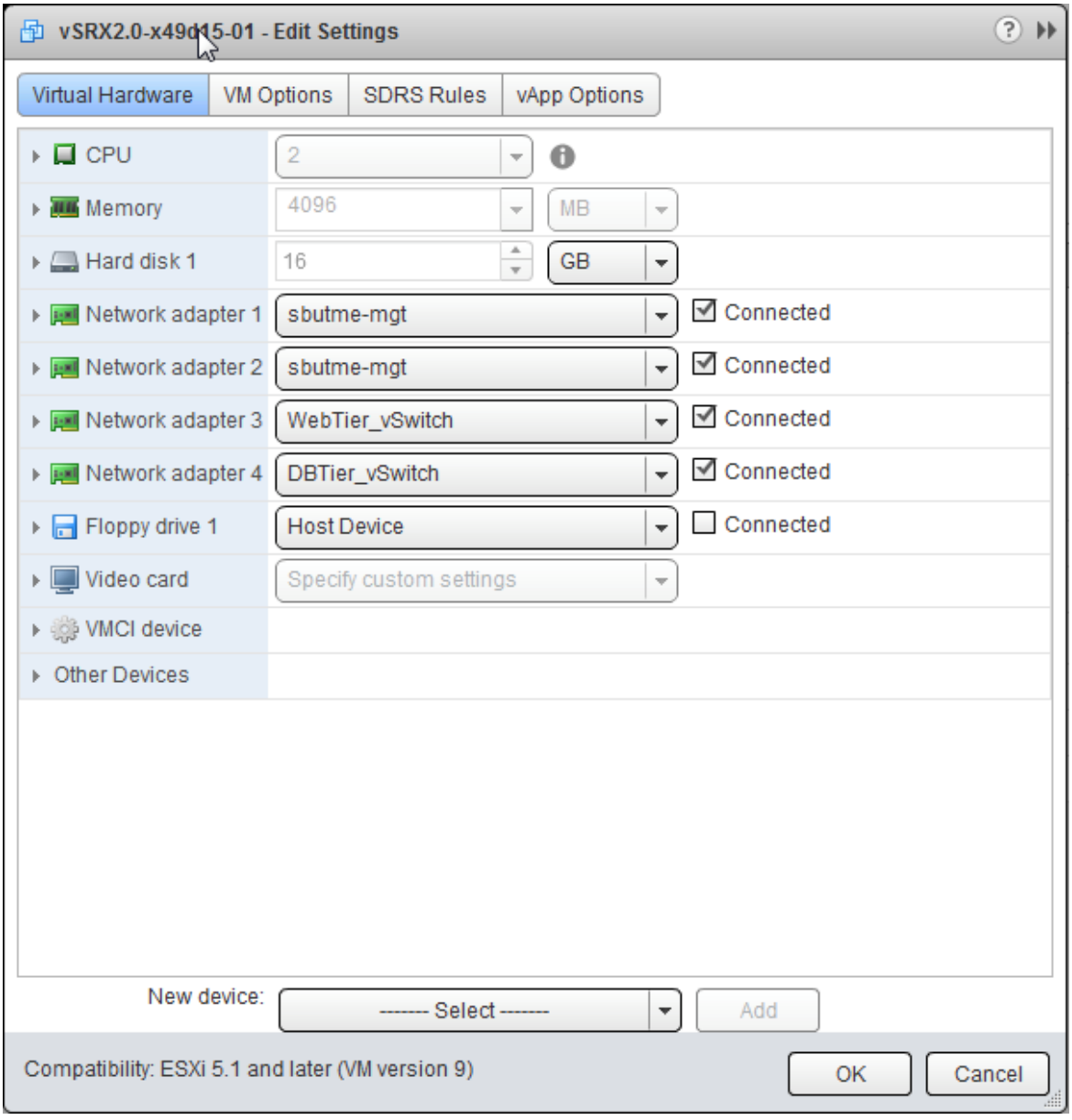
11. Select your management network from the list, and then click **Next**. The management network is assigned to the first network adapter, which is reserved for the management interface (fxp0).

12. Click **Finish** to complete the installation.

13. Open the Edit Settings page of the vSRX VM and select a virtual switch for each network adapter. Three network adapters are created by default. Network adapter 1 is for the management network (fxp0). To add a fourth adapter, select **Network** from New device list at the bottom of the page. To add more adapters, see *Add vSRX Interfaces*.

In Figure 17 on page 124, network adapter 2 uses the management network for the uplink to the external network.

Figure 17: vSRX Edit Settings Page



14. Enable promiscuous mode for the management virtual switch:

    a. Select the host where the vSRX VM is installed, and select **Manage > Networking > Virtual switches**.

b. In the list of virtual switches, select vSwitch0 to view the topology diagram for the management network connected to network adapter 1.

c. Click the **Edit** icon at the top of the list, select **Security**, and select **Accept** next to Promiscuous mode. Click **OK**.

> **NOTE**: vSwitch1 corresponds to network adapter 2, vSwitch2 corresponds to network adapter 3, and so on.

15. Enable hardware-assisted virtualization to optimize performance of the vSRX Routing Engine that runs in a nested VM:

a. Power off the vSRX VM.

b. Right-click on the vSRX VM and select **Edit Settings**.

c. On the Virtual Hardware tab, expand CPU, select **Expose hardware-assisted virtualization to guest OS**, and click **OK**.

On the Manage tab, select **Settings > VM Hardware** and expand CPU to verify that the **Hardware virtualization** option is shown as Enabled.

> **NOTE**: The default vSRX VM login ID is root with no password. By default, vSRX is assigned a DHCP-based IP address if a DHCP server is available on the network.

RELATED DOCUMENTATION

- Using Virtual NUMA
- Virtual Machine vCPU and vNUMA Rightsizing

Load an Initial Configuration on a vSRX with VMware

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, you can use a mounted ISO image to pass the initial startup Junos OS configuration to a vSRX VM. This ISO image contains a file in the root directory called juniper.conf. The configuration file uses curly brackets ({) and indentation to display the hierarchical structure of the configuration. Terminating or leaf statements in the configuration hierarchy are displayed with a trailing semicolon (;) to define configuration details, such as root password, management IP address, default gateway, and other configuration statements.

NOTE: The juniper.conf file must be in the format same as displayed using `show configuration` command and it cannot be in `set` command format.

The process to bootstrap a vSRX VM with an ISO configuration image is as follows:

1. Create the **juniper.conf** configuration file with your Junos OS configuration.

   An example of a juniper.conf file follows.

```
system {
    host-name iso-mount-test;
    root-authentication {
        encrypted-password "$5$wCXP/Ma4$aqMJBhy82.wI643ijb73yHKKl9TXApPycGKKn.PjpA8"; ##
SECRET-DATA
    }
    login {
        user regress {
            uid 2001;
            class super-user;
            authentication {
                encrypted-password "$6$FGJM2YEb
$KTGIehvNt9Mf.u3ESWGB1aSQeXrSeg6zoCWZw0D6M6vnmWb8DAWsprNXyKZeW6M3kErFFTFtAuNpGjDjfwX4t."; ##
SECRET-DATA
            }
        }
    }
    services {
    ssh {
```

```
            root-login allow;
        }
    telnet;
        web-management {
            http {
                interface fxp0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
}
security {
    forwarding-options {
        family {
            inet6 {
                mode flow-based;
            }
        }
    }
    policies {
        default-policy {
            permit-all;
        }
    }
    zones {
        security-zone AAA {
            interfaces {
```

```
                all;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        vlan-tagging;
        unit 0 {
            vlan-id 77;
            family inet {
                address 10.1.1.0/24 {
                    arp 10.1.1.10 mac 00:10:12:34:12:34;
                }
            }

        }
    }
    ge-0/0/1 {
        vlan-tagging;
        unit 0 {
            vlan-id 1177;
            family inet {
                address 10.1.1.1/24 {
                    arp 10.1.1.10 mac 00:10:22:34:22:34;
                }
            }

        }
    }
   fxp0 {
        unit 0 {
            family inet {
                address 192.168.70.9/19;
            }
        }
    }

}
routing-options {
    static {
        route 0.0.0.0/0 next-hop 192.168.64.1;
    }
```

```
}
```

2. Create an ISO image that includes the **juniper.conf** file.
3. Mount the ISO image to the vSRX VM.
4. Boot or reboot the vSRX VM. vSRX will boot using the **juniper.conf** file included in the mounted ISO image.
5. Unmount the ISO image from the vSRX VM. To unmount the ISO image see Dismount ISO Image from VM.

NOTE: If you do not unmount the ISO image after the initial boot or reboot, all subsequent configuration changes to the vSRX are overwritten by the ISO image on the next reboot.

Create a vSRX Bootstrap ISO Image

This task uses a Linux system to create the ISO image.

To create a vSRX bootstrap ISO image:

1. Create a configuration file in plaintext with the Junos OS command syntax and save in a file called **juniper.conf**.
2. Create a new directory.

```
hostOS$ mkdir iso_dir
```

3. Copy **juniper.conf** to the new ISO directory.

```
hostOS$ cp juniper.conf iso_dir
```

NOTE: The **juniper.conf** file must contain the full vSRX configuration. The ISO bootstrap process overwrites any existing vSRX configuration.

4. Use the Linux `mkisofs` command to create the ISO image.

```
hostOS$ mkisofs -l -o test.iso iso_dir
```

```
I: -input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
175 extents written (0 MB)
```

NOTE: The `-l` option allows for a long filename.

SEE ALSO

Linux mkisofs command

Upload an ISO Image to a VMWare Datastore

To upload an ISO image to a datastore:

1. On the VMware vSphere Web Client, select the datastore you want to upload the file to.
2. Select the folder where you want to store the file and click **Upload a File** from the task bar.
3. Browse to the file on your local computer and click **Upload**.

Optionally, refresh the datastore to see the new file.

Provision vSRX with an ISO Bootstrap Image on VMWare

To provision a vSRX VM with an ISO bootstrap image:

1. From VMware vSphere client, select the host server where the vSRX VM resides.
2. Right-click the vSRX VM and select **Edit Settings**. The Edit Setting dialogue box appears.
3. Select the Hardware tab and click **Add**. The Add Hardware dialog box opens.
4. Select the CD/DVD drive and click **Next**.
5. Select **Use ISO image** and click **Next**.
6. Click **Datastore ISO File**, browse to your boostrap ISO image, and click **Next**.

7. Click **Next** and **Finish** to save this setting.
8. Click **OK** to save this CD drive to the VM.
9. Right-click the vSRX VM and select **Power>Power On** to boot the vSRX VM.
10. After the vSRX boots, verify the configuration and then select **Power> Power down** to shut down the vSRX so you can remove the ISO image.
11. Select the CD/DVD drive from the Hardware tab in the VMWare vSphere client.
12. Select the CD drive for the ISO file and click **Remove** to remove your boostrap ISO image.
13. Click **OK** to save this setting.
14. Right-click the vSRX VM and select **Power>Power On** to boot the vSRX VM.

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, you can use a mounted ISO image to pass the initial startup Junos OS configuration to a vSRX VM. This ISO image contains a file in the root directory called juniper.conf. The configuration file uses curly brackets ({) and indentation to display the hierarchical structure of the configuration. Terminating or leaf statements in the configuration hierarchy are displayed with a trailing semicolon (;) to define configuration details, such as root password, management IP address, default gateway, and other configuration statements.

RELATED DOCUMENTATION

- Linux mkisofs command

Validate the vSRX .ova File for VMware

The vSRX open virtual application (OVA) image is securely signed. You can validate the OVA image, if necessary, but you can install or upgrade vSRX without validating the OVA image.

Before you validate the OVA image, ensure that the Linux/UNIX PC or Windows PC on which you are performing the validation has the following utilities available: tar, openssl, and ovftool. See the OVF Tool Documentation for details about the VMware Open Virtualization Format (OVF) tool, including a Software Download link.

To validate the OVA image on a Linux machine:

1. Download the vSRX OVA image and the Juniper Networks Root certificate file (**JuniperRootRSACA.pem**) from the vSRX Juniper Networks Software Download page.

NOTE: You need to download the Juniper Networks Root certificate file only once; you can use the same file to validate OVA images for future releases of vSRX.

2. (Optional) If you downloaded the OVA image and the certificate file to a PC running Windows, copy the two files to a temporary directory on a PC running Linux or UNIX. You can also copy the OVA image and the certificate file to a temporary directory (**/var/tmp** or **/tmp**) on a vSRX node.

   Ensure that the OVA image file and the Juniper Networks Root certificate file are not modified during the validation procedure. You can do this by providing write access to these files only to the user performing the validation procedure. This is especially important if you use an accessible temporary directory, such as **/tmp** or **/var/tmp**, because such directories can be accessed by several users. Take precautions to ensure that the files are not modified by other users during the validation procedure.

3. Navigate to the directory containing the OVA image.

   ```
   -bash-4.1$ ls
   ```

   ```
   JuniperRootCA.pem  junos-vsrx-15.1X49-DXX.4-domestic.ova
   ```

4. Unpack the OVA image by running the following command: **tar xf *ova-filename***

   where *ova-filename* is the filename of the previously downloaded OVA image.

   ```
   -bash-4.1$ mkdir tmp
   ```

   ```
   -bash-4.1$ cd tmp
   ```

   ```
   -bash-4.1$ tar xf ../junos-vsrx-15.1X49-DXX.4-domestic.ova
   ```

5. Verify that the unpacked OVA image contains a certificate chain file (**certchain.pem**) and a signature file (**vsrx.cert**).

   ```
   -bash-4.1$ ls
   ```

   ```
   certchain.pem  junos-vsrx-15.1X49-DXX.4-domestic.cert  junos-vsrx-15.1X49-DXX.4-domestic-
   disk1.vmdk  junos-vsrx-15.1X49-DXX.4-domestic.mf  junos-vsrx-15.1X49-DXX.4-domestic.ovf
   ```

6. Validate the unpacked OVF file (extension .ovf) by running the following command: **ovftool ovf-filename**

   where *ovf-filename* is the filename of the unpacked OVF file contained within the previously downloaded OVA image.

-bash-4.1$ **/usr/lib/vmware-ovftool/ovftool junos-vsrx-15.1X49-DXX.4-domestic.ovf**

```
OVF version:   1.0
VirtualApp:    false
Name:          vSRX
Version:       JUNOS 15.1
Vendor:        Juniper Networks Inc.
Product URL:
               https://www.juniper.net/us/en/products-services/software/security/vsrxseries/
Vendor URL:    https://www.juniper.net/
Download Size:  227.29 MB

Deployment Sizes:
  Flat disks:   2.00 GB
  Sparse disks: 265.25 MB

Networks:
  Name:        VM Network
  Description: The VM Network network

Virtual Machines:
  Name:               Juniper Virtual SRX
  Operating System:   freebsdguest
  Virtual Hardware:
    Families:         vmx-07
    Number of CPUs:   2
    Cores per socket: 1
    Memory:           2.00 GB

    Disks:
      Index:          0
      Instance ID:    5
      Capacity:       2.00 GB
      Disk Types:     IDE

    NICs:
      Adapter Type:   E1000
      Connection:     VM Network

      Adapter Type:   E1000
      Connection:     VM Network
```

```
Deployment Options:
  Id:          2GvRAM
  Label:       2G vRAM
  Description:
                          2G Memory
```

7. Validate the signing certificate with the Juniper Networks Root CA file by running the following command:

   openssl verify -CAfile JuniperRootRSACA.pem -untrusted *Certificate-Chain-File Signature-file*

   where **JuniperRootRSACA.pem** is the Juniper Networks Root CA file, *Certificate-Chain-File* is the filename of the unpacked certificate chain file (extension **.pem**) and *Signature-file* is the filename of the unpacked signature file (extension **.cert)**.

   `-bash-4.1$` **openssl verify -CAfile ../JuniperRootCA.pem -untrusted certchain.pem junos-vsrx-15.1X49-DXX.4-domestic.cert**

   ```
   junos-vsrx-15.1X49-DXX.4-domestic.cert: OK
   ```

8. (Optional) If you encounter validation issues with the OVA image:

   a. Determine if the contents of the OVA image have been modified. If the contents have been modified, download the OVA image from the vSRX downloads page.

   b. Determine whether the Juniper Networks Root CA file is corrupted or modified. If it was corrupted or modified, download the certificate file from the vSRX downloads page.

   c. Retry the preceding validation steps using one or both new files.

CHAPTER 7

vSRX VM Management with VMware

IN THIS CHAPTER

Add vSRX Interfaces

IN THIS SECTION

The network adapter for each interface uses SR-IOV or VMXNET 3 as the adapter type. The first network adapter is for the management interface (fxp0) and must use VMXNET 3. All additional network adapters should have the same adapter type. The three network adapters created by default use VMXNET 3.

NOTE: Starting in Junos OS Release 18.4R1:

- SR-IOV (Mellanox ConnectX-3/ConnectX-3 Pro and Mellanox ConnectX-4 EN/ConnectX-4 Lx EN) is required if you intend to scale the performance and capacity of a vSRX VM to 9 or 17 vCPUs and 16 or 32 GB vRAM.
- The DPDK version has been upgraded from 17.02 to 17.11.2 to support the Mellanox Family Adapters .

Starting in Junos OS Release 19.4R1, DPDK version 18.11 is supported on vSRX. With this feature the Mellanox Connect Network Interface Card (NIC) on vSRX now supports OSPF Multicast and VLANs.

The network adapters are mapped sequentially to the vSRX interfaces, as shown in *Requirements for vSRX on VMware*.

NOTE: If you have used the interface mapping workaround required for prior Junos releases, you do not need to make any changes when you upgrade to Junos Release 15.1X49-D70 for vSRX.

The following procedures describe how to add more network adapters:

Add SR-IOV Interfaces

SR-IOV interfaces must be added as PCI devices on VMware. To add an SR-IOV interface as a PCI Device, you must first select an available Virtual Function (VF) on the device.

Use the following procedure to locate available VFs and add PCI devices:

1. To locate one or more VFs:
   a. Use SSH to log in to the ESXi server and enter the following command to view the VFs for vmnic6 (or another vNIC):

```
# esxcli network sriovnic vf list -n vmnic6
```

```
VF ID  Active  PCI Address  Owner World ID
-----  ------  -----------  --------------
    0    true  005:16.0     982641
    1    true  005:16.2     982641
    2    true  005:16.4     982641
    3   false  005:16.6      -
    4   false  005:17.0      -
    5   false  005:17.2      -
    6   false  005:17.4      -
```

   Choose one or more VF IDs that are not active, such as 3 through 6. Note that a VF assigned to a VM that is powered off is shown as inactive.

b. Enter the `lspci` command to view the VF number of the chosen VF IDs. In the following example, find the entry that ends with [vmnic6], scroll down to the next entry ending in VF_3, and note the associated VF number 05:10.6. Note that the next VF_3 entry is for vmnic7.

```
# lspci
```

```
0000:05:00.0 Network controller: Intel Corporation 82599EB 10-Gig ... [vmnic6]
0000:05:00.1 Network controller: Intel Corporation 82599EB 10-Gig ... [vmnic7]
0000:05:10.0 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.0_VF_0]
0000:05:10.1 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.1_VF_0]
0000:05:10.2 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.0_VF_1]
0000:05:10.3 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.1_VF_1]
0000:05:10.4 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.0_VF_2]
0000:05:10.5 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.1_VF_2]
0000:05:10.6 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.0_VF_3] ----- VF ID 3 on vmnic6, with VF number 05:10.6.
0000:05:10.7 Network controller: Intel Corporation 82599 Ethernet Controller Virtual
Function [PF_0.5.1_VF_3] ----- VF ID 3 on vmnic7.
```

2. To add SR-IOV interfaces to the vSRX VM:

   a. Power off the vSRX VM and open the Edit Settings page. By default there are three network adapters using VMXNET 3.

   b. Add one or more PCI devices on the Virtual Hardware page. For each device, you must select an entry with an available VF number from Step 1. For example:

      05:10.6 | Intel Corporation 82599 Ethernet Controller Virtual Function

   c. Click **OK** and open the Edit Settings page to verify that the new network adaptors are shown on the Virtual Hardware page (one VMXNET 3 network adapter and up to nine SR-IOV interfaces as PCI devices).

      To view the SR-IOV interface MAC addresses, select the **VM Options** tab, click **Advanced** in the left frame, and then click **Edit Configuration**. In the parameters `pciPassthruN.generatedMACAddress`, N indicates the PCI device number (0 through 9).

d. Power on the vSRX VM and log in to the VM to verify that VMXNET 3 network adapter 1 is mapped to fxp0, PCI device 0 is mapped to ge-0/0/0, PCI device 1 is mapped to ge-0/0/1, and so on.

NOTE: A vSRX VM with SR-IOV interfaces cannot be cloned. You must deploy a new vSRX VM and add the SR-IOV interfaces as described here.

Add VMXNET 3 Interfaces

Use the following procedure to add VMXNET 3 interfaces:

1. Power off the vSRX VM and open the Edit Settings page on vSphere Web Client.
2. Add network adapters on the Virtual Hardware page. For each network adapter, select **Network** from New device list at the bottom of the page, expand **New Network**, and select **VMXNET 3** as the adapter type.
3. Click **OK** and open the Edit Settings page to verify that the new network adaptors are shown on the Virtual Hardware page.
4. Power on the vSRX VM and log in to the VM to verify that network adapter 1 is mapped to fxp0, network adapter 2 is mapped to ge-0/0/0, and so on. Use the `show interfaces terse` CLI command to verify that the fxp0 and ge-0/0/n interfaces are up.

Upgrade a Multicore vSRX with VMware

IN THIS SECTION

Starting in Junos OS Release 15.1X49-70 and Junos OS Release 17.3R1, you can scale the performance and capacity of a vSRX instance by increasing the number of vCPUs and the amount of vRAM allocated to the vSRX. See *Requirements for vSRX on VMware* for the software requirement specifications of a vSRX VM.

NOTE: You cannot scale down the number of vCPUs or decrease the amount of vRAM for an existing vSRX VM.

Power Down vSRX VM with VMware vSphere Web Client

In situations where you want to modify the vSRX VM XML file, you need to completely shut down vSRX and the associated VM.

To gracefully shutdown the vSRX instance with VMware vSphere Web Client:

1. Enter the vCenter server hostname or address in your browser (https://<ipaddress>:9443) to access the vSphere Web Client, and log in to the vCenter server with your credentials.
2. Check the vSRX VM you want to power off.
3. Select **Open Console** to open a console window to the vSRX VM.
4. From the vSRX console, reboot the vSRX instance.

   `vsrx#` **request system power-off**.

Upgrade a Multicore vSRX with VMware vSphere Web Client

You must power down the vSRX VM before you can update the vCPU and vRAM values for the VM.

To scale up the vSRX VM to a higher number of vCPUs or to an increased amount of vRAM:

1. On VMware vSphere Web Client, Select **Edit Settings** to open the powered down vSRX VM to open the virtual machine details window.
2. Select **Memory** and set the vRAM to the desired size.
3. Select **Processor** and set the number of vCPUs. Click **OK**.
4. Click **Power On**. The VM manager launches the vSRX VM with the new vCPU and vRAM settings.

NOTE: vSRX scales down to the closest supported value if the vCPU or vRAM settings do not match what is currently available.

Optimize Performance of vSRX

To optimize performance of vSRX on VMware:

1. For memory, select the NUMA node that line cards connect to.
2. For the CPU:

   a. Disable hyper-threading.

b. Select CPUs on the selected NUMA node.

c. Select n sockets and each socket has one core.

d. Reserve the CPU resource.

3. For the TX thread:

   - Configure a separate ESXi transmit thread per vNIC.
   - Place transmit threads on the same NUMA node.

4. For vNICs, use either 2 vNICs or 4 vNICs if you want to scale the performance of the vSRX VM.

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-70 and Junos OS Release 17.3R1, you can scale the performance and capacity of a vSRX instance by increasing the number of vCPUs and the amount of vRAM allocated to the vSRX.

Automate the Initialization of vSRX 3.0 Instances on VMware Hypervisor using VMware Tools

IN THIS SECTION

Overview

IN THIS SECTION

Open VM Tools is a set of services and modules that enhances the performance and user experience of vSRX. With this service, several features in VMware products are enabled for better management and easy user interactions with the guest OS. It includes kernel modules for enhancing the performance of virtual machines running Linux or other VMware–supported Unix–like guest operating systems. vSRX 3.0 supports VMware tools starting from Junos OS Release 20.2R1.

VMware Tools includes these components:

- VMware Tools Service
- VMware device Drivers
- VMware user process
- VMware Tools Control Panel

vSRX 3.0 runs on FreeBSD 11.x and later. FreeBSD 12 supports VMware open-vm-tools-10.3.0.

The VMware tools (binaries and libraries) are packaged into the vSRX image file and allow VM instances to query information from hypervisor and then set or use such information. by the VM instance itself.

During VM instance booting time, the boot-up script will look for Open Virtualization Format (OVF) settings or the machine ID setting. If the OVF settings are enabled, then the related VM CLI configurations are configured and the VM instance will use this CLI configuration when the VM instance is first powered on. We support autoconfiguration of hostname, IP address, gateway, DHCP, and DHCP server.

Benefits of VMware Tools for Autoconfiguration

- Execute VMware-provided or user configured scripts in guest OS during various power operations.
- Collect network, disk, and memory usage information from the guest periodically.
- Generate heartbeat from guests to hosts to determine guests' availability.
- Enable Time synchronization between a host and guest
- Allows File transfer between a host and guest
- Provides improved memory management and network performance
- Supports general mechanisms and protocols for communication between host and guests and from guest to guest
- Allows you to customize guest operating systems immediately after powering on virtual machines.

Provision VMware Tools for Autoconfiguration

There are 3 methods to make VMware tools support setting key-value are:

- Set the VM options of parameter machine ID for each key.
- Set vApp options of OVF property for each key.
- Edit the *.ova package file to add the property for each key.

Use one of the methods to set the key-value.

If you want to change any VM parameters, use the VMware GUI. When VMWare hypervisor powers on the VM instance, Open VMTool source code provides the functionality for the VM instance to query parameters from the hypervisor.

To set the VM options of parameter machine ID for settings keys:

1. On the VMware ESXi vCenter server, access the VM on vSphere Web client (FLEX or HTML5), go to **Edit Virtual Machine Setting** ->**VM Options**->**Advanced**, and then on the **Configuration Parameters** tab, click **Edit Configuration**.
2. On the **Configuration Parameters** page, add a new parameter with **Name** and **Value** for each key.

   > **NOTE**: For fxp0 IP address configuration, you can configure a key-value pair with a set of IP address or gateway, a set of DHCP address or DHCP server, or both. When you set both DHCP has higher priority over IP address.

3. Add the parameter by selecting **Add** and then click **OK**.
4. Verify the configurations by validating the configurations on the instance, verify the configuration of fxp0 and default routes using the `show interfaces terse fxp0` command, or by checking the log files at **/var/log/setup_config.log**. Log files at **/var/log/setup_config.log** provide you the debugging messages, any syntax error, IP validation, the CLI configuration, and so on.

To set the vApp options of OVF property for each key:

1. On the VMware ESXi vCenter server, access the VM on vSphere Web client (FLEX), go to **Edit Virtual Machine Setting** ->**vApp Options**->**OVF setting**, and under **OVF environment transparent** tab , select **VMWare Tools**.
2. Go to **Edit Virtual Machine Setting**->**vApp Options**->**Properties** and edit each key value.
3. To verify the configuration login and power-on for the first time as root and without password, verify the fxp0 and DHCP bindings or check the log files at **/var/log/vmware_ovf.info** and **/var/log/setup_config.log**.

To edit the OVF package file instructions:

1. Untar the *.ova. in the *.ova file. There are three files: *.ovf,*.mf, and *.vmdk.
2. Edit the *.ovf file to add some property for each key value under the production section.
3. To verify the configuration, deploy the vSRX 3.0 from vCenter server Web client and check the properties set for each key value or check the log files at **/var/log/vmware_ovf.info** and **/var/log/setup_config.log**.

CHAPTER 8

Configure vSRX Chassis Clusters in VMware

IN THIS CHAPTER

Configure a vSRX Chassis Cluster in Junos OS

IN THIS SECTION

Chassis Cluster Overview

Prerequisites

Ensure that your vSRX instances comply with the following prerequisites before you enable chassis clustering:

- Use `show version` in Junos OS to ensure that both vSRX instances have the same software version.
- Use `show system license` in Junos OS to ensure that both vSRX instances have the same licenses installed.

Chassis cluster groups a pair of the same kind of vSRX instances into a cluster to provide network node redundancy. The devices must be running the same Junos OS release. You connect the control virtual

interfaces on the respective nodes to form a *control plane* that synchronizes the configuration and Junos OS kernel state. The control link (a *virtual network* or *vSwitch*) facilitates the redundancy of interfaces and services. Similarly, you connect the *data plane* on the respective nodes over the fabric virtual interfaces to form a unified data plane. The fabric link (a virtual network or vSwitch) allows for the management of cross-node flow processing and for the management of session redundancy.

The control plane software operates in active/passive mode. When configured as a chassis cluster, one node acts as the primary device and the other as the secondary device to ensure stateful failover of processes and services in the event of a system or hardware failure on the primary device. If the primary device fails, the secondary device takes over processing of control plane traffic.

NOTE: If you configure a chassis cluster on vSRX nodes across two physical hosts, disable igmp-snooping on the bridge that each host physical interface belongs to that the control vNICs use. This ensures that the control link heartbeat is received by both nodes in the chassis cluster.

The chassis cluster data plane operates in active/active mode. In a chassis cluster, the data plane updates session information as traffic traverses either device, and it transmits information between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, traffic can enter the cluster on one node and exit from the other node.

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple *Packet Forwarding Engine*s. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.
- Support for generic routing encapsulation (*GRE*) and IP-over-IP (IP-IP) tunnels used to route encapsulated IPv4 or *IPv6* traffic by means of two internal interfaces, gr-0/0/0 and ip-0/0/0, respectively. Junos OS creates these interfaces at system startup and uses these interfaces only for processing GRE and IP-IP tunnels.

At any given instant, a cluster node can be in one of the following states: hold, primary, secondary-hold, secondary, ineligible, or disabled. Multiple event types, such as interface monitoring, Services Processing Unit (SPU) monitoring, failures, and manual failovers, can trigger a state transition.

Enable Chassis Cluster Formation

You create two vSRX instances to form a chassis cluster, and then you set the cluster ID and node ID on each instance to join the cluster. When a vSRX VM joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

You can deploy up to 255 chassis clusters in a *Layer 2* domain. Clusters and nodes are identified in the following ways:

- The *cluster ID* (a number from 1 to 255) identifies the cluster.
- The *node ID* (a number from 0 to 1) identifies the cluster node.

On SRX Series devices, the cluster ID and node ID are written into EEPROM. On the vSRX VM, vSRX stores and reads the IDs from **boot/loader.conf** and uses the IDs to initialize the chassis cluster during startup.

The chassis cluster formation commands for node 0 and node 1 are as follows:

- On vSRX node 0:

```
user@vsrx0>set chassis cluster cluster-id number node 0 reboot
```

- On vSRX node 1:

```
user@vsrx1>set chassis cluster cluster-id number node 1 reboot
```

The vSRX interface naming and mapping to vNICs changes when you enable chassis clustering. Use the same cluster ID number for each node in the cluster.

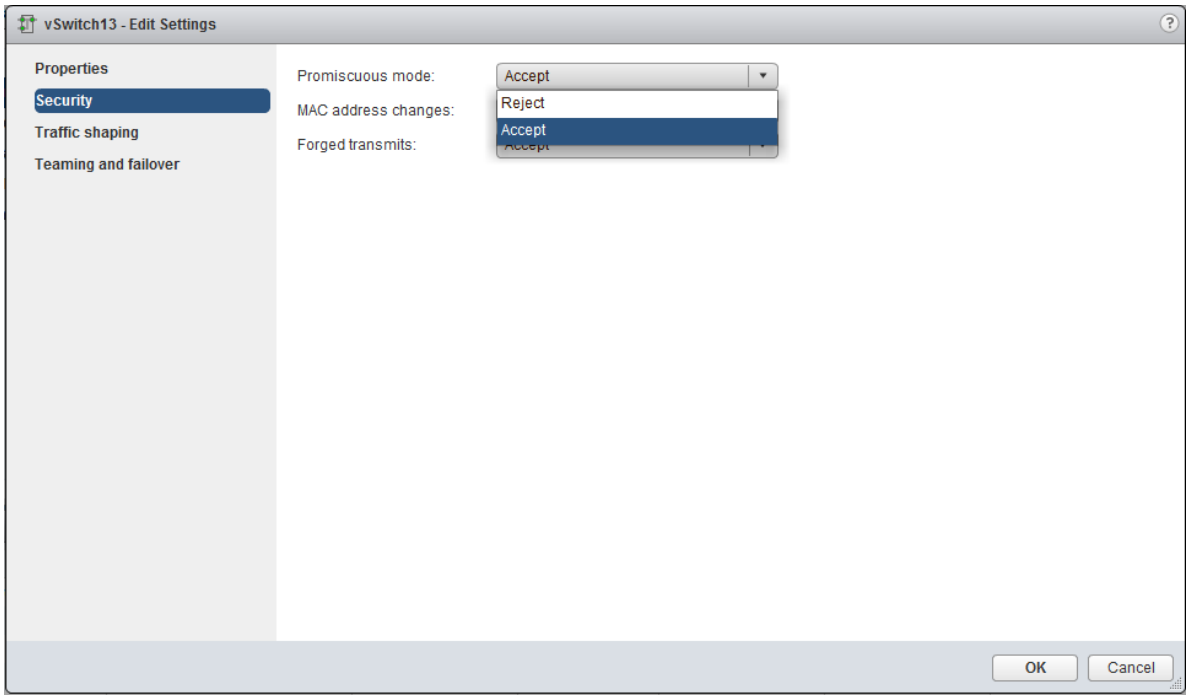
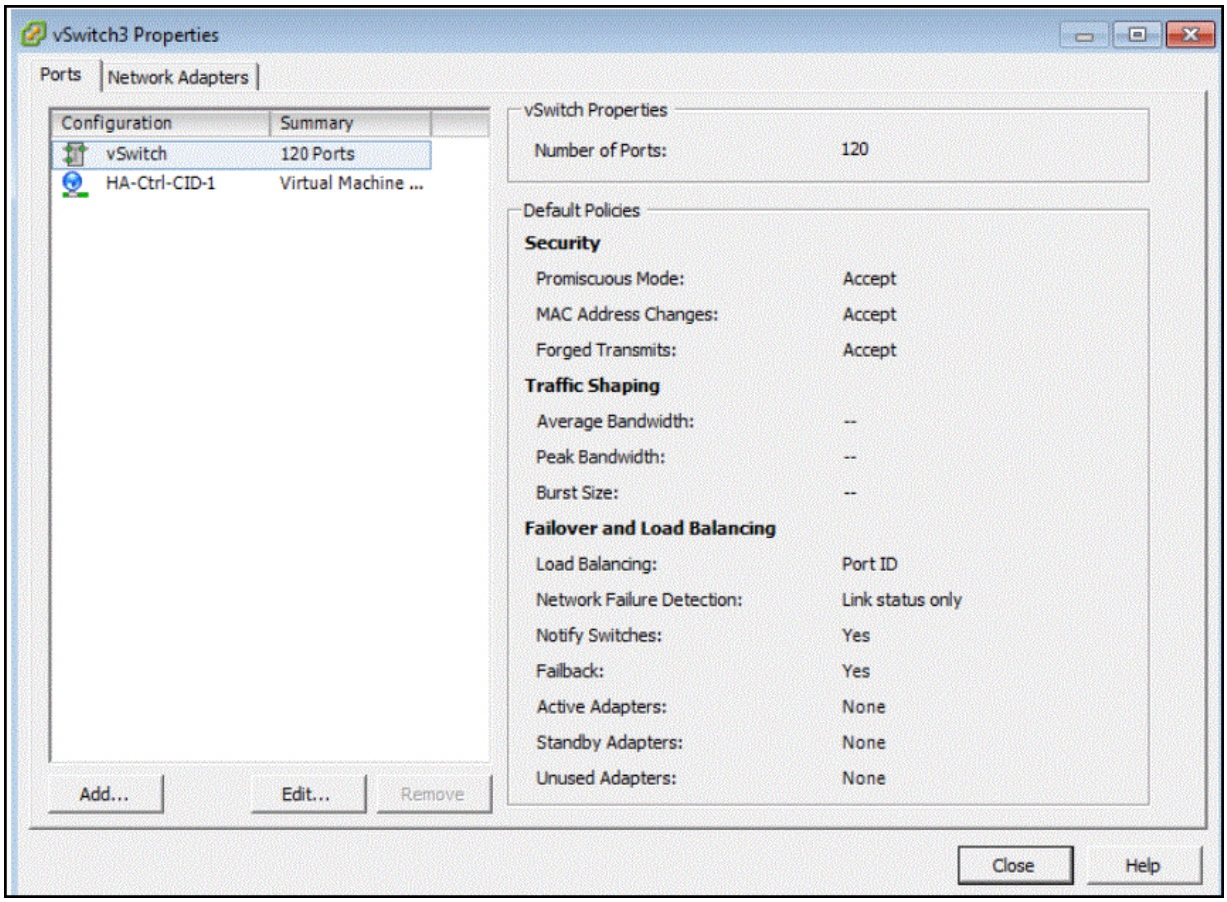
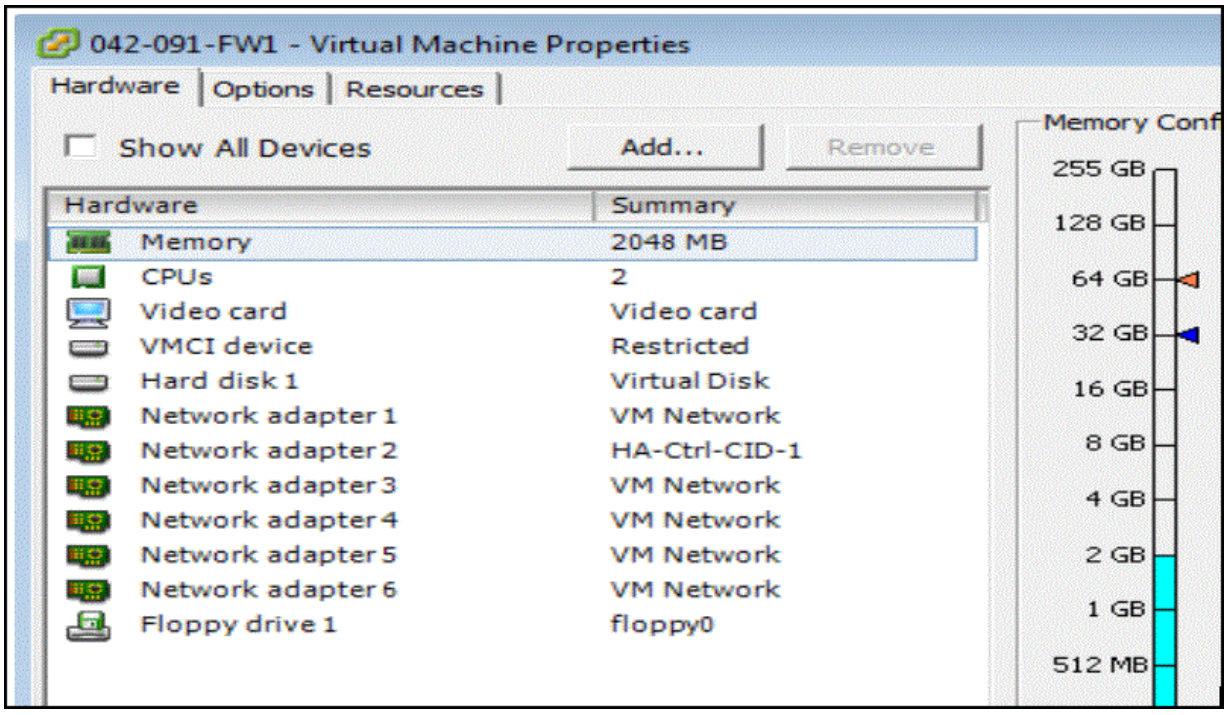
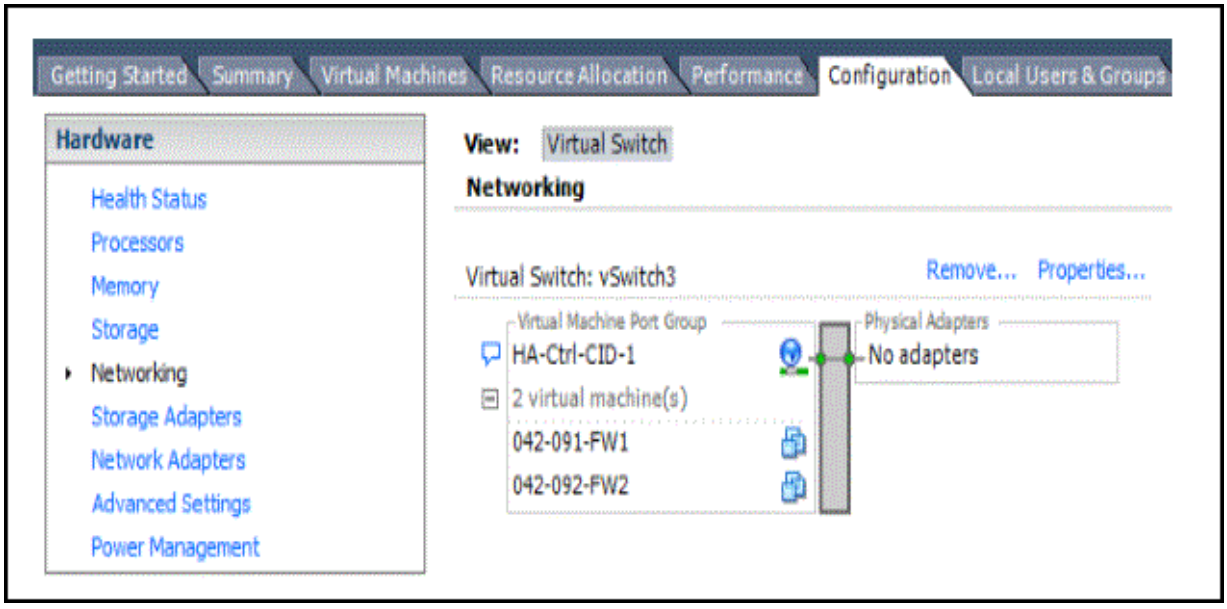
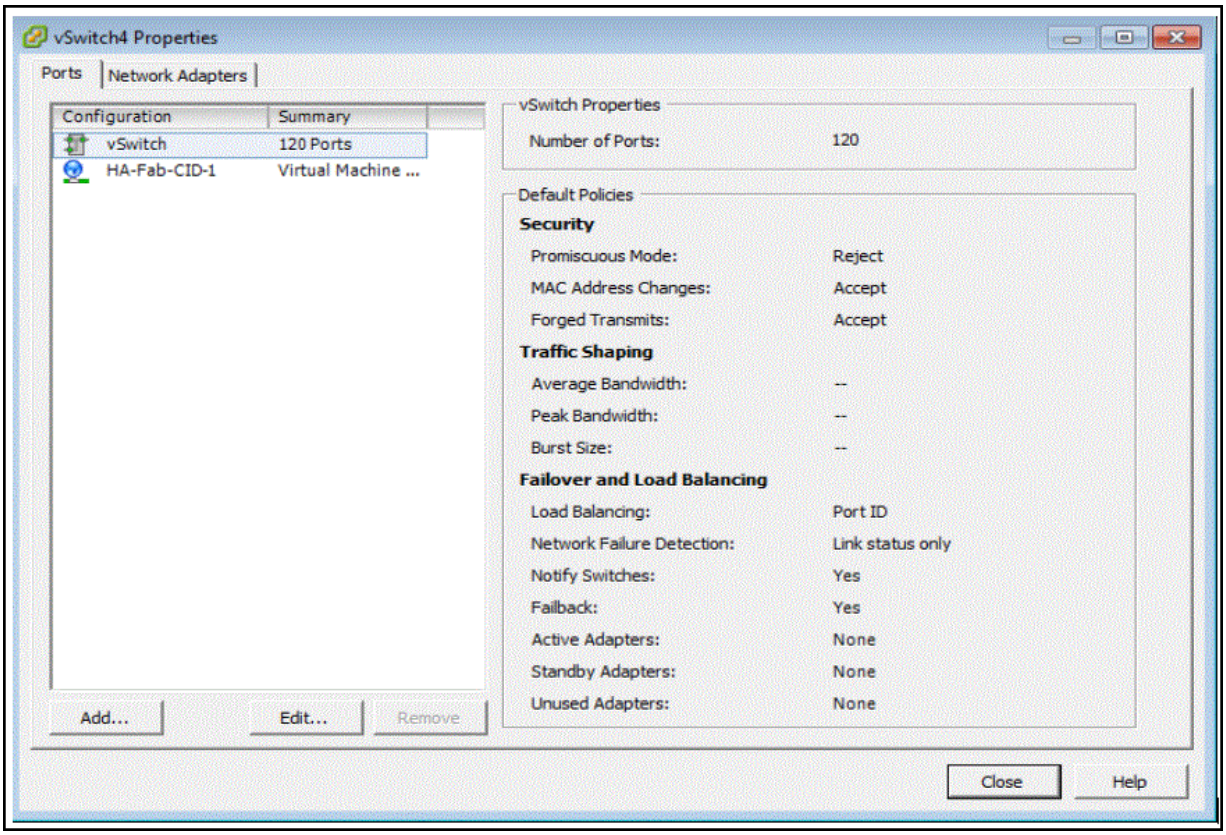
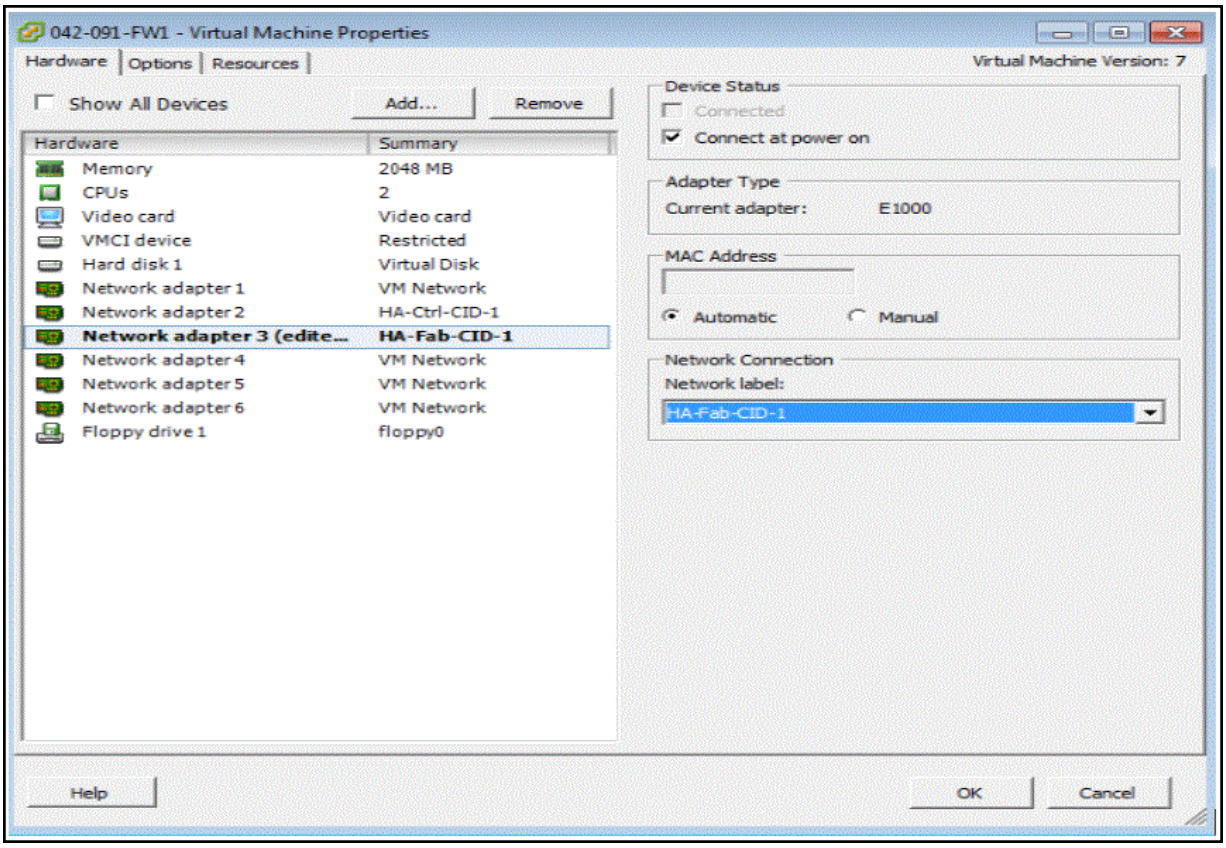
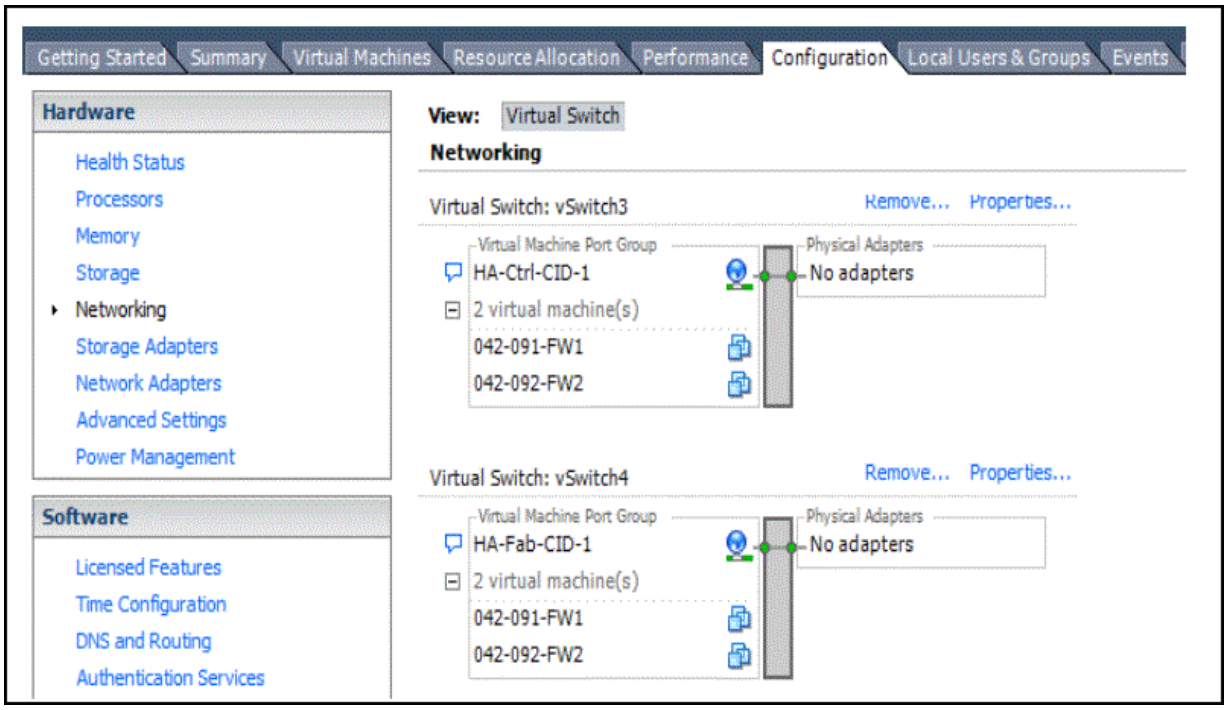
NOTE: When using multiple clusters that are connected to the same L2 domain, a unique cluster-id needs to be used for each cluster. Otherwise you may get duplicate mac addresses on the network, because the cluster-id is used to form the virtual interface mac addresses.

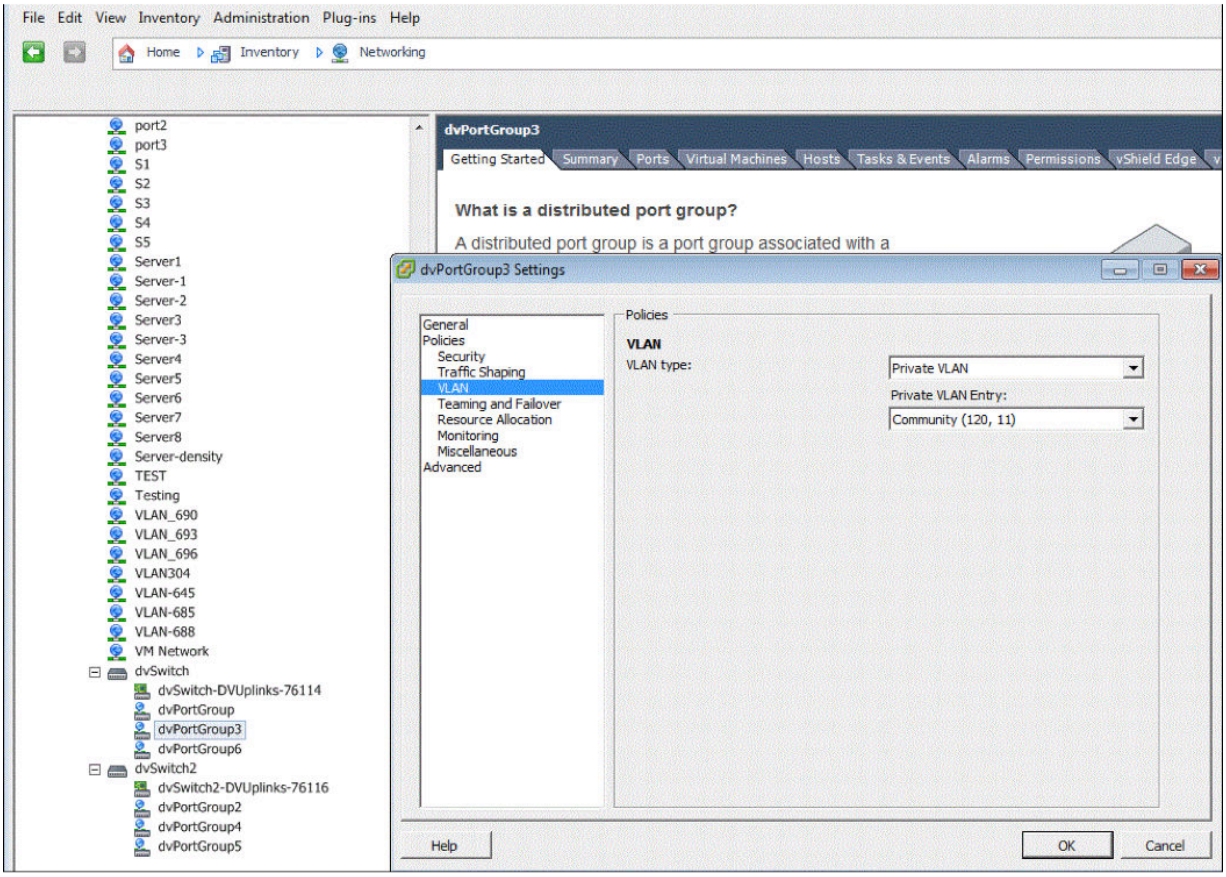
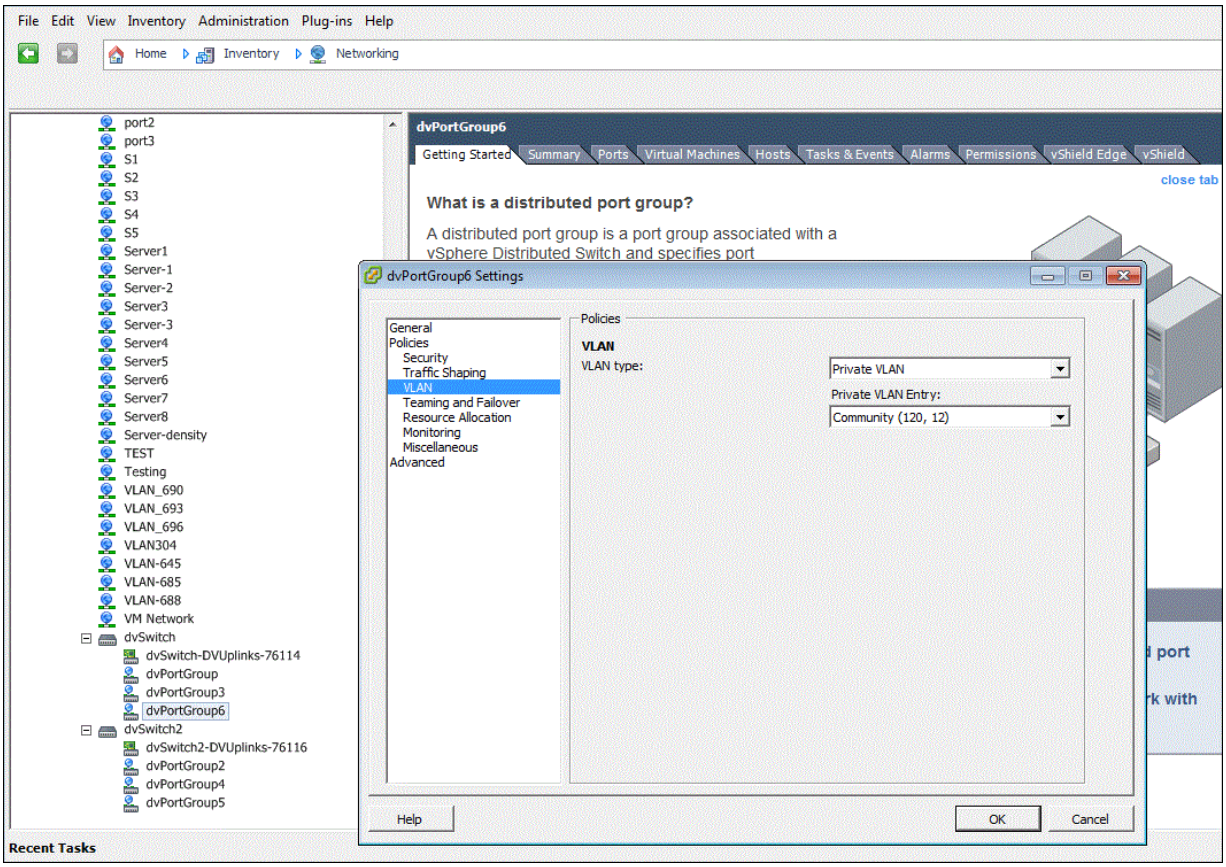
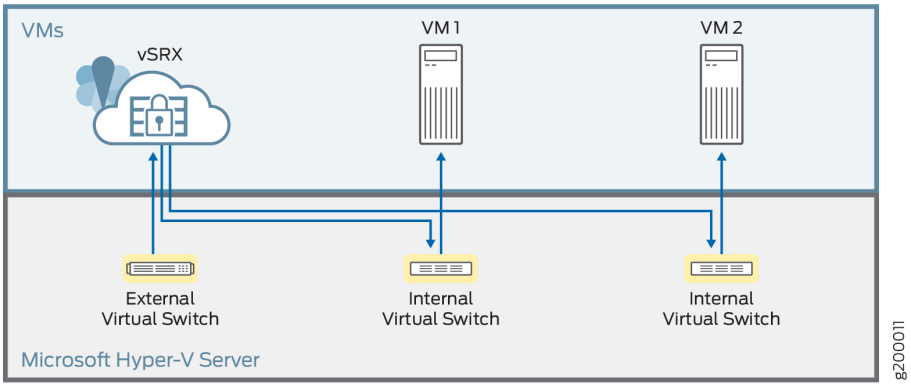
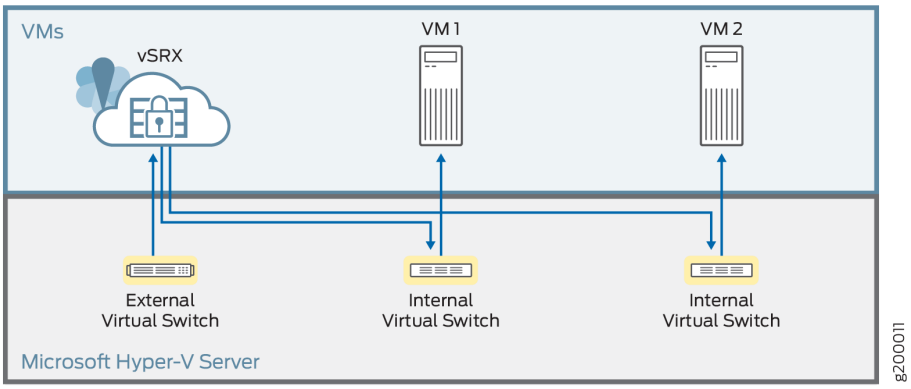
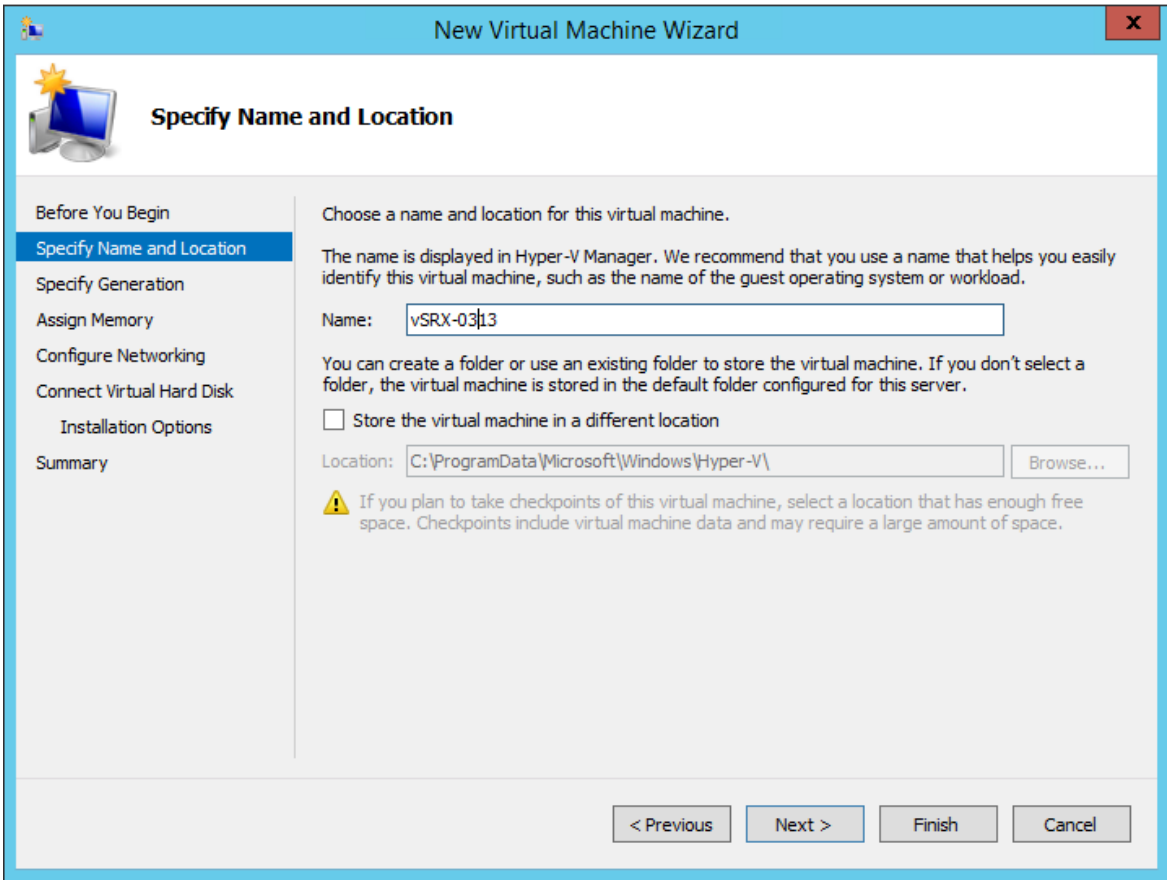
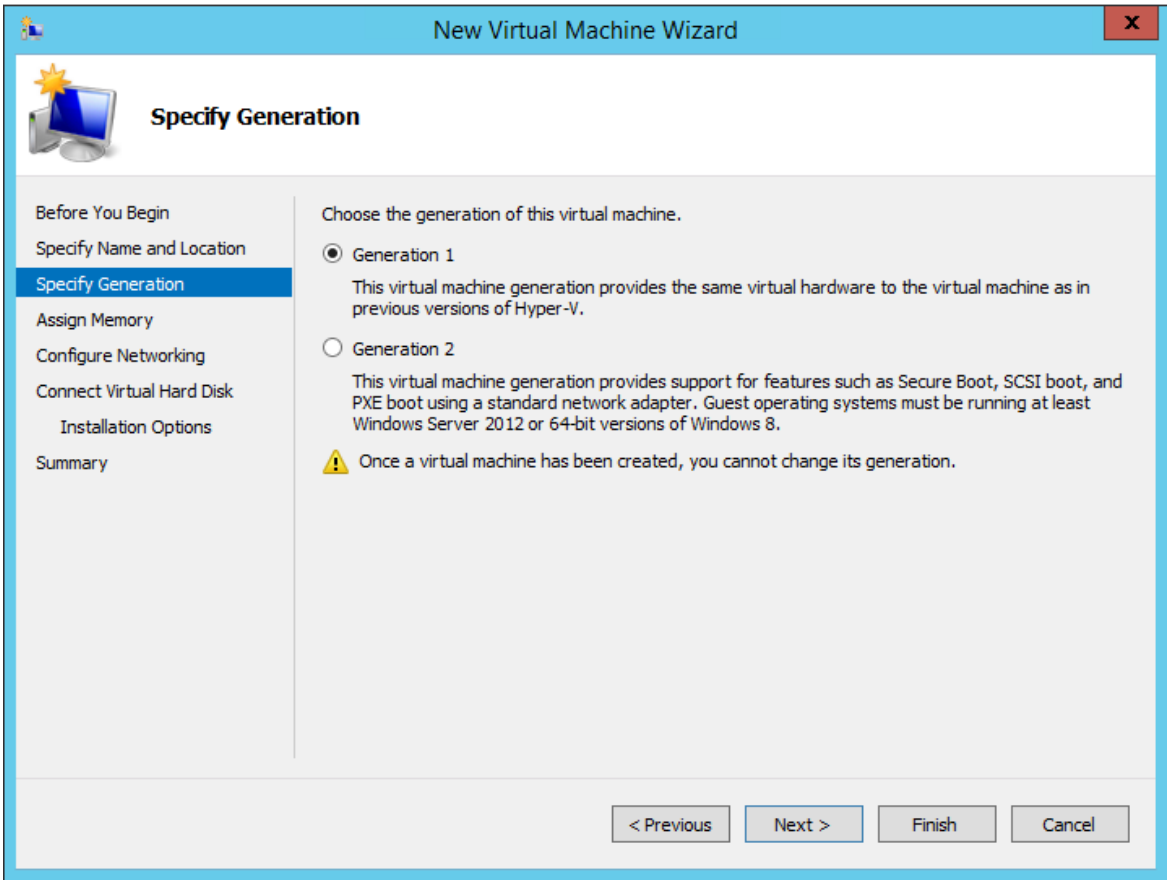
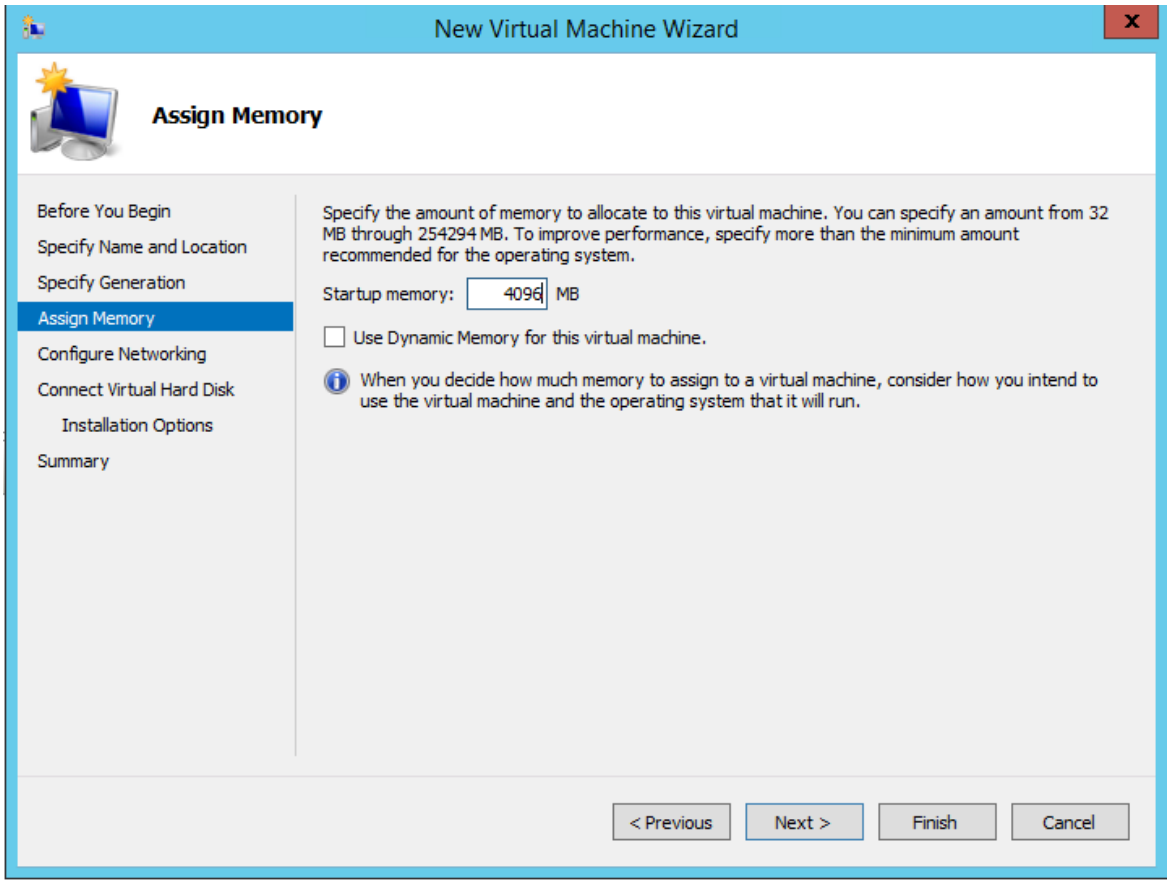
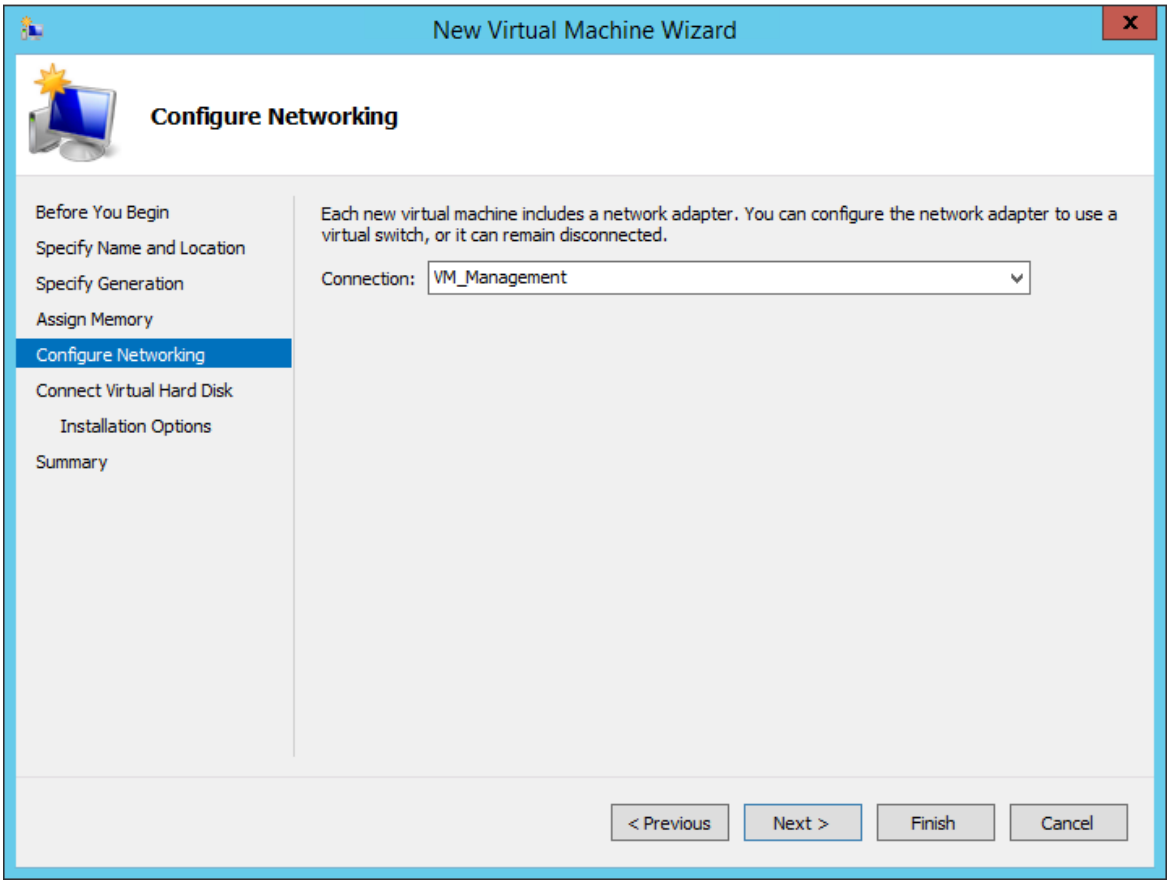
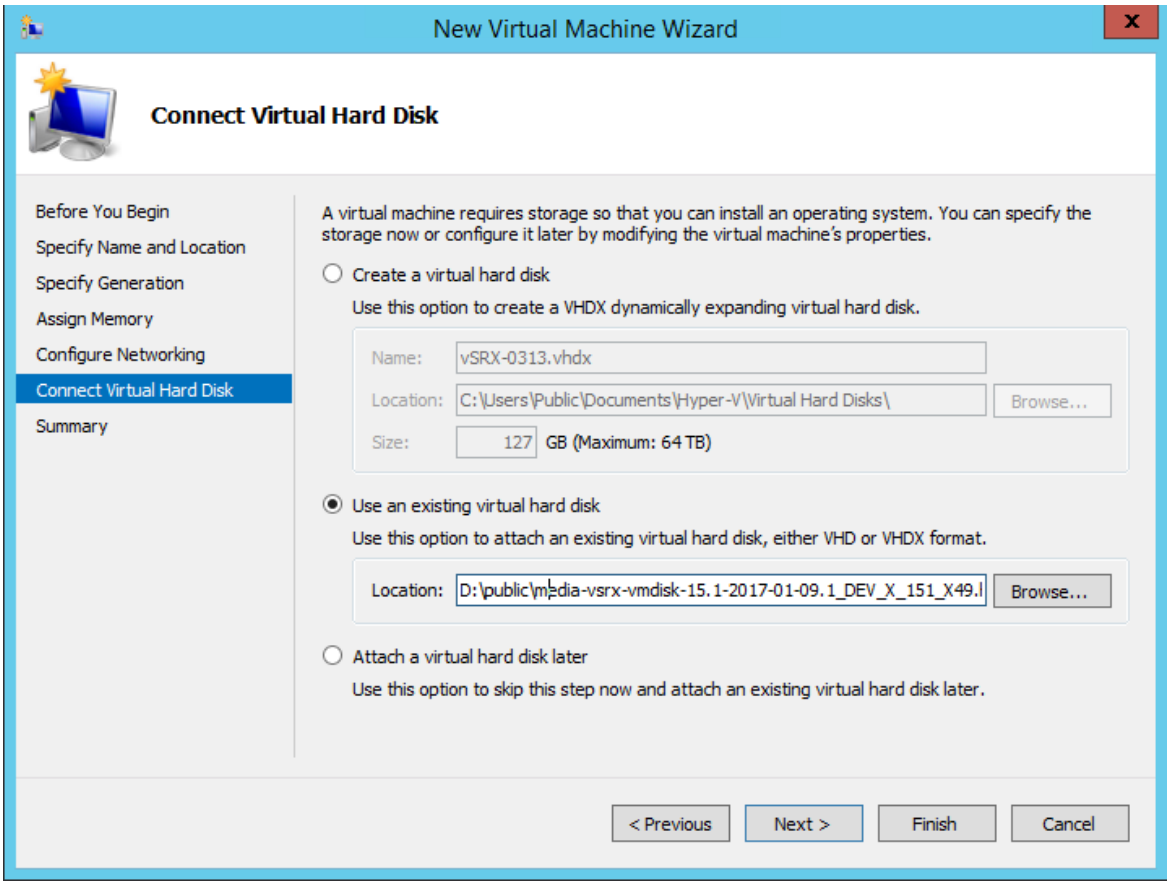
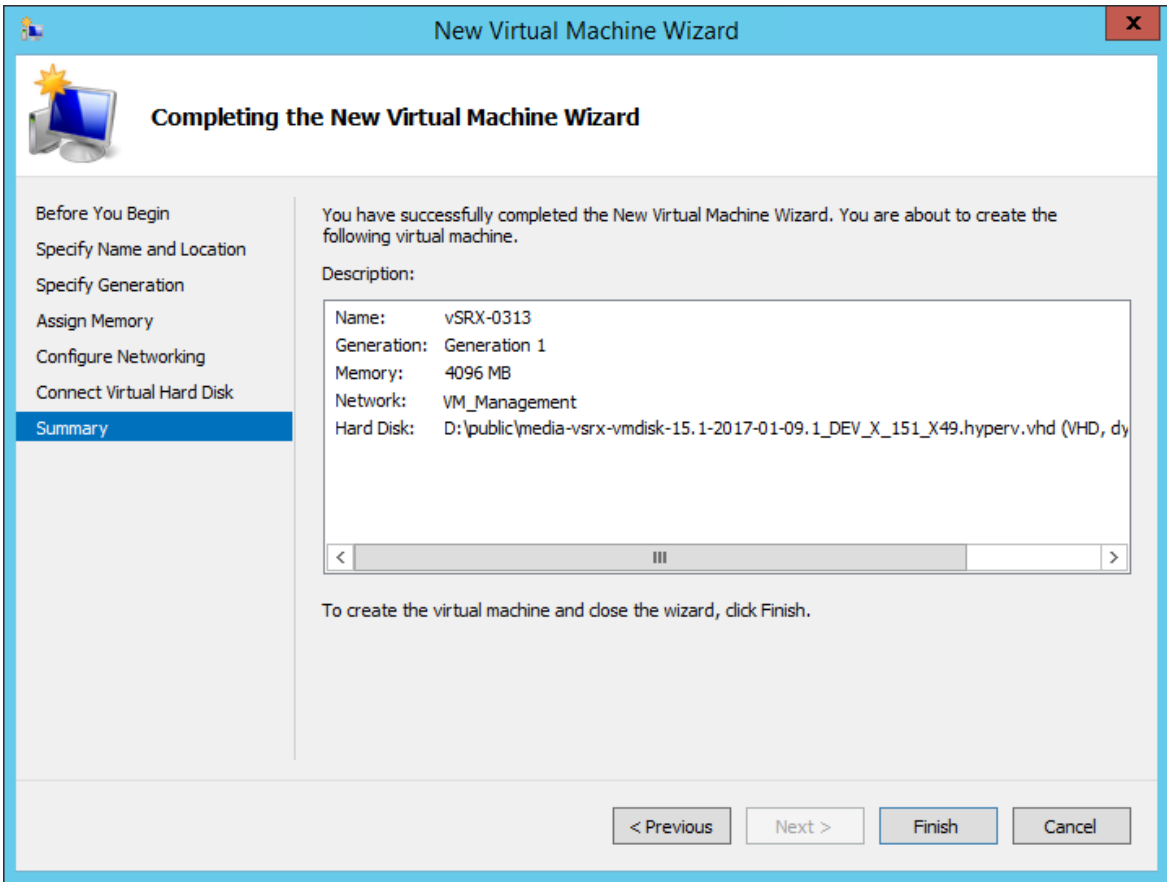
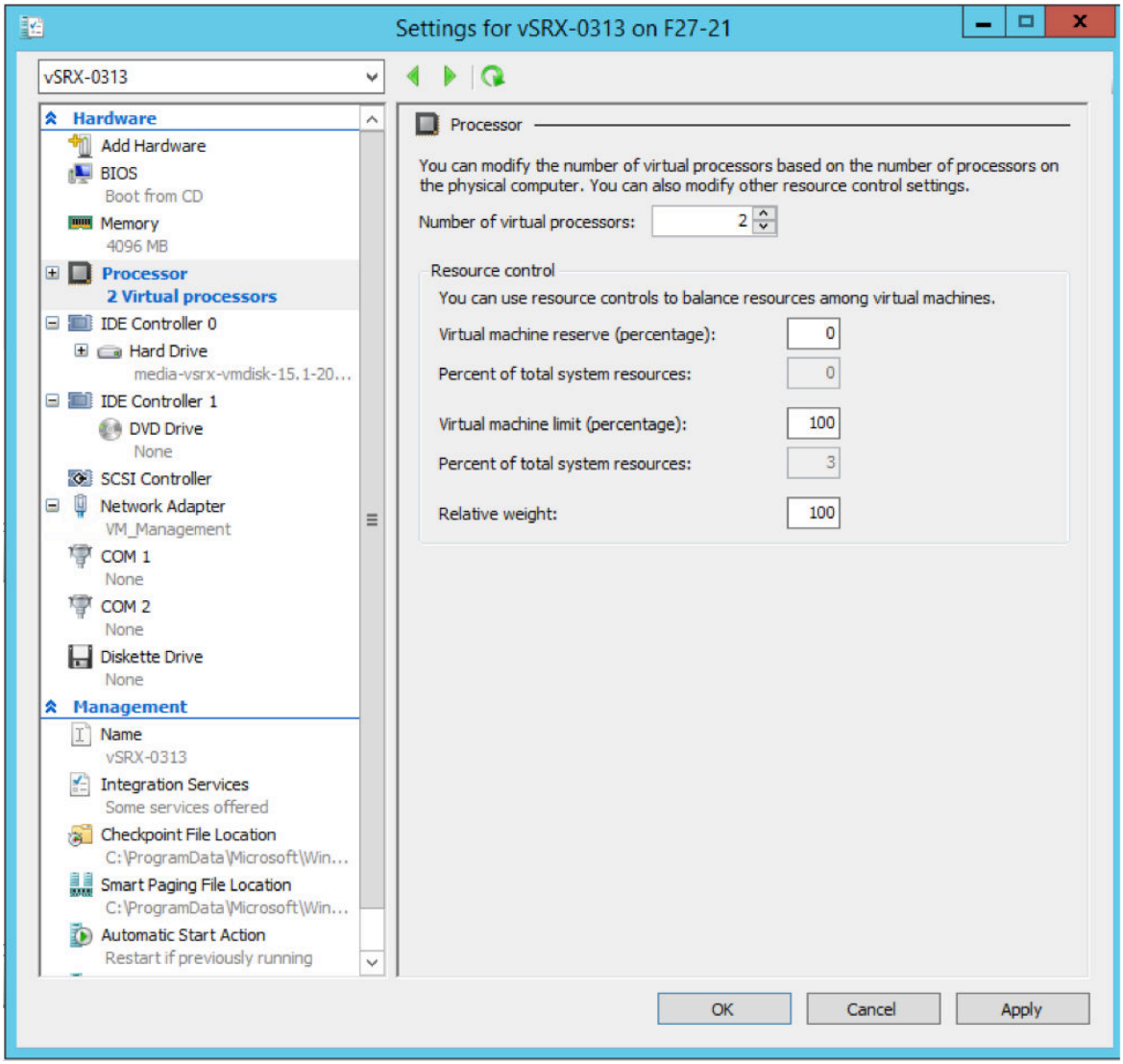
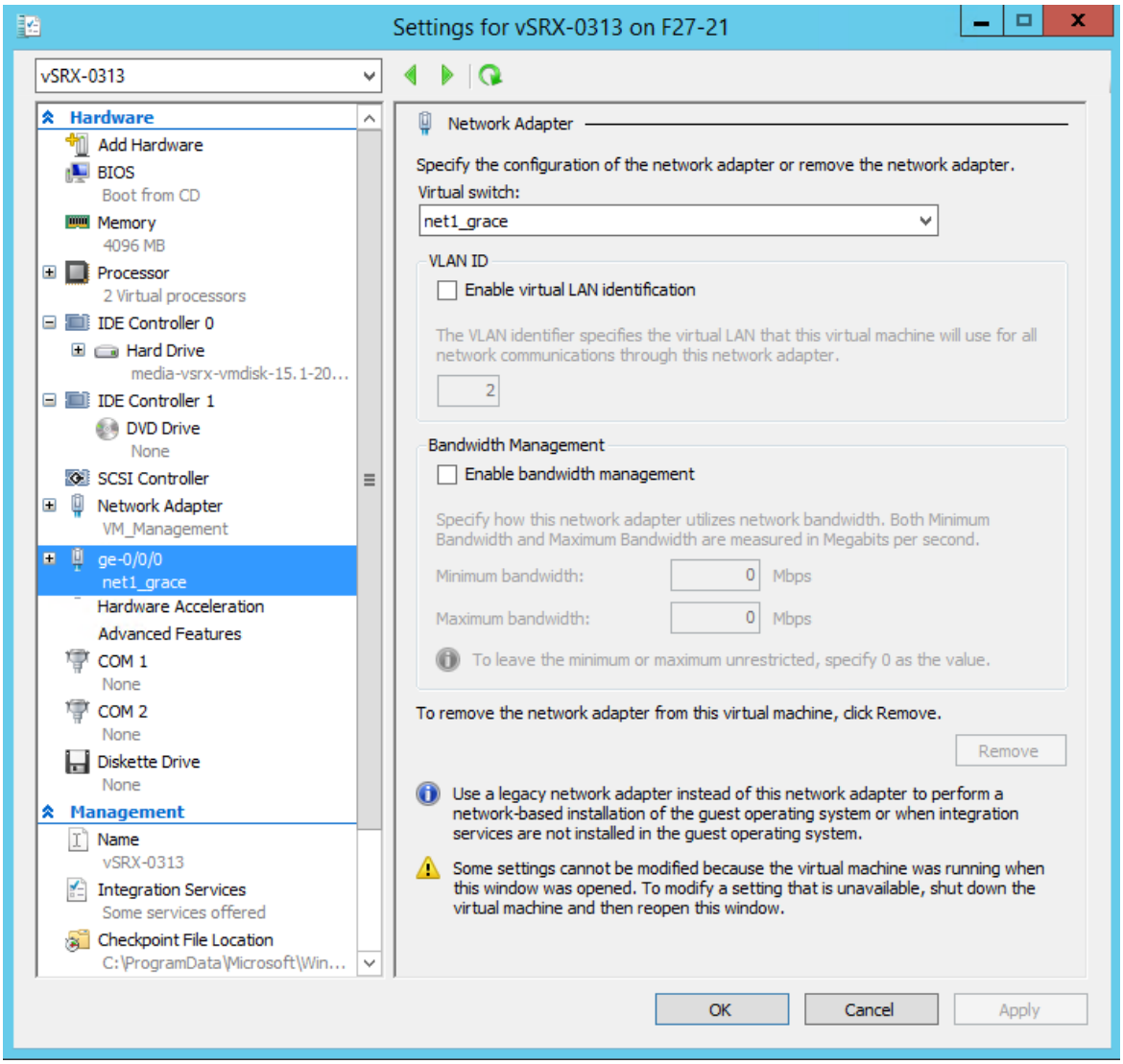
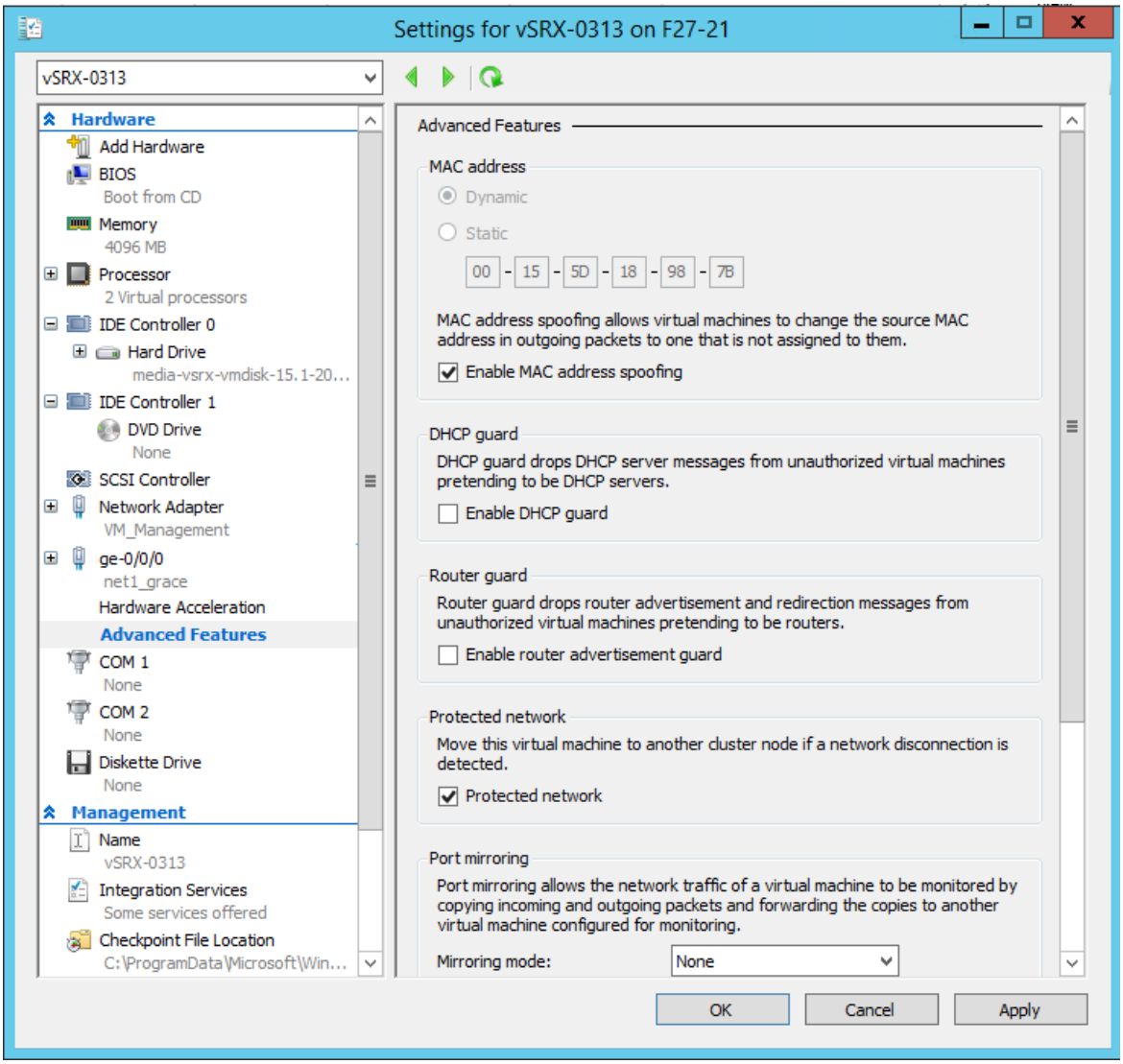
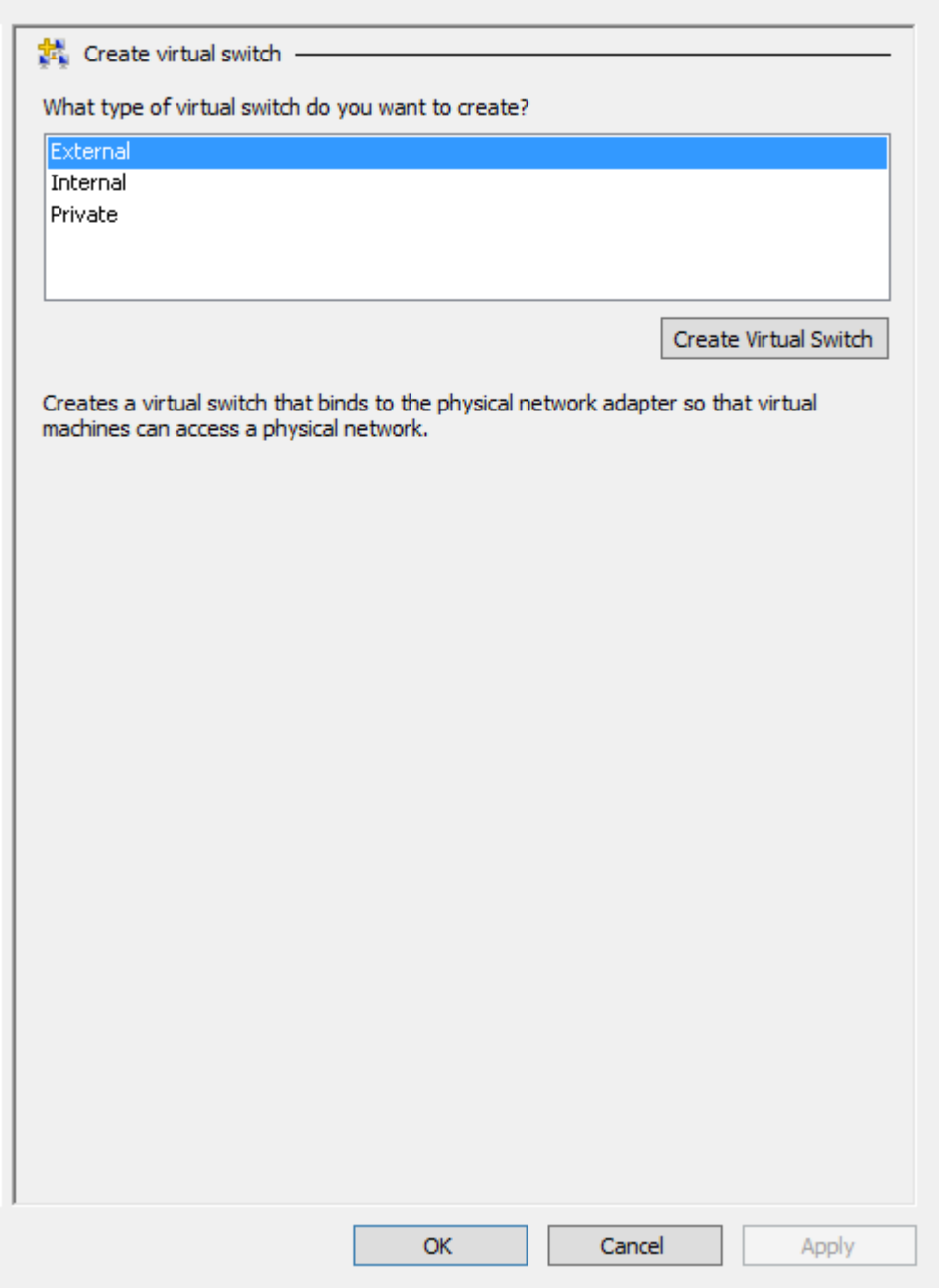
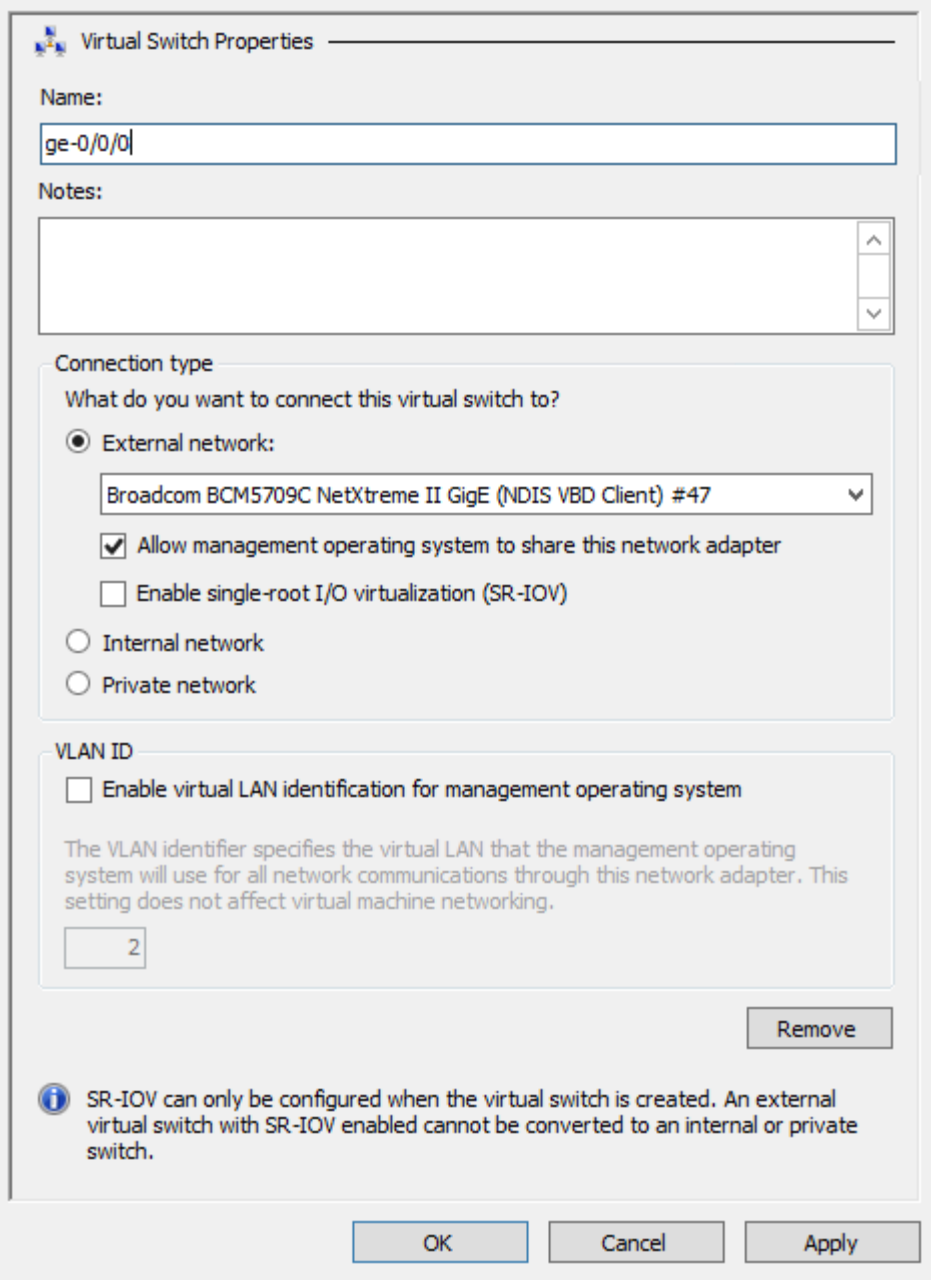
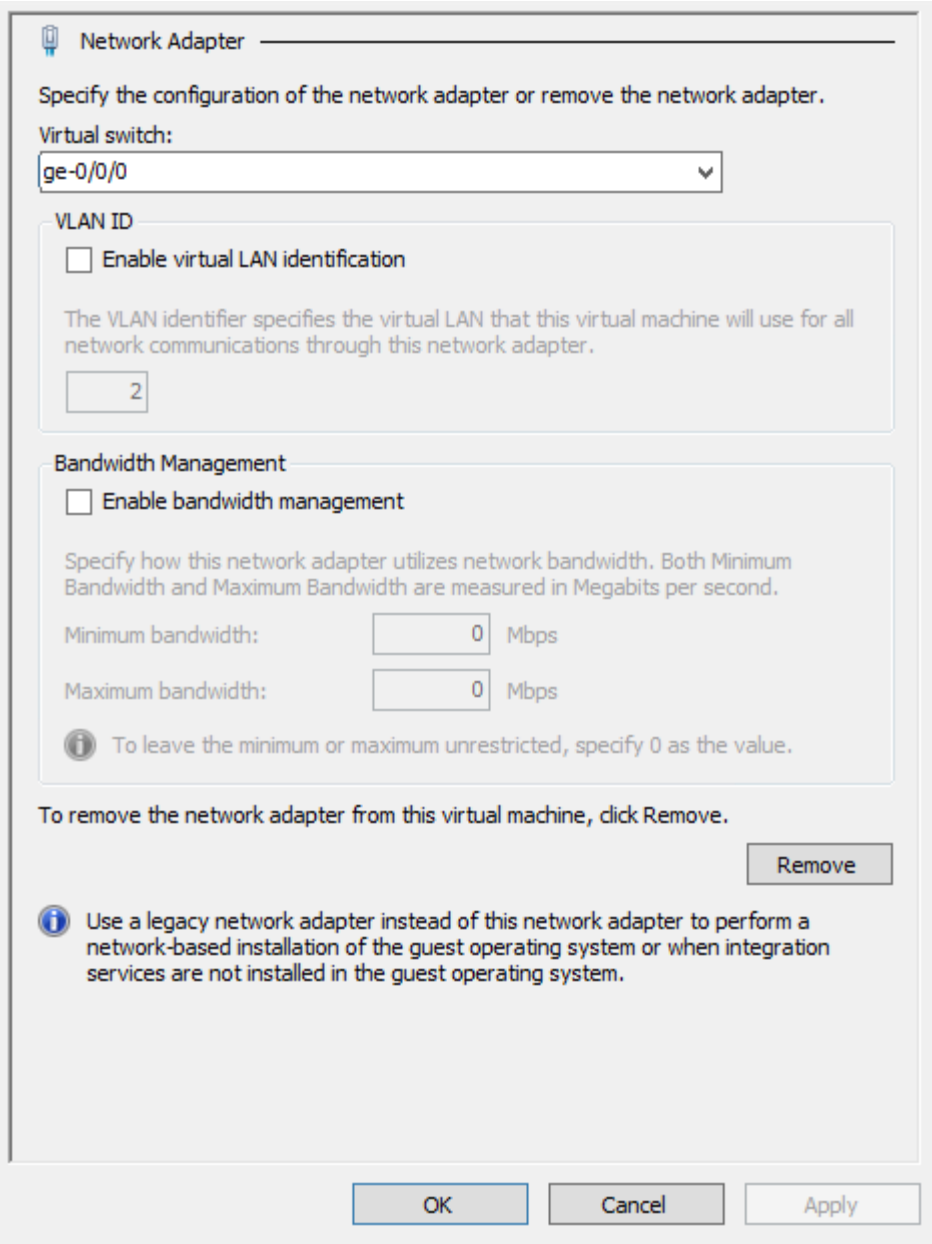
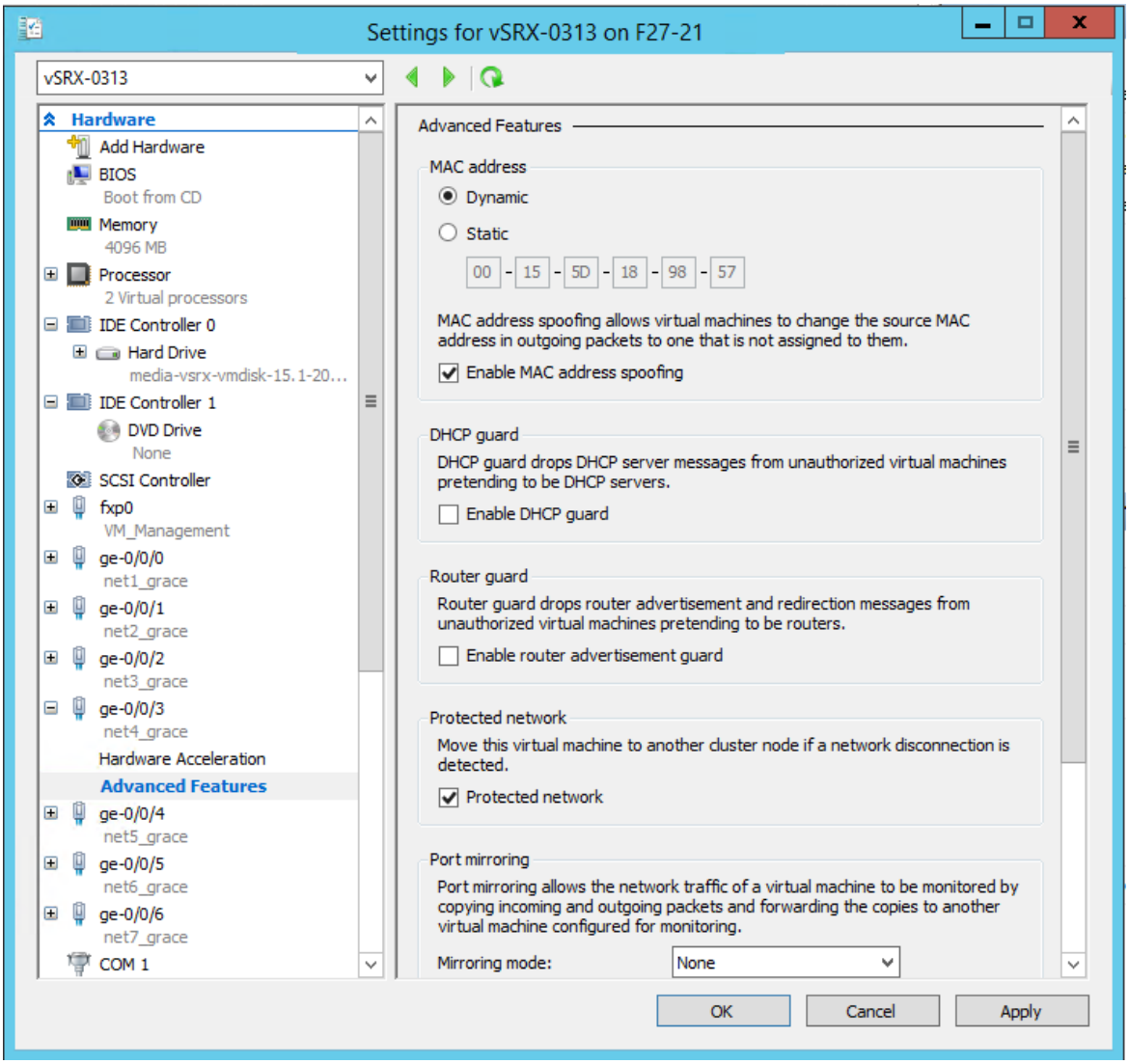
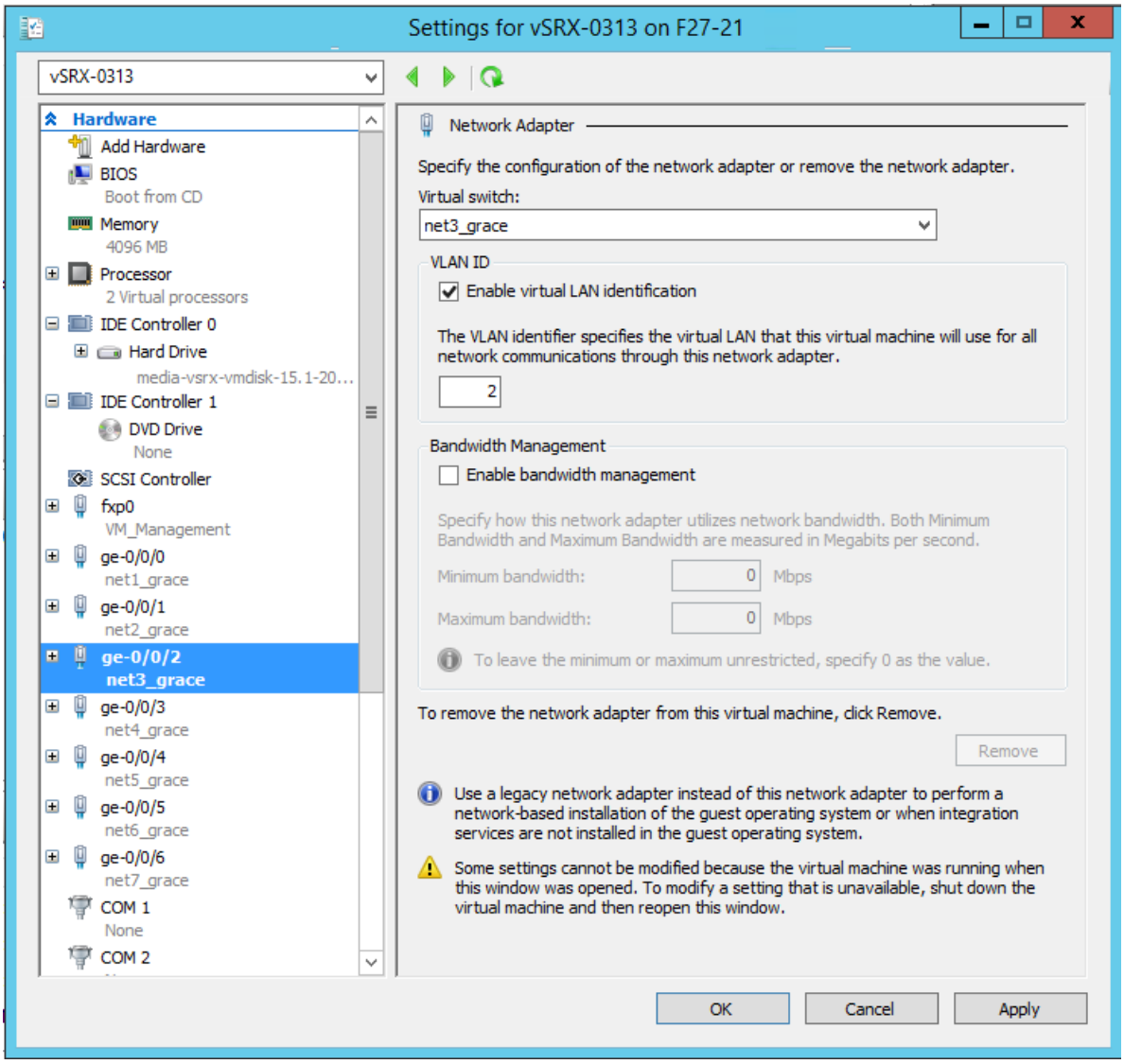
After reboot, on node 0, configure the fabric (data) ports of the cluster that are used to pass real-time objects (RTOs):

- ```
  user@vsrx0# set interfaces fab0 fabric-options member-interfaces ge-0/0/0
  user@vsrx0# set interfaces fab1 fabric-options member-interfaces ge-7/0/0
  ```

Chassis Cluster Quick Setup with J-Web

To configure chassis cluster from *J-Web*:

1. Enter the vSRX node 0 interface IP address in a Web browser.
2. Enter the vSRX username and password, and click **Log In**. The J-Web dashboard appears.
  ```

3. Click **Configuration Wizards>Chassis Cluster** from the left panel. The Chassis Cluster Setup wizard appears. Follow the steps in the setup wizard to configure the cluster ID and the two nodes in the cluster, and to verify connectivity.

> **NOTE**: Use the built-in Help icon in J-Web for further details on the Chassis Cluster Setup wizard.

Manually Configure a Chassis Cluster with J-Web

You can use the *J-Web* interface to configure the primary node 0 vSRX instance in the cluster. Once you have set the cluster and node IDs and rebooted each vSRX, the following configuration will automatically be synced to the secondary node 1 vSRX instance.

Select **Configure>Chassis Cluster>Cluster Configuration**. The Chassis Cluster configuration page appears.

Table 29 on page 147 explains the contents of the HA Cluster Settings tab.

Table 30 on page 149 explains how to edit the Node Settings tab.

Table 31 on page 149 explains how to add or edit the HA Cluster Interfaces table.

Table 32 on page 151 explains how to add or edit the HA Cluster Redundancy Groups table.

Table 29: Chassis Cluster Configuration Page

Field	Function
Node Settings	
Node ID	Displays the node ID.
Cluster ID	Displays the cluster ID configured for the node.
Host Name	Displays the name of the node.
Backup Router	Displays the router used as a gateway while the Routing Engine is in secondary state for redundancy-group 0 in a chassis cluster.
Management Interface	Displays the management interface of the node.

Table 29: Chassis Cluster Configuration Page *(Continued)*

Field	Function
IP Address	Displays the management IP address of the node.
Status	Displays the state of the redundancy group. • **Primary**–Redundancy group is active. • **Secondary**–Redundancy group is passive.
Chassis Cluster>HA Cluster Settings>Interfaces	
Name	Displays the physical interface name.
Member Interfaces/IP Address	Displays the member interface name or IP address configured for an interface.
Redundancy Group	Displays the redundancy group.
Chassis Cluster>HA Cluster Settings>Redundancy Group	
Group	Displays the redundancy group identification number.
Preempt	Displays the selected preempt option. • **True**–Primary Role can be preempted based on priority. • **False**–Primary Role cannot be preempted based on priority.
Gratuitous ARP Count	Displays the number of gratuitous Address Resolution Protocol (*ARP*) requests that a newly elected primary device in a chassis cluster sends out to announce its presence to the other network devices.
Node Priority	Displays the assigned priority for the redundancy group on that node. The eligible node with the highest priority is elected as primary for the redundant group.

Table 30: Edit Node Setting Configuration Details

Field	Function	Action
Node Settings		
Host Name	Specifies the name of the host.	Enter the name of the host.
Backup Router	Displays the device used as a gateway while the Routing Engine is in the secondary state for redundancy-group 0 in a chassis cluster.	Enter the IP address of the backup router.
Destination		
IP	Adds the destination address.	Click **Add**.
Delete	Deletes the destination address.	Click **Delete**.
Interface		
Interface	Specifies the interfaces available for the router. **NOTE**: Allows you to add and edit two interfaces for each fabric link.	Select an option.
IP	Specifies the interface IP address.	Enter the interface IP address.
Add	Adds the interface.	Click **Add**.
Delete	Deletes the interface.	Click **Delete**.

Table 31: Add HA Cluster Interface Configuration Details

Field	Function	Action
Fabric Link > Fabric Link 0 (fab0)		

Table 31: Add HA Cluster Interface Configuration Details *(Continued)*

Field	Function	Action
Interface	Specifies fabric link 0.	Enter the interface IP fabric link 0.
Add	Adds fabric interface 0.	Click **Add**.
Delete	Deletes fabric interface 0.	Click **Delete**.
Fabric Link > Fabric Link 1 (fab1)		
Interface	Specifies fabric link 1.	Enter the interface IP for fabric link 1.
Add	Adds fabric interface 1.	Click **Add**.
Delete	Deletes fabric interface 1.	Click **Delete**.
Redundant Ethernet		
Interface	Specifies a logical interface consisting of two physical Ethernet interfaces, one on each chassis.	Enter the logical interface.
IP	Specifies a redundant Ethernet IP address.	Enter a redundant Ethernet IP address.
Redundancy Group	Specifies the redundancy group ID number in the chassis cluster.	Select a redundancy group from the list.
Add	Adds a redundant Ethernet IP address.	Click **Add**.
Delete	Deletes a redundant Ethernet IP address.	Click **Delete**.

Table 32: Add Redundancy Groups Configuration Details

Field	Function	Action
Redundancy Group	Specifies the redundancy group name.	Enter the redundancy group name.
Allow preemption of primaryship	Allows a node with a better priority to initiate a failover for a redundancy group. **NOTE**: By default, this feature is disabled. When disabled, a node with a better priority does not initiate a redundancy group failover (unless some other factor, such as faulty network connectivity identified for monitored interfaces, causes a failover).	–
Gratuitous ARP Count	Specifies the number of gratuitous Address Resolution Protocol requests that a newly elected primary sends out on the active redundant Ethernet interface child links to notify network devices of a change in primary role on the redundant Ethernet interface links.	Enter a value from 1 to 16. The default is 4.
node0 priority	Specifies the priority value of node0 for a redundancy group.	Enter the node priority number as 0.
node1 priority	Specifies the priority value of node1 for a redundancy group.	Select the node priority number as 1.
Interface Monitor		
Interface	Specifies the number of redundant Ethernet interfaces to be created for the cluster.	Select an interface from the list.
Weight	Specifies the weight for the interface to be monitored.	Enter a value from 1 to 125.
Add	Adds interfaces to be monitored by the redundancy group along with their respective weights.	Click **Add**.

Table 32: Add Redundancy Groups Configuration Details *(Continued)*

Field	Function	Action
Delete	Deletes interfaces to be monitored by the redundancy group along with their respective weights.	Select the interface from the configured list and click **Delete**.
IP Monitoring		
Weight	Specifies the global weight for IP monitoring.	Enter a value from 0 to 255.
Threshold	Specifies the global threshold for IP monitoring.	Enter a value from 0 to 255.
Retry Count	Specifies the number of retries needed to declare reachability failure.	Enter a value from 5 to 15.
Retry Interval	Specifies the time interval in seconds between retries.	Enter a value from 1 to 30.
IPV4 Addresses to Be Monitored		
IP	Specifies the IPv4 addresses to be monitored for reachability.	Enter the IPv4 addresses.
Weight	Specifies the weight for the redundancy group interface to be monitored.	Enter the weight.
Interface	Specifies the logical interface through which to monitor this IP address.	Enter the logical interface address.
Secondary IP address	Specifies the source address for monitoring packets on a secondary link.	Enter the secondary IP address.
Add	Adds the IPv4 address to be monitored.	Click **Add**.
Delete	Deletes the IPv4 address to be monitored.	Select the IPv4 address from the list and click **Delete**.

SEE ALSO

Chassis Cluster Feature Guide for Security Devices

vSRX Cluster Staging and Provisioning for VMware

IN THIS SECTION

Staging and provisioning a vSRX cluster includes the following tasks:

Deploying the VMs and Additional Network Interfaces

The vSRX cluster uses three interfaces exclusively for clustering (the first two are predefined):

- Out-of-band management interface (fxp0).
- Cluster control link (em0).
- Cluster fabric links (fab0 and fab1). For example, you can specify ge-0/0/0 as fab0 on node0 and ge-7/0/0 as fab1 on node1.

Initially, the VM has only two interfaces. A cluster requires three interfaces (two for the cluster and one for management) and additional interfaces to forward data. You can add interfaces through the VMware vSphere Web Client.

1. On the VMware vSphere Web Client, click **Edit Virtual Machine Settings** for each VM to create additional interfaces.
2. Click **Add Hardware** and specify the attributes in Table 33 on page 154.

Table 33: Hardware Attributes

Attribute	Description
Adapter Type	Select VMXNET 3 from the list.
Network label	Select the network label from the list.
Connect at power on	Ensure that there is a check mark next to this option.

Creating the Control Link Connection Using VMware

To connect the control interface through the control vSwitch using the VMware vSphere Web Client:

1. Choose **Configuration > Networking**.
2. Click **Add Networking** to create a vSwitch for the control link.
   Choose the following attributes:
   - Connection Type
     - Virtual Machines
   - Network Access
     - Create a vSphere switch
     - No physical adapters
   - Port Group Properties
     - Network Label: HA Control
     - VLAN ID: None(0)

     NOTE: Port groups are not VLANs. The port group does not segment the vSwitch into separate broadcast domains unless the domains have different VLAN tags.
     - To use a VLAN as a dedicated vSwitch, you can use the default VLAN tag (0) or specify a VLAN tag.

- To use a VLAN as a shared vSwitch and use a port group, assign a VLAN tag on the port group for each chassis cluster link.

3. Right-click on the control network, click **Edit Settings**, and select **Security**.
4. Set the promiscuous mode to **Accept**, and click **OK**, as shown in Figure 18 on page 155.

Figure 18: Promiscuous Mode



NOTE: You must enable promiscuous mode on the control vSwitch for chassis cluster.

You can use the vSwitch default settings for the remaining parameters.

5. Click **Edit Settings** for both vSRX VMs to add the control interface (Network adapter 2) into the control vSwitch.

See Figure 19 on page 156 for vSwitch properties and Figure 20 on page 157 for VM properties for the control vSwitch.

Figure 19: Control vSwitch Properties



Figure 20: Virtual Machine Properties for the Control vSwitch



The control interface will be connected through the control vSwitch. See Figure 21 on page 157.

Figure 21: Control Interface Connected through the Control vSwitch



Creating the Fabric Link Connection Using VMware

To connect the fabric interface through the fabric vSwitch using the VMware vSphere Web Client:

1. Choose **Configuration > Networking**.
2. Click **Add Networking** to create a vSwitch for the fabric link.

   Choose the following attributes:

   - Connection Type
     - Virtual Machines
   - Network Access
     - Create a vSphere switch
     - No physical adapters
   - Port Group Properties
     - Network Label: HA Fabric
     - VLAN ID: None(0)

     NOTE: Port groups are not VLANs. The port group does not segment the vSwitch into separate broadcast domains unless the domains have different VLAN tags.
     - To use a VLAN as a dedicated vSwitch, you can use the default VLAN tag (0) or specify a VLAN tag.
     - To use VLAN as a shared vSwitch and use a port group, assign a VLAN tag on the port group for each chassis cluster link.

   Click **Properties** to enable the following features:

   - **General**-> **Advanced Properties**:
     - MTU: 9000
   - **Security**-> **Effective Polices**:
     - MAC Address Changes: Accept
     - Forged Transmits: Accept
3. Click **Edit Settings** for both vSRX VMs to add the fabric interface into the fabric vSwitch.

See Figure 22 on page 159 for vSwitch properties and Figure 23 on page 160 for VM properties for the fabric vSwitch.

Figure 22: Fabric vSwitch Properties



Figure 23: Virtual Machine Properties for the Fabric vSwitch



The fabric interface will be connected through the fabric vSwitch. See Figure 24 on page 161.

Figure 24: Fabric Interface Connected Through the Fabric vSwitch



Creating the Data Interfaces Using VMware

To map all the data interfaces to the desired networks:

1. Choose **Configuration > Networking**.
2. Click **Add Networking** to create a vSwitch for fabric link.

   Choose the following attributes:

   - Connection Type
     - Virtual Machines
   - Network Access
     - Create a vSphere switch
     - No physical adapters
   - Port Group Properties
     - Network Label: chassis cluster Reth

- VLAN ID: None(0)

Click **Properties** to enable the following features:

- **Security**-> **Effective Polices**:
  - MAC Address Changes: Accept
  - Forged Transmits: Accept

The data interface will be connected through the data vSwitch using the above procedure.

Prestaging the Configuration from the Console

The following procedure explains the configuration commands required to set up the vSRX chassis cluster. The procedure powers up both nodes, adds the configuration to the cluster, and allows SSH remote access.

1. Log in as the root user. There is no password.
2. Start the CLI.

```
root#cli
root@>
```

3. Enter configuration mode.

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

4. Copy the following commands and paste them into the CLI:

```
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.42.81/24
set groups node0 system hostname vsrx-node0
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.42.82/24
set groups node1 system hostname vsrx-node1
set apply-groups "${node}"
```

5. Set the root authentication password by entering a cleartext password, an encrypted password, or an SSH public key string (DSA or RSA).

```
root@# set system root-authentication plain-text-password
New password: password
```

```
Retype new password: password
set system root-authentication encrypted-password "$ABC123"
```

6. To enable SSH remote access:

```
user@host#set system services ssh
```

7. To enable IPv6:

```
user@host#set security forwarding-options family inet6 mode flow-based
```

   This step is optional and requires a system reboot.

8. Commit the configuration to activate it on the device.

```
user@host#commit
commit complete
```

9. When you have finished configuring the device, exit configuration mode.

```
user@host#exit
```

Connecting and Installing the Staging Configuration

After the vSRX cluster initial setup, set the cluster ID and the node ID, as described in *Configure a vSRX Chassis Cluster in Junos OS*.

After reboot, the two nodes are reachable on interface fxp0 with SSH. If the configuration is operational, the `show chassis cluster status` command displays output similar to that shown in the following sample output.

vsrx> `show chassis cluster status`

```
Cluster ID: 1
Node                  Priority          Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
    node0                   100         secondary      no       no
    node1                   150         primary        no       no
```

```
Redundancy group: 1 , Failover count: 1
    node0                   100         secondary       no       no
    node1                   150         primary         no       no
```

A cluster is healthy when the primary and secondary nodes are present and both have a priority greater than 0.

Deploy vSRX Chassis Cluster Nodes Across Different ESXi Hosts Using dvSwitch

Before you deploy the vSRX chassis cluster nodes for ESXi 6.0 (or greater) hosts using distributed virtual switch (dvSwitch), ensure that you make the following configuration settings from the vSphere Web Client to ensure that the high-availability cluster control link works properly between the two nodes:

- In the dvSwitch switch settings of the vSphere Web Client, disable IGMP snooping for Multicast filtering mode (see Multicast Snooping on a vSphere Distributed Switch).
- In the dvSwitch port group configuration of the vSphere Web Client, enable promiscuous mode (see Configure the Security Policy for a Distributed Port Group or Distributed Port).

This chassis cluster method uses the private virtual LAN (PVLAN) feature of dvSwitch to deploy the vSRX chassis cluster nodes at different ESXi hosts. There is no need to change the external switch configurations.

On the VMware vSphere Web Client, for dvSwitch, there are two PVLAN IDs for the primary and secondary VLANs. Select **Community** in the menu for the secondary VLAN ID type.

Use the two secondary PVLAN IDs for the vSRX control and fabric links. See Figure 25 on page 165 and Figure 26 on page 166.

Figure 25: dvPortGroup3 Settings



Figure 26: dvPortGroup6 Settings



NOTE: The configurations described above must reside at an external switch to which distributed switch uplinks are connected. If the link at the external switch supports native VLAN, then VLAN can be set to none in the distributed switch port group configuration. If native VLAN is not supported on the link, this configuration should have VLAN enabled.

You can also use regular VLAN on a distributed switch to deploy vSRX chassis cluster nodes at different ESXi hosts using dvSwitch. Regular VLAN works similarly to a physical switch. If you want to use regular VLAN instead of PVLAN, disable IGMP snooping for chassis cluster links.

However, use of PVLAN is recommended because:

- PVLAN does not impose IGMP snooping.
- PVLAN can save VLAN IDs.

NOTE: When the vSRX cluster across multiple ESXi hosts communicates through physical switches, then you need to consider the other Layer 2 parameters at: https://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/NT21/ LAHAAppNotev4.pdf.

3

PART

vSRX Deployment for Microsoft Hyper-V

CHAPTER 9

Overview

IN THIS CHAPTER

Understand vSRX with Microsoft Hyper-V

IN THIS SECTION

This section presents an overview of vSRX as deployed in Microsoft Hyper-V.

vSRX in Microsoft Hyper-V

Microsoft Hyper-V is a hypervisor-based virtualization technology. It provides software infrastructure and basic management tools that you can use to create and manage a virtualized server computing environment. This virtualized environment can be used to address a variety of business goals aimed at improving efficiency and reducing costs. Hyper-V works on x86- and x64-based systems running Windows.

You deploy a vSRX virtual security appliance on a Microsoft Hyper-V server to provide networking security features for the virtualized server computing environment. Hyper-V implements isolation of virtual machines in terms of a partition. The vSRX virtual machine runs in Microsoft Hyper-V as a child partition.

Note the following for deploying vSRX on a Microsoft Hyper-V server:

- Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.

- Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.

Figure 27 on page 170 illustrates the deployment of a vSRX in a Hyper-V environment to provide security for applications running on one or more virtual machines.

Figure 27: vSRX Deployment in Hyper-V



Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.

RELATED DOCUMENTATION

- Hyper-V on Windows Server 2016
- Microsoft Hyper-V Overview
- Microsoft Hyper-V

Requirements for vSRX on Microsoft Hyper-V

IN THIS SECTION

This section presents an overview of requirements for deploying a vSRX instance on Microsoft Hyper-V.

Software Requirements

Table 34 on page 171 lists the software requirements for the vSRX instance on Microsoft Hyper-V.

NOTE: Only the vSRX small flavor is supported on Microsoft Hyper-V. vSRX 3.0 multi-CPU versions are supported on Microsoft Hyper-V.

Table 34: Specifications for vSRX for Microsoft Hyper-V

Component	Specification
Hypervisor support	• Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012. • Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.
Memory	4 GB
Disk space	16 GB (IDE or SCSI drives)

Table 34: Specifications for vSRX for Microsoft Hyper-V *(Continued)*

Component	Specification
vCPUs	2
Virtual network adapters	8 Hyper-V specific network adapters

Table 35: Specifications for vSRX 3.0 for Microsoft Hyper-V

Component	Specification
Hypervisor support	Microsoft Hyper-V Windows Server 2016
Memory	4 GB
Disk space	18 GB (IDE)
vCPUs	2
Virtual network adapters	8 Hyper-V specific network adapters

Starting in Junos OS Release 19.1R1, the vSRX 3.0 instance supports guest OS with 2 vCPUs, 4-GB virtual RAM, and a 18-GB disk space on Microsoft Hyper-V and Azure for improved performance.

Hardware Requirements

Table 36 on page 173 lists the hardware specifications for the host machine that runs the vSRX VM.

Table 36: Hardware Specifications for the Host Machine

Component	Specification
Host memory size	Minimum 4 GB
Host processor type	x86 or x64-based multicore processor **NOTE**: DPDK requires Intel Virtualization VT-x/VT-d support in the CPU. See *About Intel Virtualization Technology.*
Gigabit (10/100/1000baseT) Ethernet adapter	Emulates the multiport DEC 21140 10/100TX 100 MB Ethernet network adapter with one to four network connections.

Best Practices for Improving vSRX Performance

Review the following practices to improve vSRX performance.

NUMA Nodes

The x86 server architecture consists of multiple sockets and multiple cores within a socket. Each socket also has memory that is used to store packets during I/O transfers from the NIC to the host. To efficiently read packets from memory, guest applications and associated peripherals (such as the NIC) should reside within a single socket. A penalty is associated with spanning CPU sockets for memory accesses, which might result in nondeterministic performance. For vSRX, we recommend that all vCPUs for the vSRX VM are in the same physical non-uniform memory access (NUMA) node for optimal performance.

CAUTION: The Packet Forwarding Engine (PFE) on the vSRX will become unresponsive if the NUMA nodes topology is configured in the hypervisor to spread the instance's vCPUs across multiple host NUMA nodes. vSRX requires that you ensure that all vCPUs reside on the same NUMA node.

We recommend that you bind the vSRX instance with a specific NUMA node by setting NUMA node affinity. NUMA node affinity constrains the vSRX VM resource scheduling to only the specified NUMA node.

Interface Mapping for vSRX on Microsoft Hyper-V

Each network adapter defined for a vSRX is mapped to a specific interface, depending on whether the vSRX instance is a standalone VM or one of a cluster pair for high availability.

NOTE: Starting in Junos OS Release 15.1X49-D100 for vSRX, support for chassis clustering to provide network node redundancy is only available on Microsoft Hyper-V Server 2016.

Note the following:

- In standalone mode:
  - fxp0 is the out-of-band management interface.
  - ge-0/0/0 is the first traffic (revenue) interface.
- In cluster mode:
  - fxp0 is the out-of-band management interface.
  - em0 is the cluster control link for both nodes.
  - Any of the traffic interfaces can be specified as the fabric links, such as ge-0/0/0 for fab0 on node 0 and ge-7/0/0 for fab1 on node 1.

Table 37 on page 174 shows the interface names and mappings for a standalone vSRX VM.

Table 37: Interface Names for a Standalone vSRX VM

Network Adapter	Interface Name in Junos OS
1	fxp0
2	ge-0/0/0
3	ge-0/0/1
4	ge-0/0/2
5	ge-0/0/3

Table 37: Interface Names for a Standalone vSRX VM *(Continued)*

Network Adapter	Interface Name in Junos OS
6	ge-0/0/4
7	ge-0/0/5
8	ge-0/0/6

Table 38 on page 175 shows the interface names and mappings for a pair of vSRX VMs in a cluster (node 0 and node 1).

Table 38: Interface Names for a vSRX Cluster Pair

Network Adapter	Interface Name in Junos OS
1	fxp0 (node 0 and 1)
2	em0 (node 0 and 1)
3	ge-0/0/0 (node 0) ge-7/0/0 (node 1)
4	ge-0/0/1 (node 0) ge-7/0/1 (node 1)
5	ge-0/0/2 (node 0) ge-7/0/2 (node 1)
6	ge-0/0/3 (node 0) ge-7/0/3 (node 1)
7	ge-0/0/4 (node 0) ge-7/0/4 (node 1)

Table 38: Interface Names for a vSRX Cluster Pair *(Continued)*

Network Adapter	Interface Name in Junos OS
8	ge-0/0/5 (node 0) ge-7/0/5 (node 1)

vSRX Default Settings on Microsoft Hyper-V

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.

Table 39 on page 176 lists the factory-default settings for security policies on the vSRX.

Table 39: Factory Default Settings for Security Policies

Source Zone	Destination Zone	Policy Action
trust	untrust	permit
trust	trust	permit
untrust	trust	deny

Release History Table

Release	Description
19.1R1	Starting in Junos OS Release 19.1R1, the vSRX 3.0 instance supports guest OS with 2 vCPUs, 4-GB virtual RAM, and a 18-GB disk space on Microsoft Hyper-V and Azure for improved performance.
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.

15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 for vSRX, support for chassis clustering to provide network node redundancy is only available on Microsoft Hyper-V Server 2016.

RELATED DOCUMENTATION

- KB Article - Interface must be in the same routing instance as the other interfaces in the zone
- About Intel Virtualization Technology
- DPDK Release Notes

CHAPTER 10

Install vSRX in Microsoft Hyper-V

IN THIS CHAPTER

Prepare for vSRX Deployment in Microsoft Hyper-V

Note the following guidelines when deploying vSRX on a Microsoft Hyper-V server:

- Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.
- Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.
- Ensure that the host CPU supports a 64-bit x86 Intel processor and is running Windows.
- Ensure that you have a user account with administrator permissions to enable the computer to deploy a vSRX virtual machine (VM) using either Microsoft Hyper-V Manager or Windows PowerShell.
- Create the virtual switches on the Hyper-V host computer necessary to support the fxp0 (out-of-band management) interface and the traffic (revenue) interface supported by the vSRX VM. You create virtual switches using either the Microsoft Hyper-V Manager or Windows PowerShell. See *Add vSRX Interfaces* for details on adding virtual switches for the vSRX VM using the Virtual Switch Manager.

Figure 28 on page 179 illustrates the deployment of a vSRX in a Hyper-V environment to provide security for applications running on one or more virtual machines.

Figure 28: Example of vSRX Deployment in Hyper-V



Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.

RELATED DOCUMENTATION

- Install Hyper-V and Create a Virtual Machine
- Create a Virtual Machine in Hyper-V
- Create a Virtual Switch for Hyper-V Virtual Machines
- Hyper-V Virtual Switch

Deploy vSRX in a Hyper-V Host Using the Hyper-V Manager

Use this procedure to deploy and configure the vSRX as a virtual security appliance in the Hyper-V environment using Hyper-V Manager.

Note the following for deploying vSRX on a Microsoft Hyper-V server:

- Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.
- Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.

NOTE: To upgrade an existing vSRX instance, see *Migration, Upgrade, and Downgrade* in the *vSRX Release Notes*.

To deploy vSRX using Hyper-V Manager:

1. Download the vSRX software image for Microsoft Hyper-V from the Juniper Networks website. The vSRX disk image supported by Microsoft Hyper-V is a virtual hard disk (VHD) format file.

   CAUTION: Do not change the filename of the downloaded software image or the installation will fail.

2. Log onto your Hyper-V host computer using the Administrator account.
3. Open the Hyper-V Manager by selecting **Start > Administrative Tools > Hyper-V Manager**. The welcome page for Hyper-V appears the first time that you open Hyper-V Manager.
4. Create a virtual machine by selecting **Action > New > Virtual Machine**. The Before You Begin screen appears for the New Virtual Machine Wizard. Click **Next** to move through each page of the wizard, or you can click the name of a page in the left pane to move directly to that page.

5. From the Specify Name and Location page (see Figure 29 on page 181), enter a name and location for the vSRX VM that you are creating and then click **Next**. We recommend that you keep this name the same as the hostname you intend to assign to the vSRX VM.

Figure 29: Specify Name and Location Page



6. From the Specify Generation page (see Figure 30 on page 182), keep the default setting of **Generation 1** as the generation of the vSRX VM and then click **Next**.

Figure 30: Specify Generation Page



7. From the Assign Memory page (see Figure 31 on page 183), enter **4096 MB** as the amount of startup memory to assign to the vSRX VM. Leave **Use Dynamic Memory for this virtual machine** clear. Click **Next**.

Figure 31: Assign Memory Page



8. From the Configure Networking page (see Figure 32 on page 184), select a virtual switch from a list of existing virtual switches on the Hyper-V host computer to connect to the vSRX management interface. The default is **Not connected**. Click **Next**.

NOTE: See *Add vSRX Interfaces* for the procedure on adding virtual switches for the vSRX VM using the Virtual Switch Manager.

Figure 32: Configure Networking Page



9. From the Connect Virtual Hard Disk page (see Figure 33 on page 185), click **Use an existing virtual hard disk** and browse to the location of the vSRX virtual hard disk (VHD) file (downloaded in Step 1). Click **Next**.

Figure 33: Connect Virtual Hard Disk Page



10. After you have finished configuring the new virtual machine, verify your selections in the Summary page (see Figure 34 on page 186) and then click **Finish** to complete the installation.

Figure 34: Summary Page



11. Right-click the vSRX VM and select **Settings** from the context menu.

12. From the Settings dialog box, under the Hardware section, select **Processor**. The Processor pane appears (see Figure 35 on page 187). Enter **2** in the **Number of virtual processors** field (the default is 1).

Figure 35: Processor Pane



13. From the Settings dialog box, under the Hardware section, select **Network Adapter**. The Network Adapter pane appears (see Figure 36 on page 188).

    From the Virtual switch drop-down list, select a virtual switch to assign to a network adapter to be used by the vSRX VM (see *Add vSRX Interfaces* for details on adding virtual switches). Each network adapter that is defined for a vSRX is mapped to a specific interface. See *Requirements for vSRX on Microsoft Hyper-V* for a summary of interface names and mappings for a vSRX VM.

NOTE: If you need to add a network adapter to assign to a virtual switch, click **Add Hardware > Network Adapter > Add**.

Figure 36: Network Adapter Pane



14. Enable the MAC address spoofing function for the vSRX VM if a network adapter is to be used as an interface for Layer 2 mode support on the vSRX. From the Network Adapter pane select **Advanced Features**. The Advanced Features pane appears (see Figure 37 on page 189). Click the **Enable MAC address spoofing** check box.

    MAC address spoofing allows each network adapter to change its source MAC address for outgoing packets to one that is not assigned to them. Enabling MAC address spoofing ensures those packets

are not dropped by the network adapter if the source MAC address fails to match the outgoing interface MAC address.

Click **OK** when you complete your vSRX VM selections.

Figure 37: Network Adapter Advanced Features Pane



15. On Microsoft Hyper-V Server 2016, you will need to enable nested virtualization for the vSRX VM before you power on the vSRX instance. This procedure can only be performed in the Hyper-V environment using Windows PowerShell (see, *Deploy vSRX in a Hyper-V Host Using Windows PowerShell*, Step 9). You cannot enable nested virtualization from the Hyper-V Manager because nessted virtualization is not supported on Microsoft Hyper-V Server 2012.

NOTE:

NOTE: Nested virtualization can only be configured on a host running Microsoft Hyper-V Server 2016. In addition, Dynamic Memory must be disabled on the virtual machine containing the nested instance of Hyper-V.

16. Launch and power on the vSRX instance in the Hyper-V Manager by selecting the vSRX VM from the list of virtual machines. Right-click and select **Start** from the context menu (or select **Action > Start**).
17. Configure the basic settings for the vSRX (see *Configure vSRX Using the CLI*).

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.

RELATED DOCUMENTATION

- Install Hyper-V and Create a Virtual Machine
- Create a Virtual Machine in Hyper-V
- Virtual Machine Settings in Hyper-V Manager Explained

Deploy vSRX in a Hyper-V Host Using Windows PowerShell

Use this procedure to deploy and configure the vSRX as a virtual security appliance in the Hyper-V environment using Windows PowerShell.

Note the following for deploying vSRX on a Microsoft Hyper-V server:

- Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.

- Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.

NOTE: To upgrade an existing vSRX instance, see *Migration, Upgrade, and Downgrade* in the *vSRX Release Notes*.

To deploy vSRX using Windows PowerShell:

1. Download the vSRX software image for Microsoft Hyper-V from the Juniper Networks website. The vSRX disk image supported by Microsoft Hyper-V is a virtual hard disk (VHD) format file.

   CAUTION: Do not change the filename of the downloaded software image or the installation will fail.

2. On the Windows desktop, click the **Start** button and type **Windows PowerShell**.
3. Right-click **Windows PowerShell** and select **Run as administrator**.
4. Run the following command to enable Hyper–V using PowerShell:

   ```
   Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
   ```

5. Enter the `New-VM` command to create the vSRX VM. The command syntax is as follows:

   ```
   PS C:>\Users\Administrator> New-VM -Name <Name> -MemoryStartupBytes <Memory> -BootDevice <BootDevice> -
   VHDPath <VHDPath> -Path <Path> -Generation <Generation> -Switch <SwitchName>
   ```

   See Table 40 on page 191 for a summary of the parameters in the `New-VM` command.

Table 40: New-VM Command Parameters

Parameter	Description
-Name	Specify a name for the vSRX VM that you are creating. We recommend keeping this name the same as the hostname you intend to give to the vSRX VM.
-MemoryStartupBytes	Enter `4GB` as the amount of startup memory to assign to the vSRX VM.
-BootDevice	Enter `VHD` as the device that the vSRX VM boots to when it starts.

Table 40: New-VM Command Parameters *(Continued)*

Parameter	Description
-VHDPath	Specify the location of the vSRX virtual hard disk (VHD) file that you want to deploy.
-Path	Specify the location to store the vSRX VM configuration files.
-Generation	Enter `1` to create a generation 1 virtual machine for the vSRX.
-SwitchName	Specify the name of the virtual switch that you want the vSRX VM to assign to a network adapter used by the vSRX VM. Each network adapter that is defined for a vSRX is mapped to a specific interface. See *Requirements for vSRX on Microsoft Hyper-V* for a summary of interface names and mappings for a vSRX VM. **NOTE**: To locate the name of a previously created virtual switch, use the `Get-VMSwitch` command. See *Add vSRX Interfaces* for the procedure on adding virtual switches for the vSRX VM using the Virtual Switch Manager.

The following is an example of the `New-VM` command syntax for creating a vSRX VM:

```
PS C:>\Users\Administrator> New-VM -Name vSRX_0109 -MemoryStartupBytes 4GB -BootDevice VHD -VHDPath C:\Users
\Public\Documents\Hyper-V\vsrx-0109-powershell\vsrx\media-vsrx-vmdisk-151X49D80.hyper-v.vhd -Path 'C:\Users
\Public\Documents\Hyper-V\vsrx-0109\' Generation 1 SwitchName test
```

6. Set the number of processors for the newly created vSRX VM by entering the `Set-VMProcessor` command. Specify `Count 2` for the number of processors. For example:

   ```
   PS C:>\Users\Administrator> Set-VMProcessor -VMName <vSRVName> -Count 2
   ```

7. Verify the newly created vSRX VM by entering the `Get-VM` command. For example:

   ```
   PS C:>\Users\Administrator> Get-VM -VMName <vSRVName>
   ```

   The output for the command is as follows:

   ```
   Name       State    CPUUSage(%) MemoryAssigned(M)  Uptime    State                Version
   vSRX_0109  Off        0           0                 00:00:00 Operating normally     8.0
   ```

8. Enable the MAC address spoofing function for the vSRX VM if a network adapter is to be used as an interface for Layer 2 mode support on the vSRX. MAC address spoofing allows the vSRX VM's

network adapter to change its source MAC address for outgoing packets to one that is not assigned to them. Enabling MAC address spoofing ensures those packets are not dropped by the network adapter if the source MAC address fails to match the outgoing interface MAC address.

The command syntax is as follows:

```
PS C:>\Users\Administrator> Set-VMNetworkAdapter -VMName <vSRVName> -computerName <HyperVHostName> -
VMNetworkAdapter <NetworkAdapterName> -MacAddressSpoofing On
```

Verify that MacAddressSpoofing is On.

```
PS C:>\Users\Administrator> Get-VMNetworkAdapter -VMName <vSRVName> -computerName <HyperVHostName> | fl
<HyperVHostName>name,macaddressspoofing
```

The output for the command is as follows:

```
Name                 : vSRX_0109
MacAddressSpoofing   : On
```

9. Enable nested virtualization for the vSRX VM by using the `Set-VMProcessor` command, where **`VMName`** is the name of the vSRX VM you created. By default, the virtualization extensions are disabled for each VM. Nested virtualization allows you to run Hyper-V inside of a Hyper-V virtual machine. For example:

   ```
   PS C:>\Users\Administrator> Set-VMProcessor -VMName <vSRX_0109> -ExposeVirtualizationExtensions $true
   ```

   NOTE: Nested virtualization can only be configured on a host running Microsoft Hyper-V Server 2016. In addition, Dynamic Memory must be disabled on the virtual machine containing the nested instance of Hyper-V.

10. Launch and power on the vSRX VM by using the `Start-VM` command, where **`Name`** is the name of the vSRX VM you created. For example:

    ```
    PS C:>\Users\Administrator> Start-VM -Name <vSRX_0109>
    ```

11. Configure the basic settings for the vSRX (see *Configure vSRX Using the CLI*).

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX only on Microsoft Hyper-V Server 2012 R2 or 2012.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, you can deploy the vSRX on Microsoft Hyper-V Server 2016.

RELATED DOCUMENTATION

- Hyper-V Module for Windows PowerShell
- Create a Virtual Machine in Hyper-V
- Run Hyper-V in a Virtual Machine with Nested Virtualization

CHAPTER 11

vSRX VM Management with Microsoft Hyper-V

IN THIS CHAPTER

Configure vSRX Using the CLI

To configure the instance using the CLI:

1. Verify that the vSRX instance is powered on.
2. Log in as the root user (whose username is *root*). There is no password.
3. Start the CLI.

```
root#cli
root@>
```

4. Enter configuration mode.

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

5. Set the root authentication password by entering a *cleartext* password, an encrypted password, or an SSH public key string (*DSA* or *RSA*). The following is an example of a plain-text password. The CLI prompts you for the password and then encrypts it.

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

```
New password: password
Retype new password: password
```

6. Configure the hostname.

```
[edit]
root@# set system host-name host-name
```

7. Configure the management interface.

```
[edit]
root@# set interfaces fxp0 unit 0 family inet dhcp-client
```

8. Configure the traffic interfaces.

```
[edit]
root@# set interfaces ge-0/0/0 unit 0 family inet dhcp-client
```

9. Configure basic security zones and bind them to traffic interfaces.

```
[edit]
root@# set security zones security-zone trust interfaces ge-0/0/0.0
```

10. Verify the configuration changes.

```
[edit]
root@# commit check
configuration check succeeds
```

11. Commit the configuration to activate it on the instance.

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

NOTE: Certain Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds

to each licensed feature. To conform to software feature licensing requirements, you must purchase one license per feature per instance. The presence of the appropriate software unlocking key on your virtual instance allows you to configure and use the licensed feature. See Managing Licenses for vSRX for details.

RELATED DOCUMENTATION

- CLI User Guide
- Junos OS for SRX Series

Configure vSRX Using the J-Web Interface

IN THIS SECTION

Access the J-Web Interface and Configuring vSRX

To configure vSRX using the *J-Web* Interface:

1. Launch the J-Web interface from a Web browser.

   NOTE: You will be prompted to accept a system-generated certificate to access a vSRX VM using the J-Web interface.

2. Enter the vSRX out-of-band management (fxp0) interface IP address in the Address box.
3. Specify the username and password.
4. Click **Log In**, and select the **Configuration Wizards** tab from the left navigation panel. The J-Web Setup wizard page opens.

5. Click **Setup**.

   You can use the Setup wizard to configure the vSRX VM or edit an existing configuration.

   - Select **Edit Existing Configuration** if you have already configured the wizard using the factory mode.
   - Select **Create New Configuration** to configure the vSRX VM using the wizard.

     The following configuration options are available in the guided setup:

     - Basic

       Select **basic** to configure the vSRX VM name and user account information as shown in Table 41 on page 198.

       - Instance name and user account information

         Table 41: Instance Name and User Account Information

Field	Description
Instance name	Type the name of the vSRX instance.
Root password	Create a default root user password.
Verify password	Verify the default root user password.
Operator	Add an optional administrative account in addition to the root account. User role options include: • **Super User**: This user has full system administration rights and can add, modify, and delete settings and users. • **Operator**: This user can perform system operations such as a system reset but cannot change the configuration or add or modify users. • **Read only**: This user can only access the system and view the configuration. • **Disabled**: This user cannot access the system.

     - Select either **Time Server** or **Manual**. Table 42 on page 199 lists the system time options.

Table 42: System Time Options

Field	Description
Time Server	
Host Name	Type the hostname of the time server. For example: **ntp.example.com**.
IP	Type the IP address of the time server in the IP address entry field. For example: **192.0.2.254**.
NOTE: You can enter either the hostname or the IP address.	
Manual	
Date	Click the current date in the calendar.
Time	Set the hour, minute, and seconds. Choose **AM** or **PM**.
Time Zone (mandatory)	
Time Zone	Select the time zone from the list. For example: GMT Greenwich Mean Time GMT.

- Expert

  a. Select **Expert** to configure the basic options as well as the following advanced options:

     - Four or more internal zones
     - Internal zone services
     - Application of security policies between internal zones

  b. Click the **Need Help** icon for detailed configuration information.

  You see a success message after the basic configuration is complete.

Apply the Configuration

To apply the configuration settings for vSRX:

1. Review and ensure that the configuration settings are correct, and click **Next**. The Commit Configuration page appears.
2. Click **Apply Settings** to apply the configuration changes to vSRX.
3. Check the connectivity to vSRX, as you might lose connectivity if you have changed the management zone IP. Click the URL for reconnection instructions on how to reconnect to the instance.
4. Click **Done** to complete the setup.

   After successful completion of the setup, you are redirected to the J-Web interface.

> ⚠ **CAUTION**: After you complete the initial setup, you can relaunch the J-Web Setup wizard by clicking **Configuration>Setup**. You can either edit an existing configuration or create a new configuration. If you create a new configuration, the current configuration in vSRX will be deleted.

Add vSRX Feature Licenses

Certain Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds to each licensed feature. To conform to software feature licensing requirements, you must purchase one license per feature per instance. The presence of the appropriate software unlocking key on your virtual instance allows you to configure and use the licensed feature.

See Managing Licenses for vSRX for details.

Add vSRX Interfaces

IN THIS SECTION

The Hyper-V virtual switch is a software-based Layer 2 Ethernet network switch that connects VMs to either physical or virtual networks. A virtual switch can be configured from Hyper-V Manager or Windows PowerShell . The Hyper-V host uses the virtual switches to connect virtual machines to the internet through the host computer's network connection. You configure networking for the vSRX by adding, removing, and modifying its associated network adapters in the Hyper-V host as necessary.

NOTE: To perform this procedure, you must have appropriate permissions. Contact your Virtual Server administrator to request the proper permissions to add a virtual switch and network adapter..

For the vSRX VM, you pair a network adapter with a virtual switch for the vSRX to receive and transmit traffic. You map network adapters to the specific vSRX interfaces: Network adapter 1 is mapped to the fxp0 (out-of-band management) interface, network adapter 2 is mapped to the ge-0/0/0 (revenue) interface, network adapter 3 is mapped to ge-0/0/1, and so on (see *Requirements for vSRX on Microsoft Hyper-V*). Hyper-V supports a maximum of eight network adapters.

NOTE: When adding virtual switches, there are no limits imposed by Hyper-V. The practical limit depends on the available computing resources.

This section includes the following topics on adding vSRX interfaces in Hyper-V:

Add Virtual Switches

To add virtual switches for the vSRX VM using the Virtual Switch Manager in the Hyper-V Manager:

1. Open the Hyper-V Manager by selecting **Start > Administrative Tools > Hyper-V Manager**.
2. Select **Action > Virtual Switch Manager**. The Virtual Switch Manager appears.

3. Under the Virtual Switches section, select **New virtual network switch**. The Create Virtual Switch pane appears (see Figure 38 on page 202).

Figure 38: Create Virtual Switch Pane



4. Choose the type of virtual switch to create:

- External—Gives virtual machines access to a physical network to communicate with servers and clients on an external network. It allows virtual machines on the same Hyper-V server to communicate with each other.
- Internal—Allows communication between virtual machines on the same Hyper-V server, and between the virtual machines and the management host operating system.
- Private—Allows communication only between virtual machines on the same Hyper-V server. A private network is isolated from all external network traffic on the Hyper-V server. This type of network is useful when you must create an isolated networking environment, like an isolated test domain.

In most cases when adding a vSRX network adapter, select **External** as the type of virtual switch. Internal and private virtual switches are intended to keep network traffic within the Hyper-V server.

NOTE: For the fxp0 (out-of-band management) interface, connect it to External virtual switch, which could connect to an external network.

For the ge-0/0/0 (revenue port) interface, if only communication between VMs in the same Hyper-V server is needed, Internal or Private virtual switch should be sufficient. However, if communication between the VM and an external network is needed, connect it to External virtual switch.

5. Select **Create Virtual Switch**. The Virtual Switch Properties pane appears (see Figure 39 on page 204).

Figure 39: Virtual Switch Properties Pane



6. Specify a name for the virtual switch.
7. Choose the physical network interface card b(NIC) that you want to use (only a requirement when you select **External**).

8. Isolate network traffic from the management Hyper-V host operating system or other virtual machines that share the same virtual switch by selecting **Enable virtual LAN identification**. You can change the VLAN ID to any number or leave the default. See "Configure the vSRX to Use a VLAN" on page 208 for details.
9. Click **OK**, then click **Yes** to apply networking changes and to close the Virtual Switch Manager window.
10. If necessary, repeat Steps 3 through 9 to add additional network adapters for use by the vSRX VM.
11. Right-click the vSRX VM and select **Settings** from the context menu. From the Settings dialog box, under the Hardware section, click **Network Adapter**. The Network Adapter pane appears (see Figure 40 on page 206).

12. From the Virtual switch drop-down list, select the **virtual switch** that you want to assign to this network adapter. See *Requirements for vSRX on Microsoft Hyper-V* for a summary of interface names and mappings for a vSRX VM.

Figure 40: Adding Virtual Switch to Network Adapter Example



13. If a network adapter is to be used as an interface for Layer 2 mode support on the vSRX, then from the Network Adapter pane select **Advanced Features**. Select the **Enable MAC address spoofing** check box to enable the MAC address spoofing function for the network adapter (see Figure 41 on page 207).

MAC address spoofing allows each network adapter to change its source MAC address for outgoing packets to one that is not assigned to them. Enabling MAC address spoofing ensures those packets are not dropped by the network adapter if the source MAC address fails to match the outgoing interface MAC address.

Figure 41: Network Adapter Enable MAC Address Spoofing Example



14. Click **Apply** and **OK** to save the changes in the Settings dialog box.
15. Launch and power on the vSRX instance in the Hyper-V Manager by selecting the vSRX VM from the list of virtual machines, and then right-click and select **Start** from the context menu (or select **Action > Start**).

SEE ALSO

- Create a Virtual Switch for Hyper-V Virtual Machines
- Create a Virtual Network

Configure the vSRX to Use a VLAN

Hyper-V supports the configuration of VLANs on a network adapter in the host computer. For each network adapter that you configure for the vSRX VM, if required, you can add a VLAN identifier to specify the VLAN that the vSRX VM will use for all network communications through the network adapter.

By default, Hyper-V enables trunk mode for a VLAN. Trunk mode allows multiple VLAN IDs to share a connection between the physical network adapter and the physical network.

To give the vSRX VM external access on the virtual network in multiple VLANs, you will need to configure the port on the physical network to be in trunk mode. You will also need to know the specific VLANs that are used and all of the VLAN IDs used by the virtual machines that the virtual network supports.

To utilize a Hyper-V VLAN, ensure that you are using a physical network adapter that supports 802.1q VLAN tagging. By default, the virtual network adapter in Hyper-V is in untagged mode and you might need to enable the feature on a virtual network adapter.

NOTE: By using Windows PowerShell, you can determine the mode of the vNIC (`Get-VmNetworkAdapterVlan` command) and change the mode of the vNIC (`Set-VmNetworkAdapterVlan` command). See Get-VMNetworkAdapterVlan and Set-VMNetworkAdapterVlan for details on both Windows PowerShell virtual network adapter commands.

To add a VLAN for a vSRX VM virtual network adapter:

1. Open the Hyper-V Manager by selecting **Start > Administrative Tools > Hyper-V Manager**.
2. Right-click the vSRX VM and select **Settings** from the context menu.
3. From the Settings dialog box, under the Hardware section, select the network adapter connected to the external virtual network. The Network Adapter pane appears.
4. Select **Enable virtual LAN identification**, and then enter the VLAN ID you intend to use (see Figure 42 on page 209). You can change the VLAN ID to any number or leave the default. This is the VLAN

identification number that the vSRX will use for all network communication through this network adapter.

Figure 42: Enable VLAN Identification Example



5. Click **OK**, and then click **Yes** to apply networking changes.
6. If necessary, repeat Steps 3 through 5 to add VLAN identification to additional network adapters in use by the vSRX VM.

SEE ALSO

Hyper-V: Configure VLANs and VLAN Tagging

Understanding Hyper-V VLANs

Power Down a vSRX VM with Hyper-V

In situations where you need to modify the vSRX VM settings from Hyper-V, you must first perform a graceful shut down of the vSRX VM using the **Shut Down** command. The vSRX VM performs an orderly closing of all programs and attempts to shut off power to avoid data loss.

NOTE: If you are using Microsoft PowerShell, use the `Stop-VM` command to perform a graceful shutdown of the vSRX VM.

To gracefully shut down the vSRX instance on the Hyper-V host computer:

1. Log onto your Hyper-V host computer using the Administrator account.
2. Open the Hyper-V Manager by selecting **Start > Administrative Tools > Hyper-V Manager**.
3. Power down the vSRX instance in the Hyper-V Manager by selecting the vSRX VM from the list of virtual machines, and then ight-click and select **Shut Down** from the context menu (or select **Action > Shut Down**).
4. Power on the vSRX instance in the Hyper-V Manager by selecting the vSRX VM from the list of virtual machines, and then right-click and select **Start** from the context menu (or select **Action > Start**).

   NOTE: If you are using Microsoft PowerShell, use the `Start-VM` command to start the vSRX VM.

CHAPTER 12

Configure vSRX Chassis Clusters

IN THIS CHAPTER

Configure a vSRX Chassis Cluster in Junos OS

IN THIS SECTION

Chassis Cluster Overview

Prerequisites

Ensure that your vSRX instances comply with the following prerequisites before you enable chassis clustering:

- Use `show version` in Junos OS to ensure that both vSRX instances have the same software version.
- Use `show system license` in Junos OS to ensure that both vSRX instances have the same licenses installed.

Chassis cluster groups a pair of the same kind of vSRX instances into a cluster to provide network node redundancy. The devices must be running the same Junos OS release. You connect the control virtual interfaces on the respective nodes to form a *control plane* that synchronizes the configuration and Junos OS kernel state. The control link (a *virtual network* or *vSwitch*) facilitates the redundancy of interfaces

and services. Similarly, you connect the *data plane* on the respective nodes over the fabric virtual interfaces to form a unified data plane. The fabric link (a virtual network or vSwitch) allows for the management of cross-node flow processing and for the management of session redundancy.

The control plane software operates in active/passive mode. When configured as a chassis cluster, one node acts as the primary device and the other as the secondary device to ensure stateful failover of processes and services in the event of a system or hardware failure on the primary device. If the primary device fails, the secondary device takes over processing of control plane traffic.

NOTE: If you configure a chassis cluster on vSRX nodes across two physical hosts, disable igmp-snooping on the bridge that each host physical interface belongs to that the control vNICs use. This ensures that the control link heartbeat is received by both nodes in the chassis cluster.

The chassis cluster data plane operates in active/active mode. In a chassis cluster, the data plane updates session information as traffic traverses either device, and it transmits information between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, traffic can enter the cluster on one node and exit from the other node.

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple *Packet Forwarding Engine*s. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.
- Support for generic routing encapsulation (*GRE*) and IP-over-IP (IP-IP) tunnels used to route encapsulated IPv4 or *IPv6* traffic by means of two internal interfaces, gr-0/0/0 and ip-0/0/0, respectively. Junos OS creates these interfaces at system startup and uses these interfaces only for processing GRE and IP-IP tunnels.

At any given instant, a cluster node can be in one of the following states: hold, primary, secondary-hold, secondary, ineligible, or disabled. Multiple event types, such as interface monitoring, Services Processing Unit (SPU) monitoring, failures, and manual failovers, can trigger a state transition.

Enable Chassis Cluster Formation

You create two vSRX instances to form a chassis cluster, and then you set the cluster ID and node ID on each instance to join the cluster. When a vSRX VM joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

You can deploy up to 255 chassis clusters in a *Layer 2* domain. Clusters and nodes are identified in the following ways:

- The *cluster ID* (a number from 1 to 255) identifies the cluster.
- The *node ID* (a number from 0 to 1) identifies the cluster node.

On SRX Series devices, the cluster ID and node ID are written into EEPROM. On the vSRX VM, vSRX stores and reads the IDs from **boot/loader.conf** and uses the IDs to initialize the chassis cluster during startup.

The chassis cluster formation commands for node 0 and node 1 are as follows:

- On vSRX node 0:

```
user@vsrx0>set chassis cluster cluster-id number node 0 reboot
```

- On vSRX node 1:

```
user@vsrx1>set chassis cluster cluster-id number node 1 reboot
```

The vSRX interface naming and mapping to vNICs changes when you enable chassis clustering. Use the same cluster ID number for each node in the cluster.

NOTE: When using multiple clusters that are connected to the same L2 domain, a unique cluster-id needs to be used for each cluster. Otherwise you may get duplicate mac addresses on the network, because the cluster-id is used to form the virtual interface mac addresses.

After reboot, on node 0, configure the fabric (data) ports of the cluster that are used to pass real-time objects (RTOs):

- ```
  user@vsrx0# set interfaces fab0 fabric-options member-interfaces ge-0/0/0
  user@vsrx0# set interfaces fab1 fabric-options member-interfaces ge-7/0/0
  ```

Chassis Cluster Quick Setup with J-Web

To configure chassis cluster from *J-Web*:

1. Enter the vSRX node 0 interface IP address in a Web browser.
2. Enter the vSRX username and password, and click **Log In**. The J-Web dashboard appears.
3. Click **Configuration Wizards>Chassis Cluster** from the left panel. The Chassis Cluster Setup wizard appears. Follow the steps in the setup wizard to configure the cluster ID and the two nodes in the cluster, and to verify connectivity.
  ```

NOTE: Use the built-in Help icon in J-Web for further details on the Chassis Cluster Setup wizard.

Manually Configure a Chassis Cluster with J-Web

You can use the *J-Web* interface to configure the primary node 0 vSRX instance in the cluster. Once you have set the cluster and node IDs and rebooted each vSRX, the following configuration will automatically be synced to the secondary node 1 vSRX instance.

Select **Configure>Chassis Cluster>Cluster Configuration**. The Chassis Cluster configuration page appears.

Table 29 on page 147 explains the contents of the HA Cluster Settings tab.

Table 30 on page 149 explains how to edit the Node Settings tab.

Table 31 on page 149 explains how to add or edit the HA Cluster Interfaces table.

Table 32 on page 151 explains how to add or edit the HA Cluster Redundancy Groups table.

Table 43: Chassis Cluster Configuration Page

Field	Function
Node Settings	
Node ID	Displays the node ID.
Cluster ID	Displays the cluster ID configured for the node.
Host Name	Displays the name of the node.
Backup Router	Displays the router used as a gateway while the Routing Engine is in secondary state for redundancy-group 0 in a chassis cluster.
Management Interface	Displays the management interface of the node.
IP Address	Displays the management IP address of the node.

Table 43: Chassis Cluster Configuration Page *(Continued)*

Field	Function
Status	Displays the state of the redundancy group. • **Primary**–Redundancy group is active. • **Secondary**–Redundancy group is passive.
Chassis Cluster>HA Cluster Settings>Interfaces	
Name	Displays the physical interface name.
Member Interfaces/IP Address	Displays the member interface name or IP address configured for an interface.
Redundancy Group	Displays the redundancy group.
Chassis Cluster>HA Cluster Settings>Redundancy Group	
Group	Displays the redundancy group identification number.
Preempt	Displays the selected preempt option. • **True**–Primary Role can be preempted based on priority. • **False**–Primary Role cannot be preempted based on priority.
Gratuitous ARP Count	Displays the number of gratuitous Address Resolution Protocol (*ARP*) requests that a newly elected primary device in a chassis cluster sends out to announce its presence to the other network devices.
Node Priority	Displays the assigned priority for the redundancy group on that node. The eligible node with the highest priority is elected as primary for the redundant group.

Table 44: Edit Node Setting Configuration Details

Field	Function	Action
Node Settings		
Host Name	Specifies the name of the host.	Enter the name of the host.
Backup Router	Displays the device used as a gateway while the Routing Engine is in the secondary state for redundancy-group 0 in a chassis cluster.	Enter the IP address of the backup router.
Destination		
IP	Adds the destination address.	Click **Add**.
Delete	Deletes the destination address.	Click **Delete**.
Interface		
Interface	Specifies the interfaces available for the router. **NOTE**: Allows you to add and edit two interfaces for each fabric link.	Select an option.
IP	Specifies the interface IP address.	Enter the interface IP address.
Add	Adds the interface.	Click **Add**.
Delete	Deletes the interface.	Click **Delete**.

Table 45: Add HA Cluster Interface Configuration Details

Field	Function	Action
Fabric Link > Fabric Link 0 (fab0)		

Table 45: Add HA Cluster Interface Configuration Details *(Continued)*

Field	Function	Action
Interface	Specifies fabric link 0.	Enter the interface IP fabric link 0.
Add	Adds fabric interface 0.	Click **Add**.
Delete	Deletes fabric interface 0.	Click **Delete**.
Fabric Link > Fabric Link 1 (fab1)		
Interface	Specifies fabric link 1.	Enter the interface IP for fabric link 1.
Add	Adds fabric interface 1.	Click **Add**.
Delete	Deletes fabric interface 1.	Click **Delete**.
Redundant Ethernet		
Interface	Specifies a logical interface consisting of two physical Ethernet interfaces, one on each chassis.	Enter the logical interface.
IP	Specifies a redundant Ethernet IP address.	Enter a redundant Ethernet IP address.
Redundancy Group	Specifies the redundancy group ID number in the chassis cluster.	Select a redundancy group from the list.
Add	Adds a redundant Ethernet IP address.	Click **Add**.
Delete	Deletes a redundant Ethernet IP address.	Click **Delete**.

Table 46: Add Redundancy Groups Configuration Details

Field	Function	Action
Redundancy Group	Specifies the redundancy group name.	Enter the redundancy group name.
Allow preemption of primaryship	Allows a node with a better priority to initiate a failover for a redundancy group. **NOTE**: By default, this feature is disabled. When disabled, a node with a better priority does not initiate a redundancy group failover (unless some other factor, such as faulty network connectivity identified for monitored interfaces, causes a failover).	–
Gratuitous ARP Count	Specifies the number of gratuitous Address Resolution Protocol requests that a newly elected primary sends out on the active redundant Ethernet interface child links to notify network devices of a change in primary role on the redundant Ethernet interface links.	Enter a value from 1 to 16. The default is 4.
node0 priority	Specifies the priority value of node0 for a redundancy group.	Enter the node priority number as 0.
node1 priority	Specifies the priority value of node1 for a redundancy group.	Select the node priority number as 1.
Interface Monitor		
Interface	Specifies the number of redundant Ethernet interfaces to be created for the cluster.	Select an interface from the list.
Weight	Specifies the weight for the interface to be monitored.	Enter a value from 1 to 125.
Add	Adds interfaces to be monitored by the redundancy group along with their respective weights.	Click **Add**.

Table 46: Add Redundancy Groups Configuration Details *(Continued)*

Field	Function	Action
Delete	Deletes interfaces to be monitored by the redundancy group along with their respective weights.	Select the interface from the configured list and click **Delete**.
IP Monitoring		
Weight	Specifies the global weight for IP monitoring.	Enter a value from 0 to 255.
Threshold	Specifies the global threshold for IP monitoring.	Enter a value from 0 to 255.
Retry Count	Specifies the number of retries needed to declare reachability failure.	Enter a value from 5 to 15.
Retry Interval	Specifies the time interval in seconds between retries.	Enter a value from 1 to 30.
IPV4 Addresses to Be Monitored		
IP	Specifies the IPv4 addresses to be monitored for reachability.	Enter the IPv4 addresses.
Weight	Specifies the weight for the redundancy group interface to be monitored.	Enter the weight.
Interface	Specifies the logical interface through which to monitor this IP address.	Enter the logical interface address.
Secondary IP address	Specifies the source address for monitoring packets on a secondary link.	Enter the secondary IP address.
Add	Adds the IPv4 address to be monitored.	Click **Add**.
Delete	Deletes the IPv4 address to be monitored.	Select the IPv4 address from the list and click **Delete**.

SEE ALSO

Chassis Cluster Feature Guide for Security Devices

vSRX Cluster Staging and Provisioning in Hyper-V

IN THIS SECTION

Staging and provisioning a vSRX cluster on a Hyper-V host computer includes the following tasks:

NOTE: Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, support for chassis clustering to provide network node redundancy is only available on Windows Hyper-V Server 2016.

Deploying the VMs and Additional Network Adapters in Hyper-V

The vSRX cluster uses three interfaces exclusively for clustering (the first two are predefined):

- Out-of-band management interface (fxp0).
- Cluster control link (em0).
- Cluster fabric links (fab0 and fab1). For example, you can specify ge-0/0/0 as fab0 on node0 and ge-7/0/0 as fab1 on node1.

A cluster requires three interfaces (two for the cluster and one for management) and additional interfaces to forward data. This section outlines how to create the control link and fabric link connections, and to map all data interfaces to network adapters.

NOTE: For an overview on the procedure to add virtual switches and map the virtual switch to a network adapter, see *Add vSRX Interfaces*

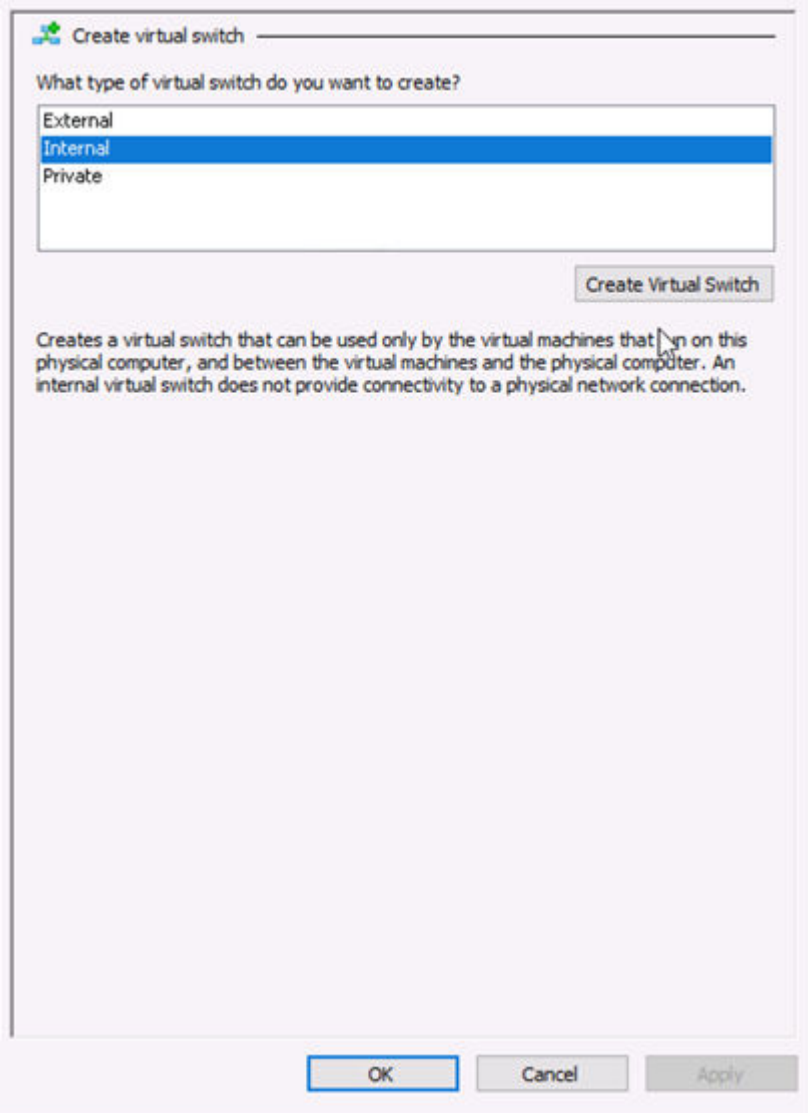
Creating the Control Link Connection in Hyper-V

To connect the control interface through the control link virtual switch using Hyper-V Manager:

1. Open the Hyper-V Manager by selecting **Start > Administrative Tools > Hyper-V Manager**.
2. From the Hyper-V Manager, select **Action > Virtual Switch Manager**. The Virtual Switch Manager appears.

3. Under the Virtual Switches section, select **New virtual network switch**. The Create Virtual Switch pane appears (see Figure 43 on page 222).

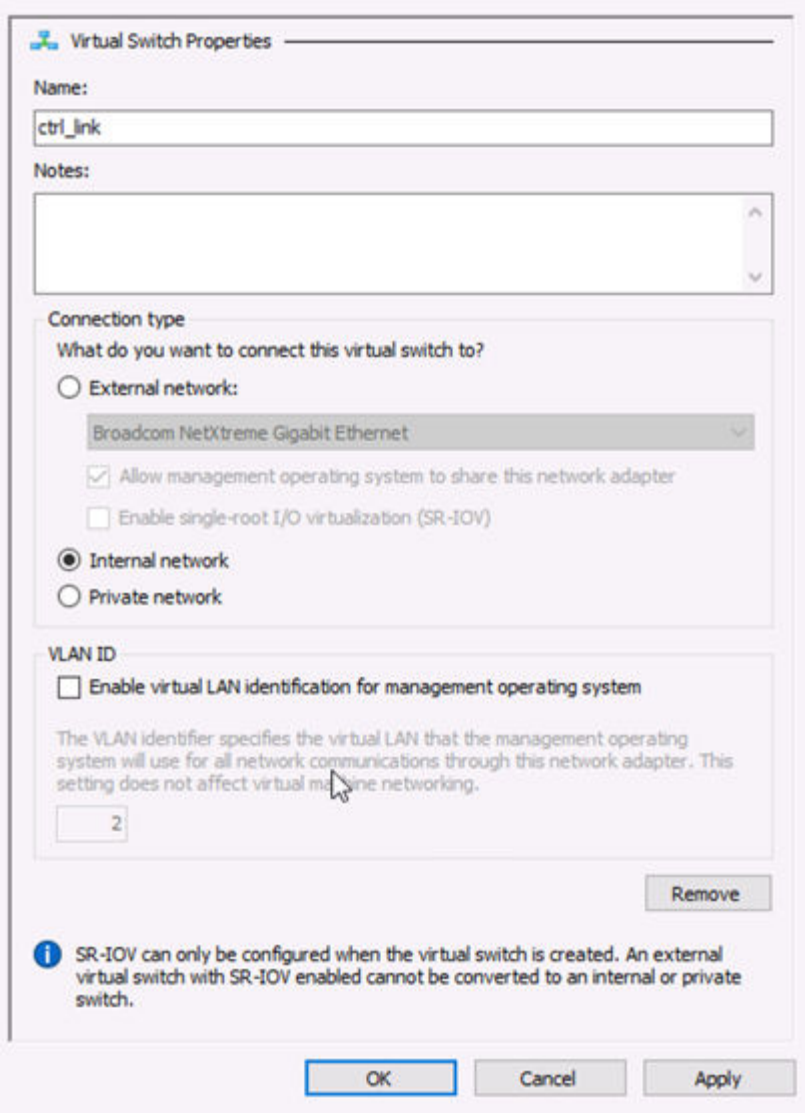
Figure 43: Create Virtual Switch Pane



4. Select **Internal** as the type of virtual switch. Internal allows communication between virtual machines on the same Hyper-V server, and between the virtual machines and the management host operating system.

5. Select **Create Virtual Switch**. The Virtual Switch Properties page appears (see Figure 44 on page 223).

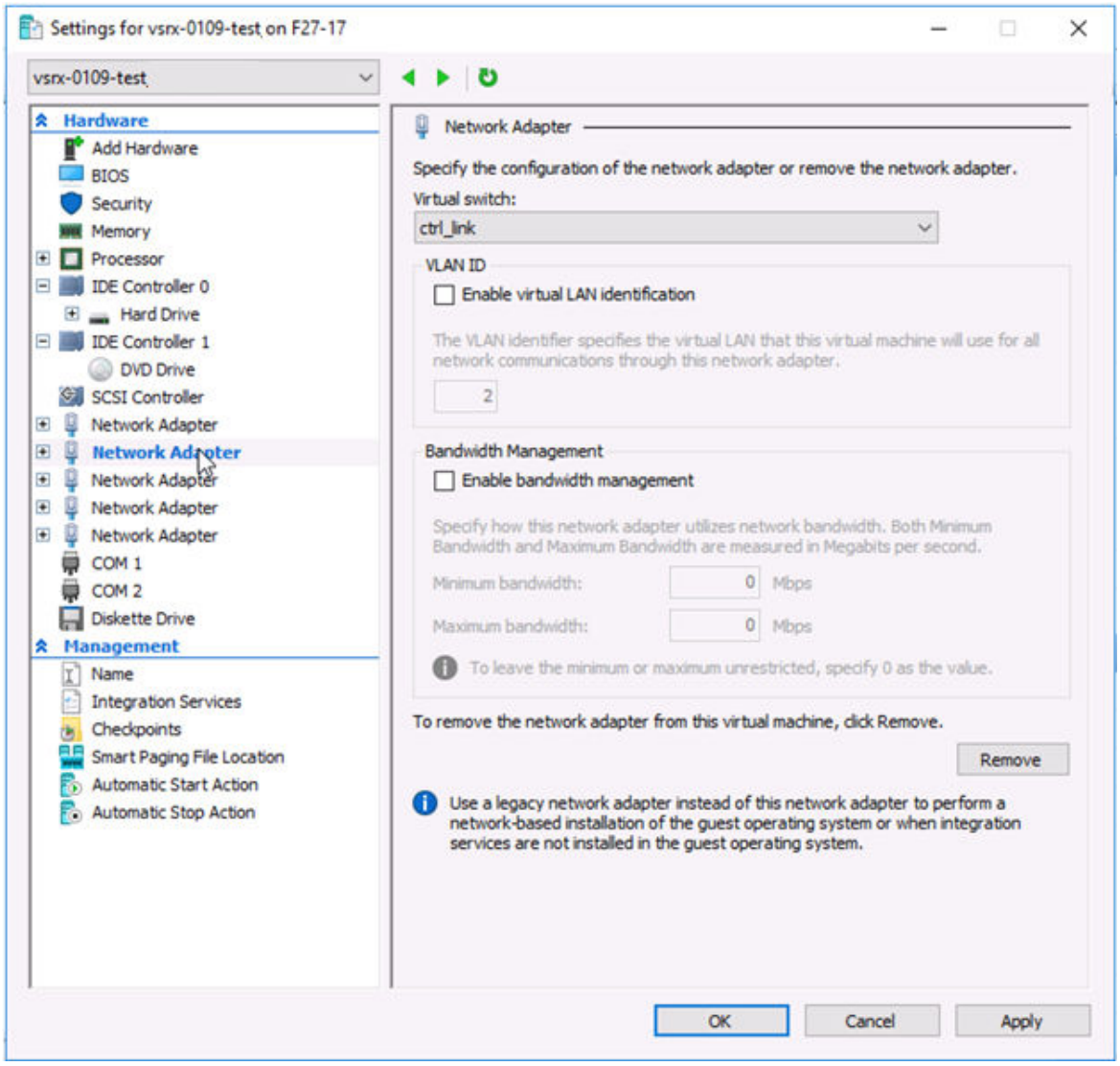
Figure 44: Virtual Switch Properties Pane



6. Specify a name for the control link virtual switch. Leave the other virtual switch properties at their default settings.
7. Click **OK** and then click **Yes** to apply networking changes and to close the Virtual Switch Manager window.

8. Right-click the vSRX VM and select **Settings** from the context menu. From the Settings dialog for the vSRX VM, the Hardware section, click **Network Adapter**. The Network Adapter pane appears (see Figure 45 on page 224). Assign network adapter 2 as the control link (em0) virtual switch.

Figure 45: Adding Virtual Switch to Network Adapter Pane Example



9. From the Virtual switch drop-down assign **ctrl_link** to the control link virtual switch.
10. From the Network Adapter pane, select **Advanced Features**. Select the **Enable MAC address spoofing** check box to enable the MAC address spoofing function for the network adapter. MAC address spoofing is a requirement for the control link interface included in the redundancy groups.
11. Click **OK** and then click **Yes** to apply network adapter changes.

Creating the Fabric Link Connection in Hyper-V

To connect the fabric interface through the fabric link virtual switch using Hyper-V Manager

1. If necessary, open the Hyper-V Manager by selecting **Start > Administrative Tools > Hyper-V Manager**.
2. From the Hyper-V Manager, select **Action > Virtual Switch Manager**. The Virtual Switch Manager appears.
3. Under the Virtual Switches section, select **New virtual network switch**. The Create Virtual Switch pane appears (see Figure 43 on page 222).
4. Select **Internal** as the type of virtual switch. Internal allows communication between virtual machines on the same Hyper-V server, and between the virtual machines and the management host operating system.
5. Select **Create Virtual Switch**. The Virtual Switch Properties page appears (see Figure 44 on page 223).
6. Specify a name for the fabric link virtual switch. Leave the other virtual switch properties at their default settings.
7. Click **OK** and then click **Yes** to apply networking changes and to close the Virtual Switch Manager window.
8. Right-click the vSRX VM and select **Settings** from the context menu. From the Settings dialog for the vSRX VM, the Hardware section, click **Network Adapter** to access the Network Adapter pane. The Network Adapter pane appears (see Figure 45 on page 224). Assign network adapter 3 as the fabric link (fab 0 or fab 1) virtual switch.
9. From the Virtual switch drop-down assign **fab0 or fab1** to the fabric link virtual switch.
10. From the Network Adapter pane, select **Advanced Features**. Select the **Enable MAC address spoofing** check box to enable the MAC address spoofing function for the network adapter. MAC address spoofing is a requirement for the fabric link interface included in the redundancy groups.
11. Click **OK** and then click **Yes** to apply network adapter changes.

Creating the Data Interfaces Using Hyper-V

To map all data interfaces to the desired network adapters:

1. If necessary, open the Hyper-V Manager by selecting **Start > Administrative Tools > Hyper-V Manager**.
2. From the Hyper-V Manager, select **Action > Virtual Switch Manager**. The Virtual Switch Manager appears.
3. Under the Virtual Switches section, select **New virtual network switch**. The Create Virtual Switch pane appears (see Figure 43 on page 222).
4. Select **Internal** as the type of virtual switch. Internal allows communication between virtual machines on the same Hyper-V server, and between the virtual machines and the management host operating system.

5. Select **Create Virtual Switch**. The Virtual Switch Properties page appears (see Figure 44 on page 223).
6. Specify a name for the data interface virtual switch. Leave the other virtual switch properties at their default settings.
7. Click **OK** and then click **Yes** to apply networking changes and to close the Virtual Switch Manager window.
8. Right-click the vSRX VM and select **Settings** from the context menu. From the Settings dialog for the vSRX VM, the Hardware section, click **Network Adapter** to access the Network Adapter pane. The Network Adapter pane appears (see Figure 45 on page 224). Assign network adapter 3 as the data interface (fab 0 or fab 1) virtual switch.
9. From the Virtual switch drop-down assign **data interface** to the virtual switch.
10. From the Network Adapter pane, select **Advanced Features**. Select the **Enable MAC address spoofing** check box to enable the MAC address spoofing function for the network adapter. MAC address spoofing is a requirement for the data interfaces included in the redundancy groups.
11. Click **OK** and then click **Yes** to apply network adapter changes. The data interface will be connected through the data virtual switch.

Prestaging the Configuration from the Console

The following procedure explains the configuration commands required to set up the vSRX chassis cluster. The procedure powers up both nodes, adds the configuration to the cluster, and allows SSH remote access.

1. Log in as the root user. There is no password.
2. Start the CLI.

```
root#cli
root@>
```

3. Enter configuration mode.

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

4. Copy the following commands and paste them into the CLI:

```
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.42.81/24
set groups node0 system hostname vsrx-node0
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.42.82/24
```

```
set groups node1 system hostname vsrx-node1
set apply-groups "${node}"
```

5. Set the root authentication password by entering a cleartext password, an encrypted password, or an SSH public key string (DSA or RSA).

```
root@# set system root-authentication plain-text-password
New password: password
Retype new password: password
set system root-authentication encrypted-password "$ABC123"
```

6. To enable SSH remote access:

```
user@host#set system services ssh
```

7. To enable IPv6:

```
user@host#set security forwarding-options family inet6 mode flow-based
```

   This step is optional and requires a system reboot.

8. Commit the configuration to activate it on the device.

```
user@host#commit
commit complete
```

9. When you have finished configuring the device, exit configuration mode.

```
user@host#exit
```

Connecting and Installing the Staging Configuration

After the vSRX cluster initial setup, set the cluster ID and the node ID, as described in *Configure a vSRX Chassis Cluster in Junos OS*.

After reboot, the two nodes are reachable on interface fxp0 with SSH. If the configuration is operational, the `show chassis cluster status` command displays output similar to that shown in the following sample output.

```
vsrx> show chassis cluster status
```

```
Cluster ID: 1
Node                  Priority          Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
    node0                   100         secondary      no       no
    node1                   150         primary        no       no

Redundancy group: 1 , Failover count: 1
    node0                   100         secondary      no       no
    node1                   150         primary        no       no
```

A cluster is healthy when the primary and secondary nodes are present and both have a priority greater than 0.

Release History Table

Release	Description
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, support for chassis clustering to provide network node redundancy is only available on Windows Hyper-V Server 2016.

4

PART

vSRX Deployment for Contrail

CHAPTER 13

Overview of vSRX Service Chains in Contrail

IN THIS CHAPTER

Understand vSRX with Contrail

IN THIS SECTION

This section presents an overview of vSRX on Contrail

vSRX on Juniper Networks Contrail

Juniper Networks Contrail is an open, standards-based software solution that delivers network *virtualization* and service automation for federated cloud networks. It provides self-service provisioning, improves network troubleshooting and diagnostics, and enables service chaining for dynamic application environments across enterprise virtual private cloud (VPC), managed Infrastructure as a Service (IaaS), and Networks Functions Virtualization (*NFV*) use cases.

You can use Contrail with open cloud orchestration systems such as OpenStack or CloudStack to instantiate instances of vSRX in a virtual environment. Contrail with vSRX provides network services such as *firewall*, *NAT*, and load balancing to virtual networks.

NOTE: vSRX on a *KVM hypervisor* requires you to enable hardware-based virtualization on a host OS that contains an Intel Virtualization Technology (VT) capable processor.

vSRX Scale Up Performance

Table 47 on page 231 shows the vSRX scale up performance based on the number of vCPUs and vRAM applied to a vSRX VM along with the Junos OS release in which a particular vSRX software specification was introduced.

Table 47: vSRX Scale Up Performance

vCPUs	vRAM	NICs	Release Introduced
2 vCPUs	4 GB	• Virtio • SR-IOV (Intel X520/X540)	Junos OS Release 15.1X49-D20
5 vCPUs	8 GB	• Virtio • SR-IOV (Intel X520/X540)	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1

You can scale the performance and capacity of a vSRX instance by increasing the number of vCPUs and the amount of vRAM allocated to the vSRX. The multi-core vSRX automatically selects the appropriate vCPUs and vRAM values at boot time, as well as the number of Receive Side Scaling (RSS) queues in the NIC. If the vCPU and vRAM settings allocated to a vSRX VM do not match what is currently available, the vSRX scales down to the closest supported value for the instance. For example, if a vSRX VM has 3 vCPUs and 8 GB of vRAM, vSRX boots to the smaller vCPU size, which requires a minimum of 2 vCPUs. You can scale up a vSRX instance to a higher number of vCPUs and amount of vRAM, but you cannot scale down an existing vSRX instance to a smaller setting.

NOTE: The number of RSS queues typically matches with the number of data plane vCPUs of a vSRX instance. For example, a vSRX with 4 data plane vCPUs should have 4 RSS queues.

RELATED DOCUMENTATION

Contrail Overview

Requirements for vSRX on Contrail

IN THIS SECTION

Software Requirements

Table 48 on page 232 lists the system software requirement specifications when deploying vSRX on Juniper Networks Contrail. The table outlines the Junos OS release in which a particular software specification for deploying vSRX on KVM was introduced. You will need to download a specific Junos OS release to take advantage of certain features.

Table 48: Specifications for vSRX on Juniper Networks Contrail

Component	Specification	Junos OS Release Introduced
Hypervisor support	Linux KVM	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1
Memory	4 GB	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1
	8 GB	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1
Disk space	20 GB IDE drive	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1

Table 48: Specifications for vSRX on Juniper Networks Contrail *(Continued)*

Component	Specification	Junos OS Release Introduced
vCPUs	2 vCPUs **NOTE**: The Contrail compute node must bare metal since vSRX as a VNF does not support nested virtualization.	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1
	5 vCPUs **NOTE**: The Contrail compute node must bare metal since vSRX as a VNF does not support nested virtualization.	Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1
vNICs	Up to 16 vNICs • Virtio • SR-IOV **NOTE**: We recommend the Intel X520/X540 physical NICs for SR-IOV support on vSRX. For SR-IOV limitations, see the *Known Behavior* section of the *vSRX Release Notes*.	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1

Table 49 on page 233 lists the software specifications on the vSRX.

Table 49: Software Specifications for vSRX 3.0 on Juniper Networks Contrail

Flavor Name	vCPU	Junos OS Release Introduced
Hypervisor support	Linux KVM	Junos OS Release 18.2R1 or later release
Memory	4 GB	Junos OS Release 18.2R1 or later release
	8 GB	Junos OS Release 18.2R1 or later release
Disk space	20 GB IDE drive	Junos OS Release 18.2R1 or later release

Table 49: Software Specifications for vSRX 3.0 on Juniper Networks Contrail *(Continued)*

Flavor Name	vCPU	Junos OS Release Introduced
vCPUs	2 vCPUs	Junos OS Release 18.2R1 or later release
	5 vCPUs	Junos OS Release 18.2R1 or later release
vNICs	Up to 16 vNICs • Virtio • SR-IOV **NOTE**: We recommend the Intel X520 physical NICs for SR-IOV support on small flavor vSRX, Intel X710 for Medium flavor vSRX.	Junos OS Release 18.2R1 or later release

Contrail Recommendations for vSRX

Table 50 on page 234 lists the recommended software versions to run vSRX on Contrail.

Table 50: Contrail Recommendations for vSRX

Software	Version	Supported Release
Contrail	2.20	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1 or later release
	3.1	Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1 or later release
	3.5	Junos OS Release 18.4R1
OpenStack	Juno or Icehouse	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1 or later release

Table 50: Contrail Recommendations for vSRX *(Continued)*

Software	Version	Supported Release
	Juno or Kilo	Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1 or later release
Host OS	Ubuntu 14.04.2	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1 or later release
Linux Kernel	3.16	Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1 or later release

NOTE: We recommend that you enable hardware-based virtualization on the host machine. You can verify CPU compatibility here: http://www.linux-kvm.org/page/Processor_support. See Contrail - Server Requirements to review any additional requirements for Contrail.

Table 51 on page 235 lists the contrail recommendations for vSRX.

Table 51: Contrail Recommendations for vSRX 3.0

Software	Version	Supported Release
Contrail	3.1	Junos OS Release 18.2R1 or later release
	3.2	Junos OS Release 18.2R1 or later release
	5.X	Junos OS Release 19.3R1 or later release
OpenStack	Centos 7 or 8	Junos OS Release 18.2R1 or later release
Host OS	Ubuntu 14.04.2	Junos OS Release 18.2R1 or later release

Table 51: Contrail Recommendations for vSRX 3.0 *(Continued)*

Software	Version	Supported Release
Linux Kernel	Queens or later	Junos OS Release 18.2R1 or later release

Hardware Recommendations

Table 52 on page 236 lists the hardware specifications for the host machine that runs the vSRX VM.

Table 52: Hardware Specifications for the Host Machine

Component	Specification
Host memory size	4 GB (minimum) .
Host processor type	Intel x86_64 multicore CPU **NOTE**: DPDK requires Intel Virtualization VT-x/VT-d support in the CPU. See *About Intel Virtualization Technology*.
Virtual network adapter	VMXNet3 device or VMWare Virtual NIC **NOTE**: Virtual Machine Communication Interface (VMCI) communication channel is internal to the ESXi hypervisor and the vSRX VM.

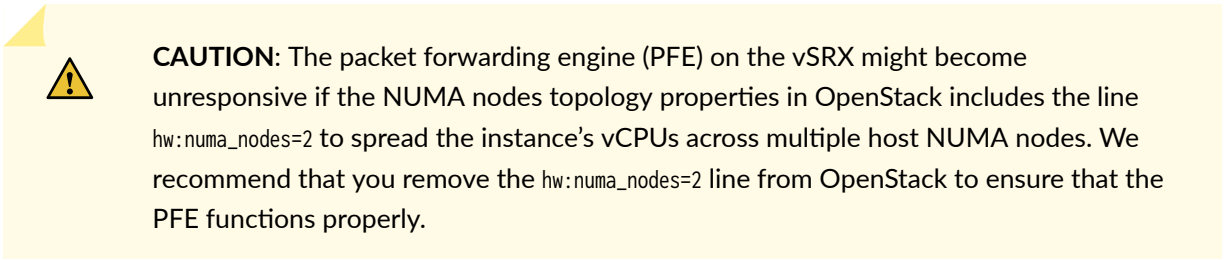
Best Practices for Improving vSRX Performance

Review the following practices to improve vSRX performance.

NUMA Nodes

The x86 server architecture consists of multiple sockets and multiple cores within a socket. Each socket also has memory that is used to store packets during I/O transfers from the NIC to the host. To efficiently read packets from memory, guest applications and associated peripherals (such as the NIC) should reside within a single socket. A penalty is associated with spanning CPU sockets for memory accesses, which might result in nondeterministic performance. For vSRX, we recommend that all vCPUs

for the vSRX VM are in the same physical non-uniform memory access (NUMA) node for optimal performance.



CAUTION: The packet forwarding engine (PFE) on the vSRX might become unresponsive if the NUMA nodes topology properties in OpenStack includes the line `hw:numa_nodes=2` to spread the instance's vCPUs across multiple host NUMA nodes. We recommend that you remove the `hw:numa_nodes=2` line from OpenStack to ensure that the PFE functions properly.

PCI NIC-to-VM Mapping

If the node on which vSRX is running is different from the node to which the Intel PCI NIC is connected, then packets will have to traverse an additional hop in the QPI link, and this will reduce overall throughput. On a Linux host OS, install the `hwloc` package and use the `lstopo` command to view information about relative physical NIC locations. On some servers where this information is not available, refer to the hardware documentation for the slot-to-NUMA node topology.

Mapping Virtual Interfaces to a vSRX VM

To determine which virtual interfaces on your Linux host OS map to a vSRX VM:

1. Use the `virsh list` command on your Linux host OS to list the running VMs.

```
hostOS# virsh list
```

```
 Id    Name                           State
----------------------------------------------------
 25    instance-00000060              running
 31    instance-0000005b              running
 34    instance-000000bd              running
 35    instance-000000bc              running
```

2. Use the `virsh domiflist` *vsrx-name* command to list the virtual interfaces on that vSRX VM.

```
hostOS# virsh domiflist 31
```

```
Interface  Type       Source     Model       MAC
-------------------------------------------------------
tapd3d9639c-d5 ethernet   -          virtio      02:d3:d9:63:9c:d5
tapc3c3751a-37 ethernet   -          virtio      02:c3:c3:75:1a:37
tap8af29333-1b ethernet   -          virtio      02:8a:f2:93:33:1b
tapf0387bee-9b ethernet   -          virtio      02:f0:38:7b:ee:9b
tap04e4b59a-91 ethernet   -          virtio      02:04:e4:b5:9a:91
```

NOTE: The first virtual interface maps to the fxp0 interface in Junos OS.

Interface Mapping for vSRX on Contrail

Each network adapter defined for a vSRX is mapped to a specific interface, depending on whether the vSRX instance is a standalone VM or one of a cluster pair for high availability. The interface names and mappings in vSRX are shown in Table 53 on page 239 and Table 54 on page 239.

Note the following:

- In standalone mode:
  - fxp0 is the out-of-band management interface.
  - ge-0/0/0 is the first traffic (revenue) interface.
- In cluster mode:
  - fxp0 is the out-of-band management interface.
  - em0 is the cluster control link for both nodes.
  - Any of the traffic interfaces can be specified as the fabric links, such as ge-0/0/0 for fab0 on node 0 and ge-7/0/0 for fab1 on node 1.

Table 53 on page 239 shows the interface names and mappings for a standalone vSRX VM.

Table 53: Interface Names for a Standalone vSRX VM

Network Adapter	Interface Name in Junos OS for vSRX
1	fxp0
2	ge-0/0/0
3	ge-0/0/1
4	ge-0/0/2
5	ge-0/0/3
6	ge-0/0/4
7	ge-0/0/5
8	ge-0/0/6

Table 54 on page 239 shows the interface names and mappings for a pair of vSRX VMs in a cluster (node 0 and node 1).

Table 54: Interface Names for a vSRX Cluster Pair

Network Adapter	Interface Name in Junos OS for vSRX
1	fxp0 (node 0 and 1)
2	em0 (node 0 and 1)
3	ge-0/0/0 (node 0) ge-7/0/0 (node 1)

Table 54: Interface Names for a vSRX Cluster Pair *(Continued)*

Network Adapter	Interface Name in Junos OS for vSRX
4	ge-0/0/1 (node 0) ge-7/0/1 (node 1)
5	ge-0/0/2 (node 0) ge-7/0/2 (node 1)
6	ge-0/0/3 (node 0) ge-7/0/3 (node 1)
7	ge-0/0/4 (node 0) ge-7/0/4 (node 1)
8	ge-0/0/5 (node 0) ge-7/0/5 (node 1)

vSRX Default Settings on Contrail

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.

Table 55 on page 240 lists the factory default settings for the vSRX security policies.

Table 55: Factory Default Settings for Security Policies

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

Table 55: Factory Default Settings for Security Policies *(Continued)*

Source Zone	Destination Zone	Policy Action
untrust	trust	deny

RELATED DOCUMENTATION

- About Intel Virtualization Technology
- PCI Devices
- DPDK Release Notes

Overview of Service Chains with vSRX

IN THIS SECTION

You can use Contrail to chain various Layer 2 through Layer 7 services such as *firewall*, *NAT*, and *IDP* through one or more vSRX VMs. For example, you can insert a vSRX firewall VM between two other virtual machines (VMs). By using vSRX and service chains, you can tailor the security needs to a targeted virtual network and VM set. This provides agility and scalability in line with the fluidity of cloud network environments.

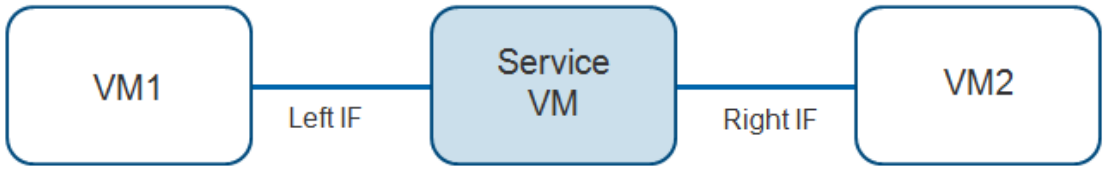
Understanding Service Chains

To create a service through vSRX, you instantiate one or more vSRX VMs to dynamically apply single or multiple services to network traffic.

Figure 46 on page 242 shows a basic service chain with a single vSRX VM. The vSRX service VM spawns a service, such as a firewall. The left interface (left IF) points to the internal end customer, who uses the

service; and the right interface (right IF) points to the external network or Internet. You can also instantiate multiple vSRX VMs to chain multiple services together. For example, you could add an IDP service after the firewall.

Figure 46: vSRX Service Chaining



When you create a service chain, Contrail creates tunnels across the underlay network that span all services in the chain.

Service Chain Modes

You can configure the following service modes:

- Transparent or bridge mode—Used for services that do not modify the packet. Also known as bump-in-the-wire or Layer 2 mode. Examples include Layer 2 firewall and IDP.
- In-network or routed mode—Provides a gateway service that routes packets between the service instance interfaces. Examples include NAT, Layer 3 firewall, and load balancing.
- In-network-nat mode—Similar to in-network mode; however, packets from the left (private) network are not routed to the right (public) source network. In-network-nat mode is particularly useful for NAT services.

> **NOTE**: Ensure that you define the service policy with the private network on the left and public on the right in order to get the public routes (usually the default) advertised into the left network.

Components of a Service Chain

Service chaining requires the following configuration components to build the chain:

- Service template
- Virtual networks
- Service instance
- Network policy

Service Templates

Service templates map out the basic configuration that Contrail uses to instantiate a service instance, or VM. Within Contrail, you configure service templates in the scope of a domain, and you can use the templates on all projects within a domain. You can use a template to launch multiple service instances of the same type in different projects within a domain. Within a service template, you select the service mode, a vSRX image name for the VM that will provide the service, and an ordered list of interfaces for the service. vSRX service VMs require the management interface to be the first interface in that ordered list. You can use OpenStack Horizon or Glance to add the vSRX image. You also select the OpenStack flavor to associate with all service instances that use the service template. An OpenStack flavor defines the number of vCPUs, storage, and memory you can assign to a VM. OpenStack includes default flavors, and you can create new flavors in the OpenStack dashboard.

Virtual Networks

Virtual networks provide the link between the service instance and the network traffic in the virtualized environment. You can create the virtual networks in Contrail or OpenStack and use those networks to direct traffic to or through the service instance.

Service Instances

A service instance is the instantiation of the selected service template to create one or more VMs that provide the service (for example, a firewall). When you create a service instance, you select a service template that defines the instance. You also associate the interfaces in the service template with the virtual networks needed to direct traffic into and out of the service instance. If you enable service scaling in the selected service template, you can instantiate more than one VM when you create the service instance.

Network Policies

By default, all traffic in a virtual network remains isolated. You configure a network policy to allow traffic between virtual networks and through the service instance. The network policy filters traffic to and from the service VM based on the rules you configure. You select the service instance VM and the virtual networks for the right and left interfaces of that VM that the network policy applies to. As a final step. you associate the network policy with each virtual network the policy applies to.

RELATED DOCUMENTATION

Contrail - Service Chaining

Spawn vSRX in a Contrail Service Chain

IN THIS SECTION

Ensure that you have installed Contrail and have loaded the vSRX images with OpenStack Horizon or Glance.

- Installation Overview (Contrail)
- Add the Image Service (glance)

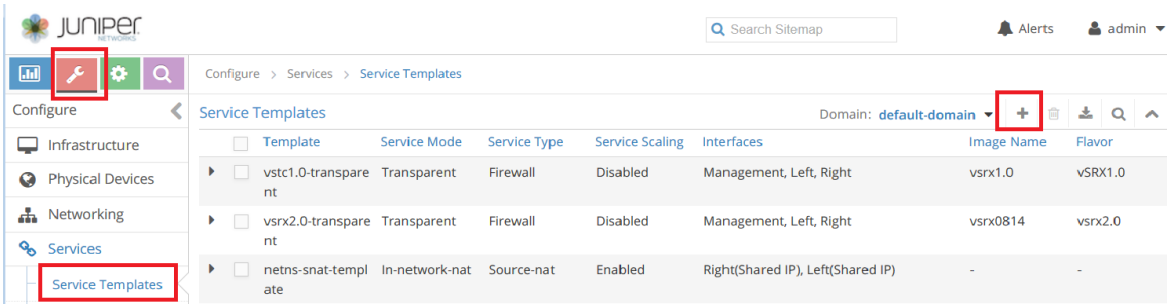
You can use Contrail to chain various Layer 2 through Layer 7 services such as firewall, NAT, and IDP through vSRX VMs.

Create a Service Template

To create a service template:

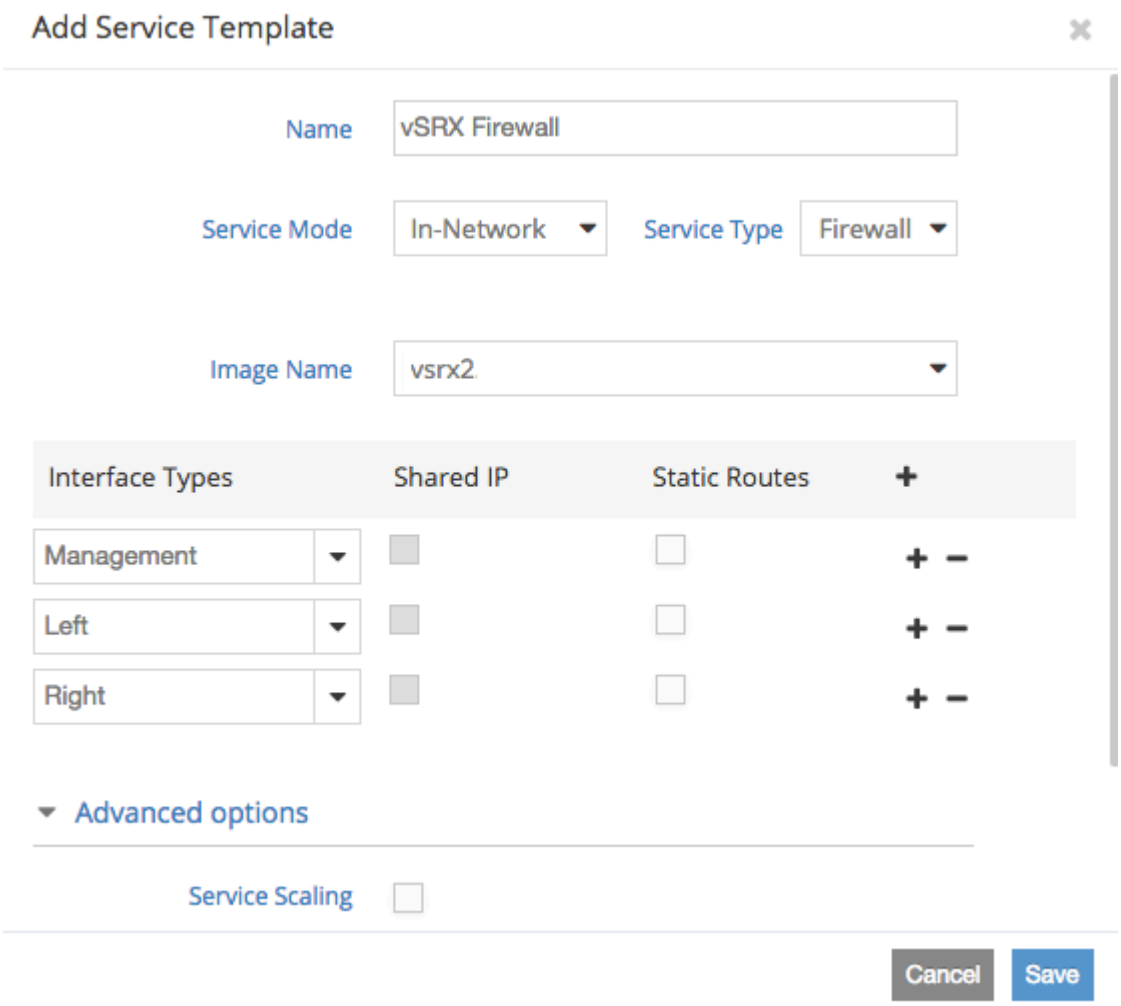
1. From Contrail, select **Configure>Services>Service Templates**. The list of existing service templates appears, as shown in Figure 47 on page 245.

Figure 47: Contrail Service Templates



2. Click **+** to create a new service template. The Add Service Template dialog box appears, as shown in Figure 48 on page 246.

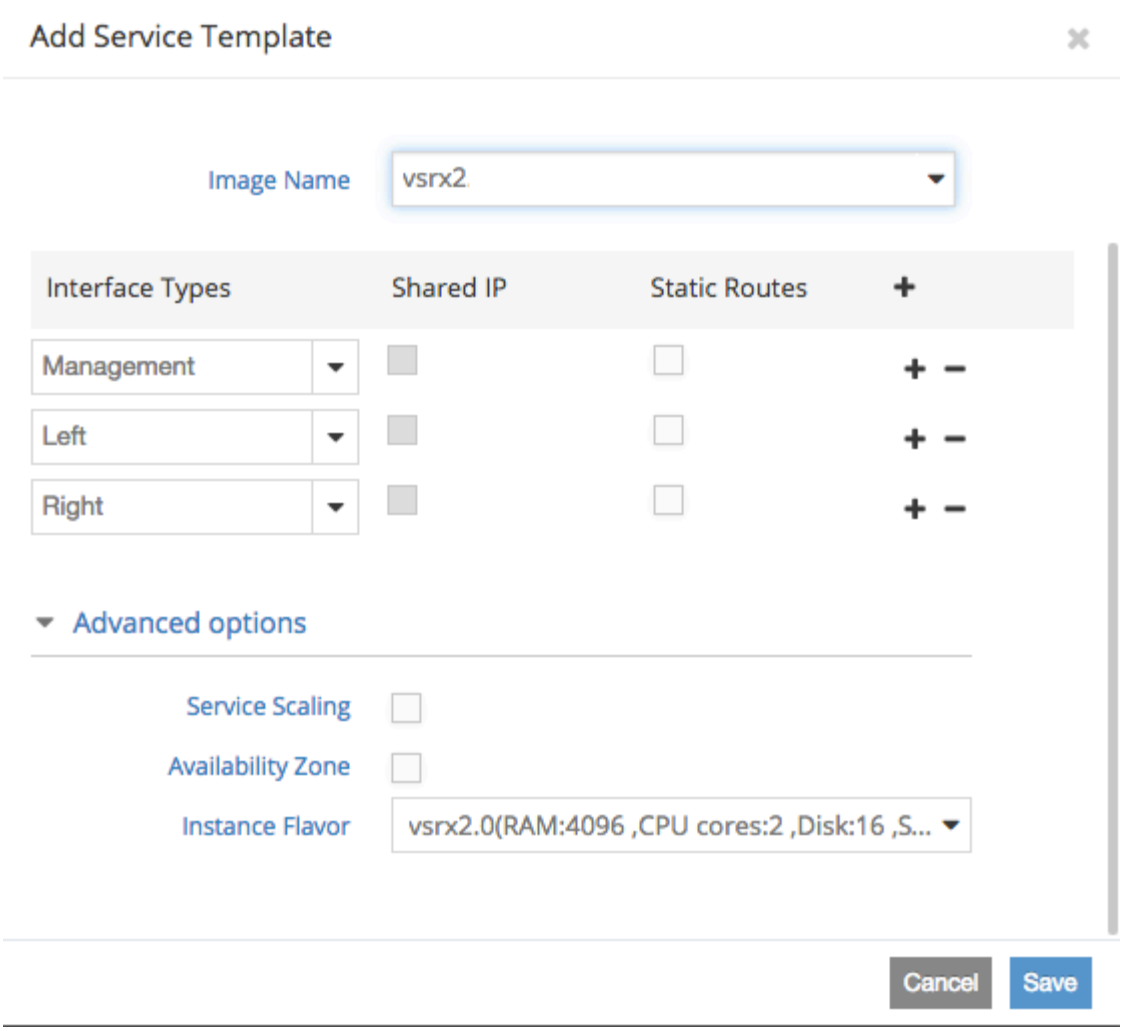
Figure 48: Contrail Add a Service Template



3. Add a name for the service template in the Name box.
4. Select the appropriate service mode and service type from the lists.
5. Select the vSRX image from the Image Name list. This is the image you installed previously in the OpenStack image service.
6. Click **+** to add three interfaces.
7. Select **Management** for the first interface type, **Left** for the second interface type, and **Right** for the third interface type. You associate the left and right interfaces with the left and right virtual networks when you create the service instance. Any additional interfaces must be of type Other.

8. Expand **Advanced Options** and select an instance flavor from the Instance Flavor list, as shown in Figure 49 on page 247. You can use an appropriate default flavor from OpenStack or a custom flavor you created previously for vSRX.

Figure 49: Advanced Options - Add Service Template



9. Optionally, check **Scaling** to create multiple identical vSRX instances from this service template for load balancing.
10. Click **Save** to create this new service template.

See Contrail - Creating an In-Network or In-Network-NAT Service Chain for more details.

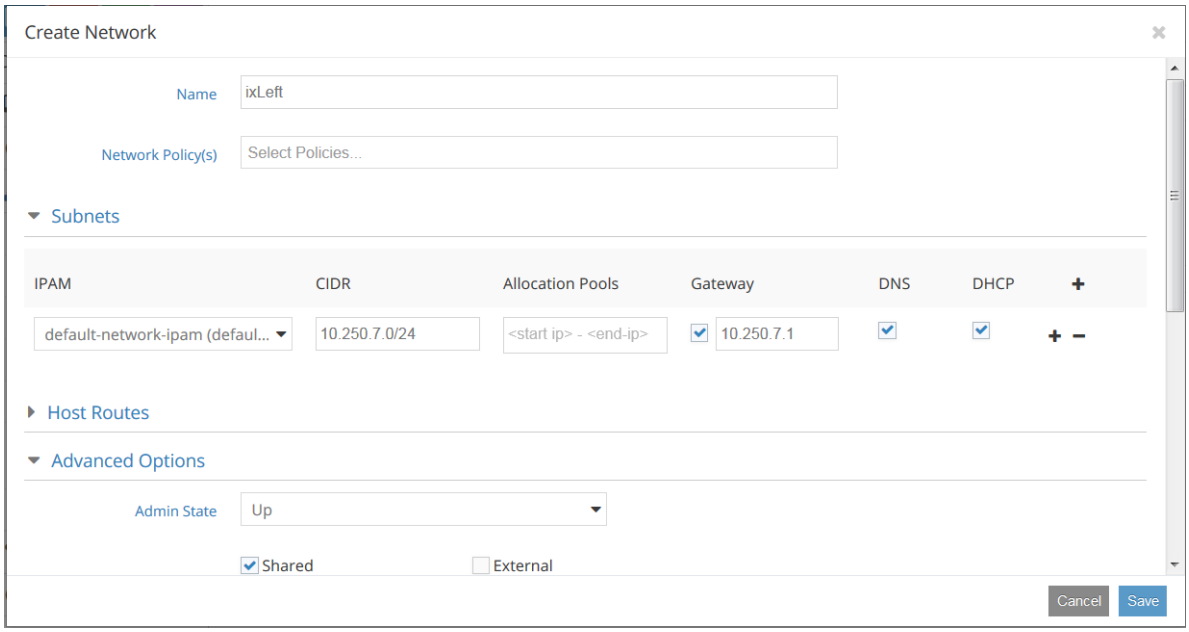
Create Left and Right Virtual Networks

Ensure that you have IP Address Management (IPAM) set up for your project.

To create a virtual network:

1. From Contrail, select **Configure>Networking>Networks**. The list of existing networks appears.
2. Verify that your project is displayed as active in the upper right Project list, and click **+** to create a new virtual network. The Create Network dialog box appears, as shown in Figure 50 on page 248

Figure 50: Creating a Virtual Network in Contrail



3. Enter a name for the left virtual network.

   Do not select a network policy yet. You create the network policy after you create the service instance and then you update this virtual network to add the policy.
4. Expand **Subnet** and click **+** to add IPAM to this virtual network.
5. Select the appropriate IPAM from the list.
6. Set the CIDR and Gateway fields.
7. Expand **Advanced Options** and select appropriate options for your network.
8. Click **Save**. The new virtual network appears in the list of configured networks.
9. Repeat this procedure for the right virtual network.

See Contrail - Creating a Virtual Network for more details

Create a vSRX Service Instance

To create a vSRX service instance:

1. Select **Configure>Services>Service Instances**. The list of existing service instances appears.

2. Click **+** to create a new service instance. The Create Service Instance dialog box appears.
3. Enter a name for the service instance.

   NOTE: Do not use white space in the service instance name.

4. Select the service template you created for vSRX from the Services Template list. This service template includes the vSRX image used to provide the service.
5. Select **Management** from the Interface 1 list. Management must be the first interface for vSRX service instances.
6. Select **Left** from the Interface 2 list, and **Right** from the Interface 3 list.
7. Select **Auto Configured** for the Management interface.
8. Select the left virtual network for the left interface, and the right virtual network for the right interface.
9. Click **Save** to save this service instance. Contrail launches the vSRX VM for this service instance.
10. Optionally, select **Configure>Services>Service Instances** to view this new vSRX instance status. You can expand the row for this instance in the table and click **View Console** to access the vSRX console port.

NOTE: You can also view this service instance from the OpenStack Instances table, but you should only use Contrail to delete service instances.

See Contrail - Creating an In-Network or In-Network-NAT Service Chain for more details.

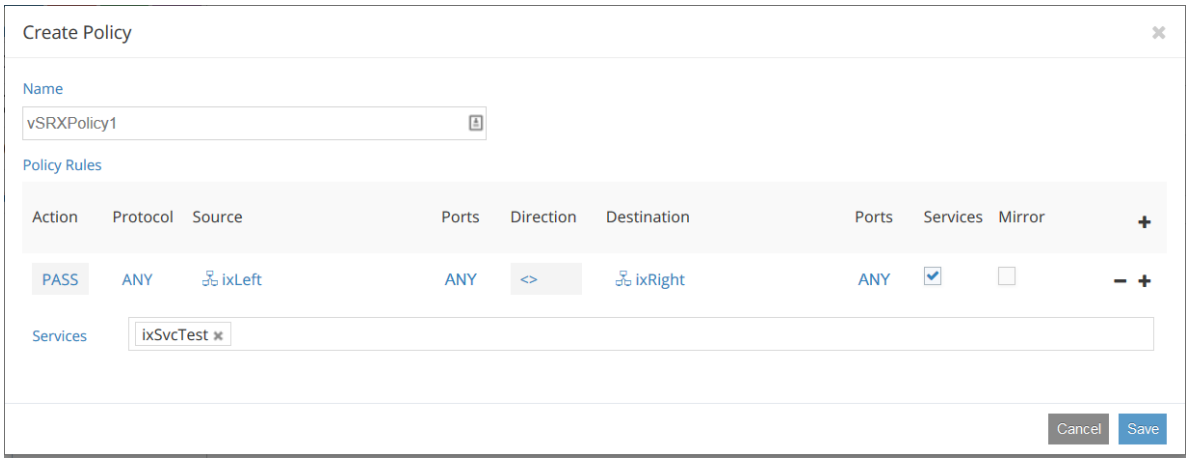
Create a Network Policy

To create a network policy:

1. Select **Configure>Networking>Policies**. The table of policies appears.

2. Click **+** to create a new policy. The Create Policy dialog box appears, as shown in Figure 51 on page 250.

Figure 51: Creating a Network Policy in Contrail



3. Name the policy.
4. Click **+** to create a new rule for this policy.
5. Select the left virtual network you created from the Source list and select the right virtual network from the Destination list.
6. Select the appropriate protocol from the Protocol list and select the source and destination ports for this policy.
7. Select **Services** and select the vSRX instance you want to apply this policy to.
8. Optionally, add more policy rules to this policy.
9. Click **Save** to create this policy.

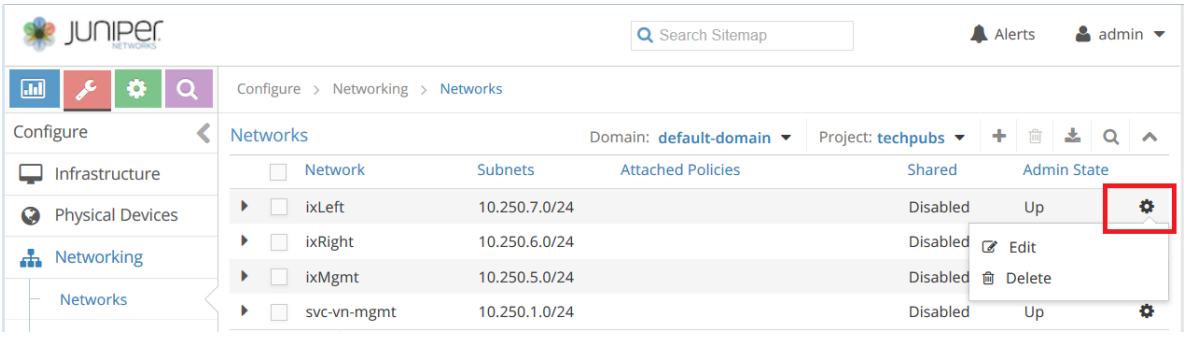
See Contrail - Creating a Network Policy for more details.

Add a Network Policy to a Virtual Network

To add a network policy to a virtual network:

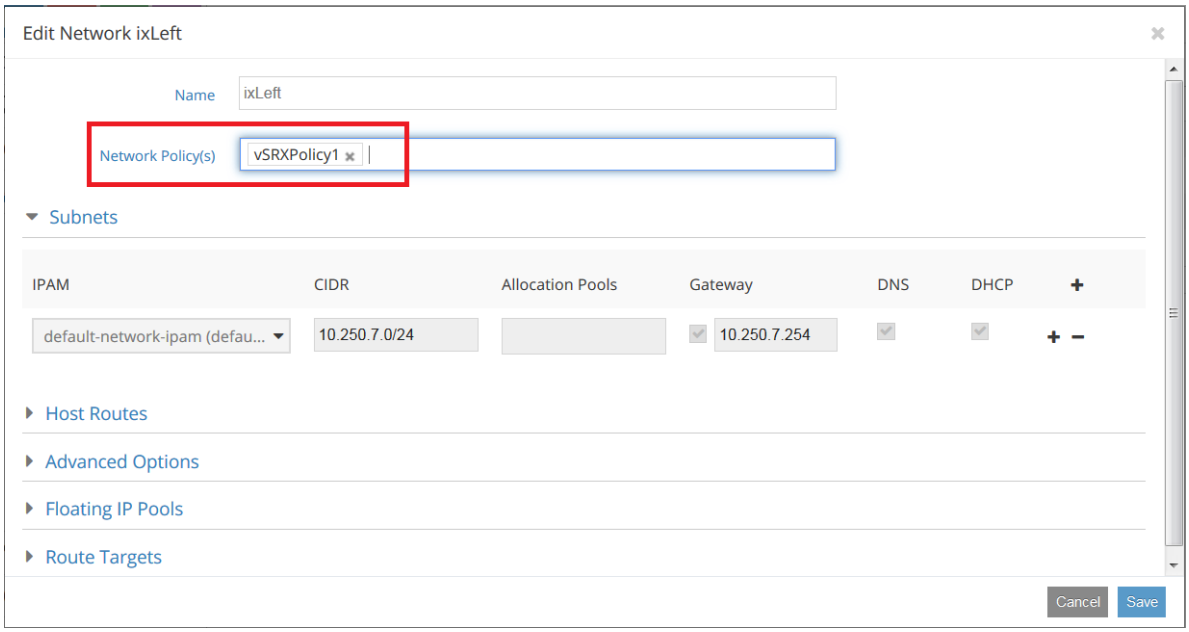
1. Select **Configure>Networking**, and select the settings icon to the right of the virtual network you want to add a network policy to, as shown in Figure 52 on page 251.

Figure 52: Contrail Virtual Networks



2. Click **Edit**. The Edit Networks dialog box appears, as shown in Figure 53 on page 251.

Figure 53: Adding a Network Policy to a Virtual Network in Contrail



3. Select the appropriate policy from the Networks Policy(s) list.
4. Click **Save** to save this change.
5. Repeat this procedure for the other virtual network in this service chain.

See Contrail - Associating a Network to a Policy for more details.

RELATED DOCUMENTATION

- Contrail - Creating an In-Network or In-Network-NAT Service Chain
- Contrail - Installation Overview

CHAPTER 14

Install vSRX in Contrail

IN THIS CHAPTER

Enable Nested Virtualization

We recommend that you enable nested *virtualization* on your host OS or OpenStack compute node. Nested virtualization is enabled by default on Ubuntu but is disabled by default on *CentOS*.

Use the following command to determine if nested virtualization is enabled on your host OS. The result should be Y.

hostOS# **cat /sys/module/kvm_intel/parameters/nested**

hostOS# Y

NOTE: APIC virtualization (APICv) does not work well with nested VMs such as those used with KVM. On Intel CPUs that support APICv (typically v2 models, for example E5 v2 and E7 v2), you must disable APICv on the host server before deploying vSRX.

To enable nested virtualization on the host OS:

1. Depending on your host operating system, perform the following:
   - On CentOS, open the **/etc/modprobe.d/dist.conf** file in your default editor.

```
hostOS# vi /etc/modprobe.d/dist.conf
```

- On Ubuntu, open the **/etc/modprobe.d/qemu-system-x86.conf** file in your default editor.

```
hostOS# vi /etc/modprobe.d/qemu-system-x86.conf
```

2. Add the following line to the file:

```
hostOS# options kvm-intel nested=y enable_apicv=n
```

3. Save the file and reboot the host OS.
4. (Optional) After the reboot, verify that nested virtualization is enabled.

```
hostOS# cat /sys/module/kvm_intel/parameters/nested
```

```
hostOS# Y
```

5. On Intel CPUs that support APICv (for example, E5 v2 and E7 v2), disable APICv on the host OS.

```
root@host# sudo rmmod kvm-intel
root@host# sudo sh -c “echo ’options kvm-intel enable_apicv=n’ >> /etc/modprobe.d/dist.conf”
root@host# sudo modprobe kvm-intel
```

6. Optionally, verify that APICv is now disabled.

```
root@host# cat /sys/module/kvm_intel/parameters/enable_apicv
```

```
N
```

Create an Image Flavor with OpenStack

IN THIS SECTION

Before you begin, ensure that you have a working OpenStack installation. See the OpenStack Installation Guide for more details.

OpenStack launches instances of images, based on the image installed and VM templates called *flavors*. Flavors set the memory, vCPU, and storage requirements for the vSRX image. You can use the Horizon GUI or the OpenStack `nova` commands to create flavors for the vSRX VMs. See *Requirements for vSRX on Contrail* for the software requirement specifications for a vSRX VM.

CAUTION: The packet forwarding engine (PFE) on the vSRX might become unresponsive if the NUMA nodes topology properties in OpenStack includes the line `hw:numa_nodes=2` to spread the instance's vCPUs across multiple host NUMA nodes. We recommend that you remove the `hw:numa_nodes=2` line from OpenStack to ensure that the PFE functions properly.

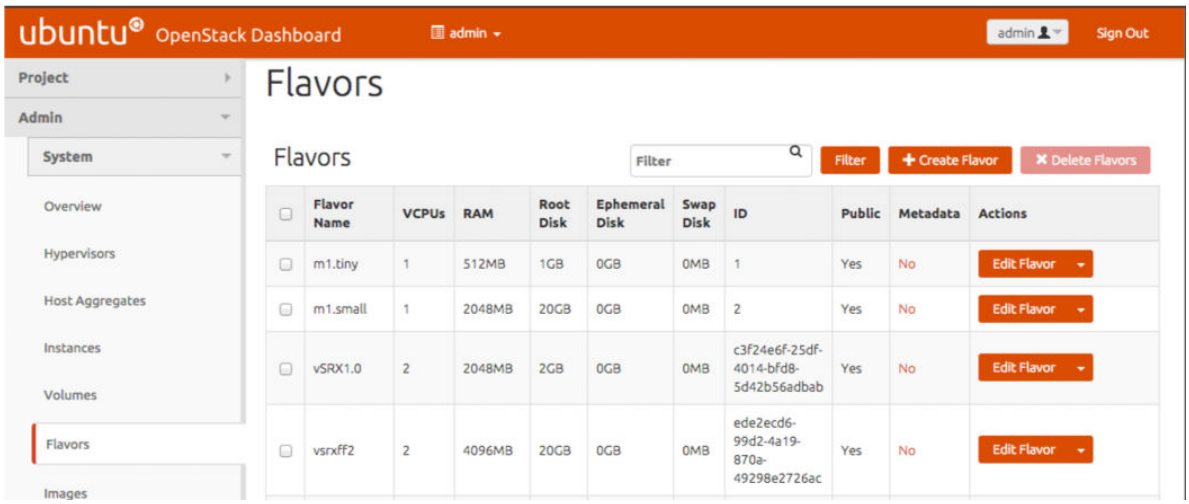
Create an Image Flavor for vSRX with Horizon

OpenStack uses VM templates, or flavors, to set the memory, vCPU, and storage requirements for an image. OpenStack includes a default set of flavors, but we recommend that you create a flavor to match the vSRX image requirements.

To create an image flavor for vSRX with Horizon:

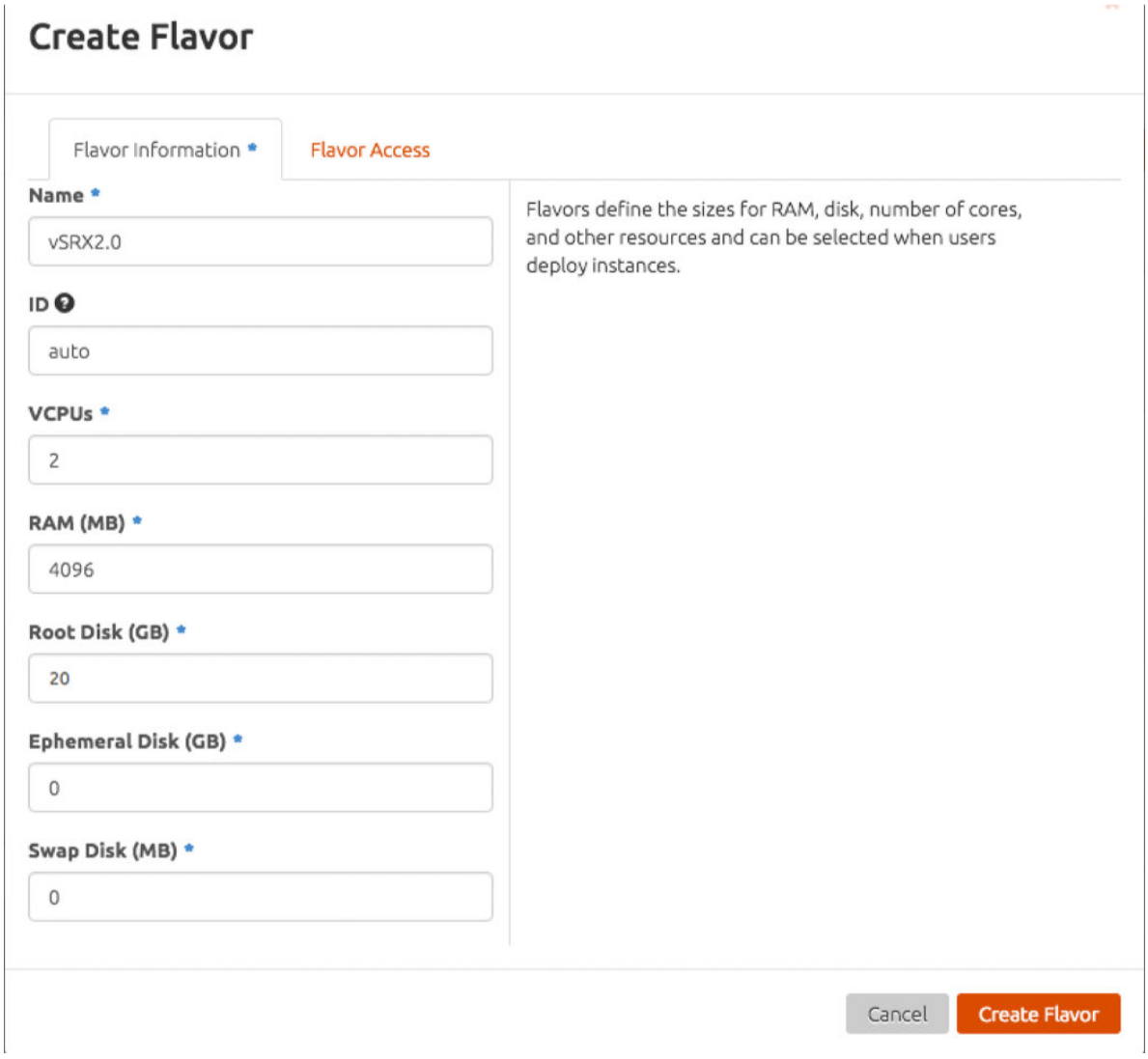
1. From the Horizon GUI, select your project, and select **Admin>System Panel>Flavors**. The list of existing image flavors appears, as shown in Figure 54 on page 256.

Figure 54: OpenStack Flavors



2. Click **Create Flavor**. The Create Flavor dialog box appears, as shown in Figure 55 on page 257.

Figure 55: Create a Flavor



3. Enter a name in the Name box for this vSRX flavor.
4. Enter the appropriate value in the vCPUs box for your configuration. The minimum required for vSRX is 2 vCPUs.
5. Enter the appropriate value in the RAM MB box. The minimum required for vSRX is 4096 MB.
6. Enter the appropriate value in the Root Disk GB box. The minimum required for vSRX is 20 GB.
7. Enter the appropriate values in the Emphemeral Disk GB and Swap Disk MB boxes. The minimum required for vSRX is 0 for each.
8. Click **Create Flavor**. The flavor appears on the Flavors tab.

Create an Image Flavor for vSRX with the Nova CLI

To create an image flavor for vSRX with the `nova` CLI command:

1. Use the `nova flavor-create` command on the OpenStack compute node that will host the vSRX VM. See Table 56 on page 258 for a list of mandatory parameters.

   NOTE: See the official `OpenStack` documentation for a complete description of available options for the `nova flavor-create` command.

   Table 56: nova flavor-create Command

Command Option	Description
--is-public true	Set the flavor as publicly available.
flavor_name	Name the vSRX flavor.
auto	Select auto to automatically assign the flavor ID.
ram_megabytes	Allocate RAM for the VM, in megabytes.
disk_gigabytes	Specify disk storage size for the VM.
vcpus	Allocate the number of vCPUs for the vSRX VM.

   NOTE: Use `nova help flavor-create` for more details on the command options.

2. Optionally, use the `nova flavor-list` to verify the flavors.

The following example creates a vSRX flavor with 4096 MB RAM, 2 vCPUs, and disk storage up to 20 GB:

```
$ nova flavor-create  --is-public true vsrx_flavor auto 4096 20 2
```

RELATED DOCUMENTATION

- OpenStack Installation Guide
- OpenStack End User Guide

Upload the vSRX Image

IN THIS SECTION

Contrail integrates with OpenStack for public, private, or hybrid cloud orchestration. You can install the vSRX image and use this installed image to provide security services in a service chain with Contrail.

Before installing vSRX, ensure that you have installed either Contrail and, optionally, OpenStack Glance.

- Contrail - Installation Overview
- OpenStack - Add the Image Service (glance)

You can upload the vSRX image with either Horizon, the OpenStack GUI dashboard, or Glance, the OpenStack CLI-based image services project.

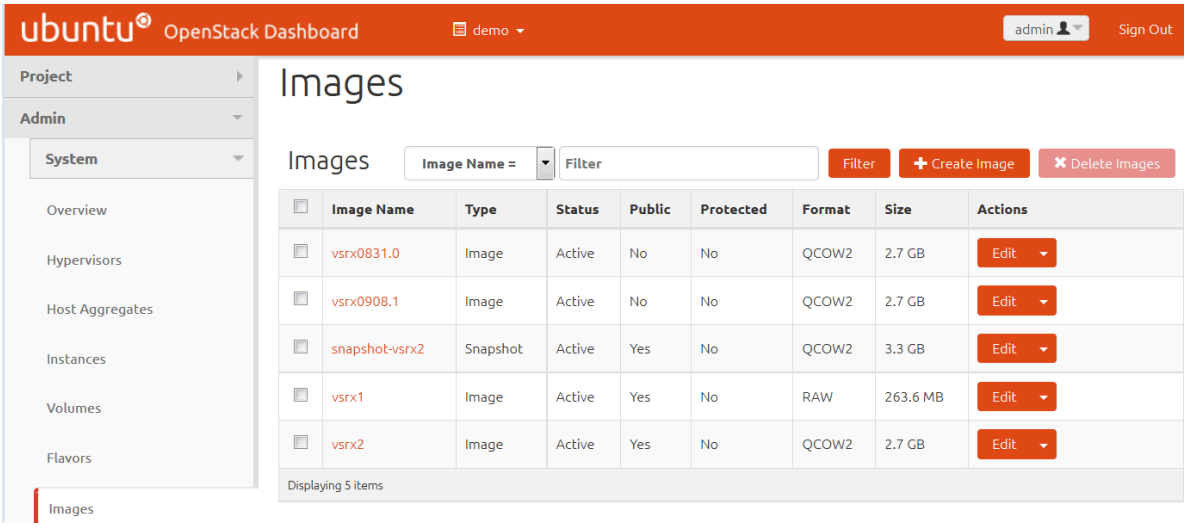
NOTE: To upgrade an existing vSRX instance, see *Migration, Upgrade, and Downgrade* in the *vSRX Release Notes*.

Upload the vSRX Image with OpenStack Horizon

To upload a vSRX image with Horizon:

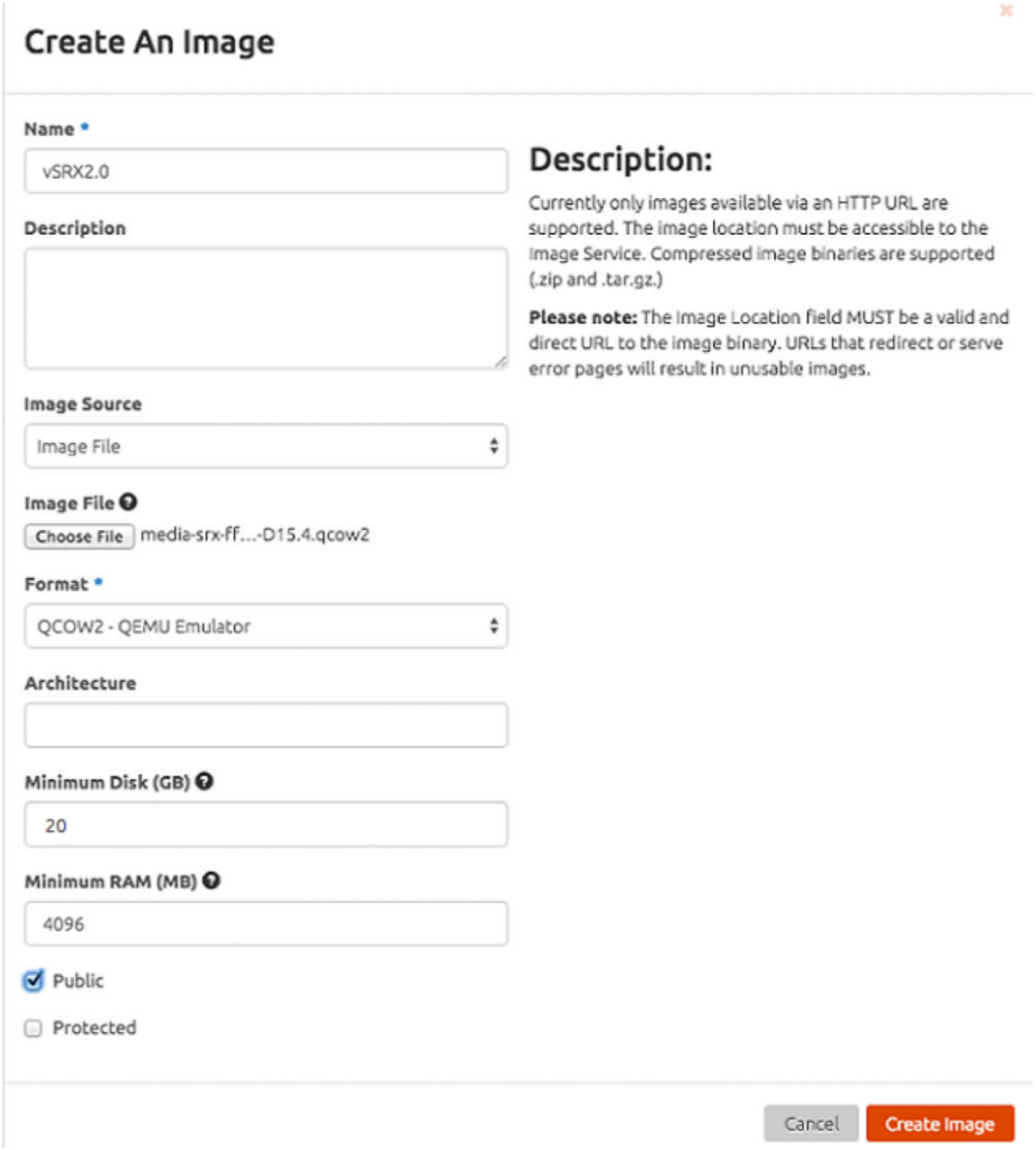
1. From the Horizon GUI, select your project, and select **Compute>Images**. The list of existing images appears, as shown in Figure 56 on page 260.

Figure 56: OpenStack Images



2. Click **Create Image**. The Create Image dialog box appears, as shown in Figure 57 on page 261.

Figure 57: Create an Image



3. Enter a name for the vSRX image, and enter the image location.
4. Select **QCOW2- QEMU Emulator** from the Format list.
5. Enter the appropriate value in the Minimum Disk (GB) box for your configuration. The minimum required for vSRX is 20 GB.

6. Enter the appropriate value in the Minimum RAM (MB) box. The minimium required for vSRX is 4096 MB.
7. Select **Public**.
8. Click **Create Image**. OpenStack uploads the image to the image service. The image appears on the Images tab.

> **NOTE**: The default vSRX VM login ID is root with no password. By default, vSRX is assigned a DHCP-based IP address if a DHCP server is available on the network.

Upload the vSRX Image with the OpenStack Glance CLI

To upload a vSRX image with the Glance CLI:

1. Log in to the appropriate OpenStack compute node.
2. Use `wget` to download the vSRX image to the compute node.
3. Use `glance image-create` to add the image to the image service with a base configuration for disk, format, and memory requirements. Use `glance help image-create` for complete details on this command-line tool.

   For example, the following command adds the vSRX QCOW2 image to the image service with 20 GB disk space and 4096 MB of RAM:

   glance image-create --name='vSRXimage' --is-public=true --container-format=bare --disk-format=qcow2 --min-disk=20 --min-ram=4096 --file=media-srx-ffp-vsrx-vmdisk-15.1X49-D120.qcow2

   > **NOTE**: vSRX requires at least 20 GB of disk space and 4096 MB of RAM.

> **NOTE**: The default vSRX VM login ID is root with no password. By default, vSRX is assigned a DHCP-based IP address if a DHCP server is available on the network.

RELATED DOCUMENTATION

- Contrail - Installation Overview
- OpenStack Installation Guide for Ubuntu 14.04
- OpenStack - Add the Image Service (glance)
- OpenStack - Upload and Manage Images

Use Cloud-Init in an OpenStack Environment to Automate the Initialization of vSRX Instances

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX image to help simplify configuring new vSRX instances operating in an OpenStack environment according to a specified user-data file. Cloud-init is performed during the first-time boot of a vSRX instance.

Cloud-init is an OpenStack software package for automating the initialization of a cloud instance at boot-up. It is available in Ubuntu and most major Linux and FreeBSD operating systems. Cloud-init is designed to support multiple different cloud providers so that the same virtual machine (VM) image can be directly used in multiple hypervisors and cloud instances without any modification. Cloud-init support in a VM instance runs at boot time (first-time boot) and initializes the VM instance according to the specified user-data file.

A user-data file is a special key in the metadata service that contains a file that cloud-aware applications in the VM instance can access upon a first-time boot. In this case, it is the validated Junos OS configuration file that you intend to upload to a vSRX instance as the active configuration. This file uses the standard Junos OS command syntax to define configuration details, such as root password, management IP address, default gateway, and other configuration statements.

When you create a vSRX instance, you can use cloud-init with a validated Junos OS configuration file (**juniper.conf**) to automate the initialization of new vSRX instances. The user-data file uses the standard Junos OS syntax to define all the configuration details for your vSRX instance. The default Junos OS configuration is replaced during the vSRX instance launch with a validated Junos OS configuration that you supply in the form of a user-data file.

NOTE: If using a release *earlier* than Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, the user-data configuration file cannot exceed 16 KB. If your user-data file exceeds this limit, you must compress the file using gzip and use the compressed file. For example, the gzip junos.conf command results in the junos.conf.gz file.

Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, if using a configuration drive data source in an OpenStack environment, the user-data configuration file size can be up to 64 MB.

The configuration must be validated and include details for the fxp0 interface, login, and authentication. It must also have a default route for traffic on fxp0. If any of this information is missing or incorrect, the instance is inaccessible and you must launch a new one.

WARNING: Ensure that the user-data configuration file is not configured to perform autoinstallation on interfaces using Dynamic Host Configuration Protocol (DHCP) to assign an IP address to the vSRX. Autoinstallation with DHCP will result in a "commit fail" for the user-data configuration file.

Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, the cloud-init functionality in vSRX has been extended to support the use of a configuration drive data source in an OpenStack environment. The configuration drive uses the user-data attribute to pass a validated Junos OS configuration file to the vSRX instance. The user-data can be plain text or MIME file type text/plain. The configuration drive is typically used in conjunction with the Compute service, and is present to the instance as a disk partition labeled `config-2`. The configuration drive has a maximum size of 64 MB, and must be formatted with either the vfat or ISO 9660 filesystem.

The configuration drive data source also provides the flexibility to add more than one file that can be used for configuration. A typical use case would be to add a Day0 configuration file and a license file. In this case, there are two methods that can be employed to use a configuration drive data source with a vSRX instance:

- User-data (Junos OS Configuration File) alone—This approach uses the user-data attribute to pass the Junos OS configuration file to each vSRX instance. The user-data can be plain text or MIME file type text/plain.
- Junos OS configuration file and license file—This approach uses the configuration drive data source to send the Junos OS configuration and license file(s) to each vSRX instance.

NOTE: If a license file is to be configured in vSRX, it is recommended to use the `-file` option rather than the `user-data` option to provide the flexibility to configure files larger than the 16 KB limit of user-data.

To use a configuration drive data source to send Junos OS configuration and license file(s) to a vSRX instance, the files needs to be sent in a specific folder structure. In this application, the folder structure of the configuration drive data source in vSRX is as follows:

```
-   OpenStack
        -   latest
              -   junos-config
                      -   configuration.txt
              -   junos-license
                      -   License_file_name.lic
                      -   License_file_name.lic
```

`//OpenStack//latest/junos-config/configuration.txt`

`//OpenStack//latest/junos-license/license.lic`

Before you begin:

- Create a configuration file with the Junos OS command syntax and save it. The configuration file can be plain text or MIME file type text/plain. The string `#junos-config` must be the first line of the user-data configuration file before the Junos OS configuration.

  NOTE: The `#junos-config` string is mandatory in the user-data configuration file; if it is not included, the configuration will not be applied to the vSRX instance as the active configuration.

- Determine the name for the vSRX instance you want to initialize with a validated Junos OS configuration file.
- Determine the flavor for your vSRX instance, which defines the compute, memory, and storage capacity of the vSRX instance.
- Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, if using a configuration drive, ensure the following criteria is met to enable cloud-init support for a configuration drive in OpenStack:
  - The configuration drive must be formatted with either the `vfat` or `iso9660` filesystem.

> **NOTE**: The default format of a configuration drive is an ISO 9660 file system. To explicitly specify the ISO 9660/vfat format, add the `config_drive_format=iso9660/vfat` line to the `nova.conf` file.

- The configuration drive must have a filesystem label of `config-2`.
- The folder size must be no greater than 64 MB.

Depending on your OpenStack environment, you can use either an OpenStack command-line interface (such as `nova boot` or `openstack server create`) or the OpenStack Dashboard ("Horizon") to launch and initialize a vSRX instance.

Perform Automatic Setup of a vSRX Instance Using an OpenStack Command-Line Interface

You can launch and manage a vSRX instance using either the `nova boot` or `openstack server create` commands, which includes the use of a validated Junos OS configuration user-data file from your local directory to initialize the active configuration of the target vSRX instance.

To initiate the automatic setup of a vSRX instance from an OpenStack command-line client:

1. If you have not done so already, create a configuration file with the Junos OS command syntax and save the file. The configuration file can be plain text or MIME file type text/plain.

   The user-data configuration file must contain the full vSRX configuration that is to be used as the active configuration on each vSRX instance, and the string `#junos-config` must be the first line of the user-data configuration file before the Junos OS configuration.

   > **NOTE**: The `#junos-config` string is mandatory in the user-data configuration file; if it is not included, the configuration will not be applied to the vSRX instance as the active configuration.

2. Copy the Junos OS configuration file to an accessible location from where it can be retrieved to launch the vSRX instance.
3. Depending on your OpenStack environment, use the `nova boot` or `openstack server create` command to launch the vSRX instance with a validated Junos OS configuration file as the specified user-data.

   > **NOTE**: You can also use the `nova boot` equivalent in an Orchestration service such as HEAT.

   For example:

- nova boot -user-data **</path/to/vsrx_configuration.txt>** --image **vSRX_image** --flavor **vSRX_flavor_instance**
- openstack server create -user-data **</path/to/vsrx_configuration.txt>** --image **vSRX_image** --flavor **vSRX_flavor_instance**

  Where:

  -user-data **</path/to/vsrx_configuration.txt>** specifies the location of the Junos OS configuration file. The user-data configuration file size is limited to approximately 16,384 bytes.

  --image **vSRX_image** identifies the name of a unique vSRX image.

  --flavor **vSRX_flavor_instance** identifies the vSRX flavor (ID or name).

Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, to enable the use of a configuration drive for a specific request in the OpenStack compute environment, include the -config-drive true parameter in the nova boot or openstack server create command.

NOTE: It is possible to enable the configuration drive automatically on all instances by configuring the OpenStack Compute service to always create a configuration drive. To do this, specify the force_config_drive=True option in the nova.conf file.

For example, to use the user-data attribute to pass the Junos OS configuration to each vSRX instance:

nova boot -config-drive true -flavor **vSRX_flavor_instance** -image **vSRX_image** -user-data **</path/to/vsrx_configuration.txt>**

Where:

-user-data **</path/to/vsrx_configuration.txt>** specifies the location of the Junos OS configuration file. The user-data configuration file size is limited to approximately 64 MB.

-image **vSRX_image** identifies the name of a unique vSRX image.

-flavor **vSRX_flavor_instance** identifies the vSRX flavor (ID or name).

For example, to specify the configuration drive with multiple files (Junos OS configuration file and license file):

nova boot -config-drive true -flavor **vSRX_flavor_instance** -image **vSRX_image** [-file **/junos-config/configuration.txt=/path/to/file**] [-file **/junos-license/license.lic=path/to/license**]

Where:

[-file **/junos-config/configuration.txt=/path/to/file**] specifies the location of the Junos OS configuration file.

`[-file /junos-license/license.lic=path/to/license]` specifies the location of the Junos OS configuration file.

`-image vSRX_image` identifies the name of a unique vSRX image.

`-flavor vSRX_flavor_instance` identifies the vSRX flavor (ID or name).

4. Boot or reboot the vSRX instance. During the initial boot-up sequence, the vSRX instance processes the cloud-init request.

   NOTE: The boot time for the vSRX instance might increase with the use of the cloud-init package. This additional time in the initial boot sequence is due to the operations performed by the cloud-init package. During this operation, the cloud-init package halts the boot sequence and performs a lookup for the configuration data in each data source identified in the cloud.cfg. The time required to look up and populate the cloud data is directly proportional to the number of data sources defined. In the absence of a data source, the lookup process continues until it reaches a predefined timeout of 30 seconds for each data source.

5. When the initial boot-up sequence resumes, the user-data file replaces the original factory-default Junos OS configuration loaded on the vSRX instance. If the commit succeeds, the factory-default configuration will be permanently replaced. If the configuration is not supported or cannot be applied to the vSRX instance, the vSRX will boot using the default Junos OS configuration.

SEE ALSO

- Cloud-Init Documentation
- OpenStack command-line clients
- Compute service (nova) command-line client
- Openstack Server Create
- Enabling the configuration drive (configdrive)
- Instances

Perform Automatic Setup of a vSRX Instance from the OpenStack Dashboard (Horizon)

Horizon is the canonical implementation of the OpenStack Dashboard. It provides a Web-based user interface to OpenStack services including Nova, Swift, Keystone, and so on. You can launch and manage a vSRX instance from the OpenStack Dashboard, which includes the use of a validated Junos OS configuration user-data file from your local directory to initialize the active configuration of the target vSRX instance.

To initiate the automatic setup of a vSRX instance from the OpenStack Dashboard:

1. If you have not done so already, create a configuration file with the Junos OS command syntax and save the file. The configuration file can be plain text or MIME file type text/plain.

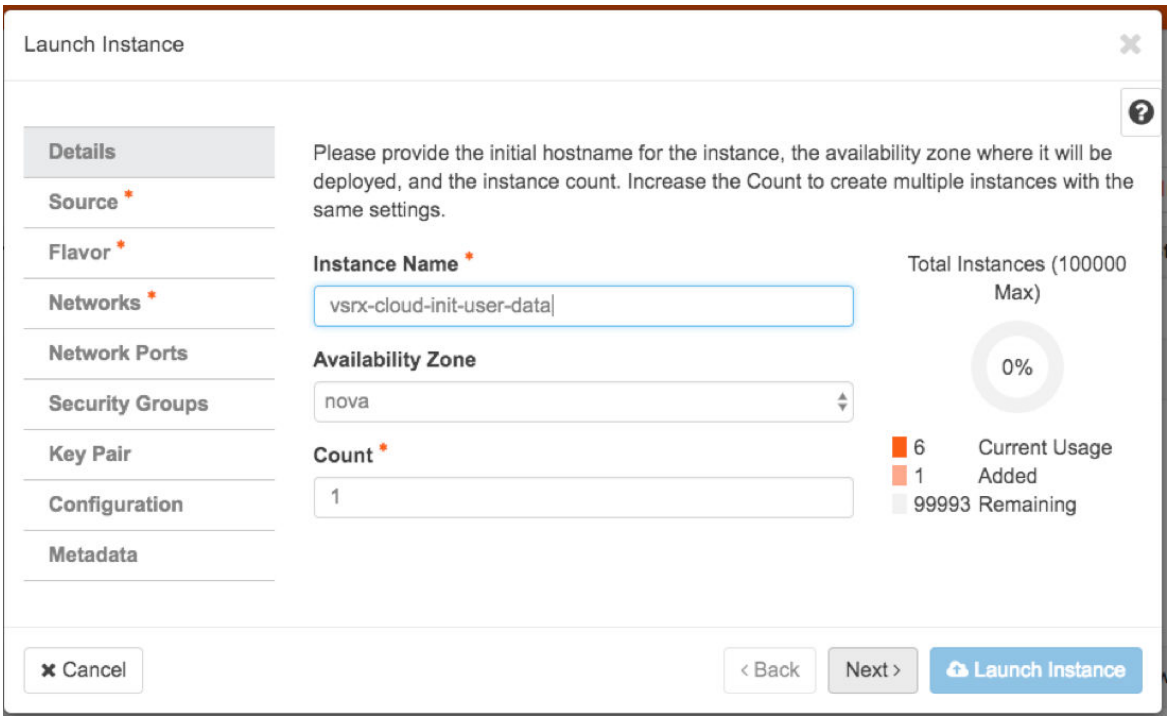
   The user-data configuration file must contain the full vSRX configuration that is to be used as the active configuration on each vSRX instance, and the string `#junos-config` must be the first line of the user-data configuration file before the Junos OS configuration.

   > **NOTE**: The `#junos-config` string is mandatory in the user-data configuration file; if it is not included, the configuration will not be applied to the vSRX instance as the active configuration.

2. Copy the Junos OS configuration file to an accessible location from where it can be retrieved to launch the vSRX instance.
3. Log in to the OpenStack Dashboard using your login credentials and then select the appropriate project from the drop-down menu at the top left.
4. On the Project tab, click the **Compute** tab and select **Instances**. The dashboard shows the various instances with its image name, its private and floating IP addresses, size, status, availability zone, task, power state, and so on.
5. Click **Launch Instance**. The Launch Instance dialog box appears.

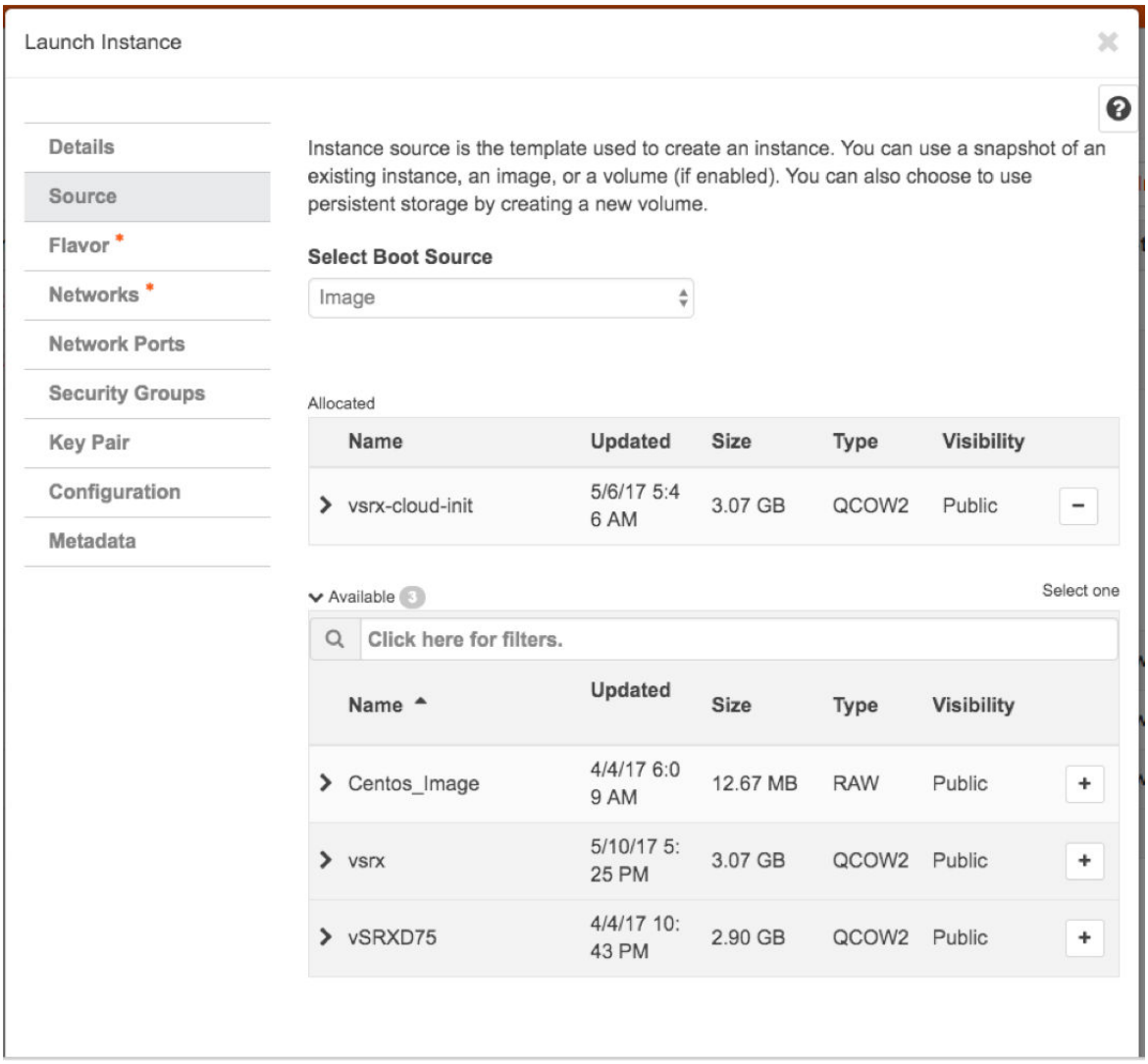
6. From the Details tab (see Figure 5 on page 54), enter an instance name for the vSRX VM along with the associated availability zone (for example, Nova) and then click **Next**. We recommend that you keep this name the same as the hostname assigned to the vSRX VM.

Figure 58: Launch Instance Details Tab



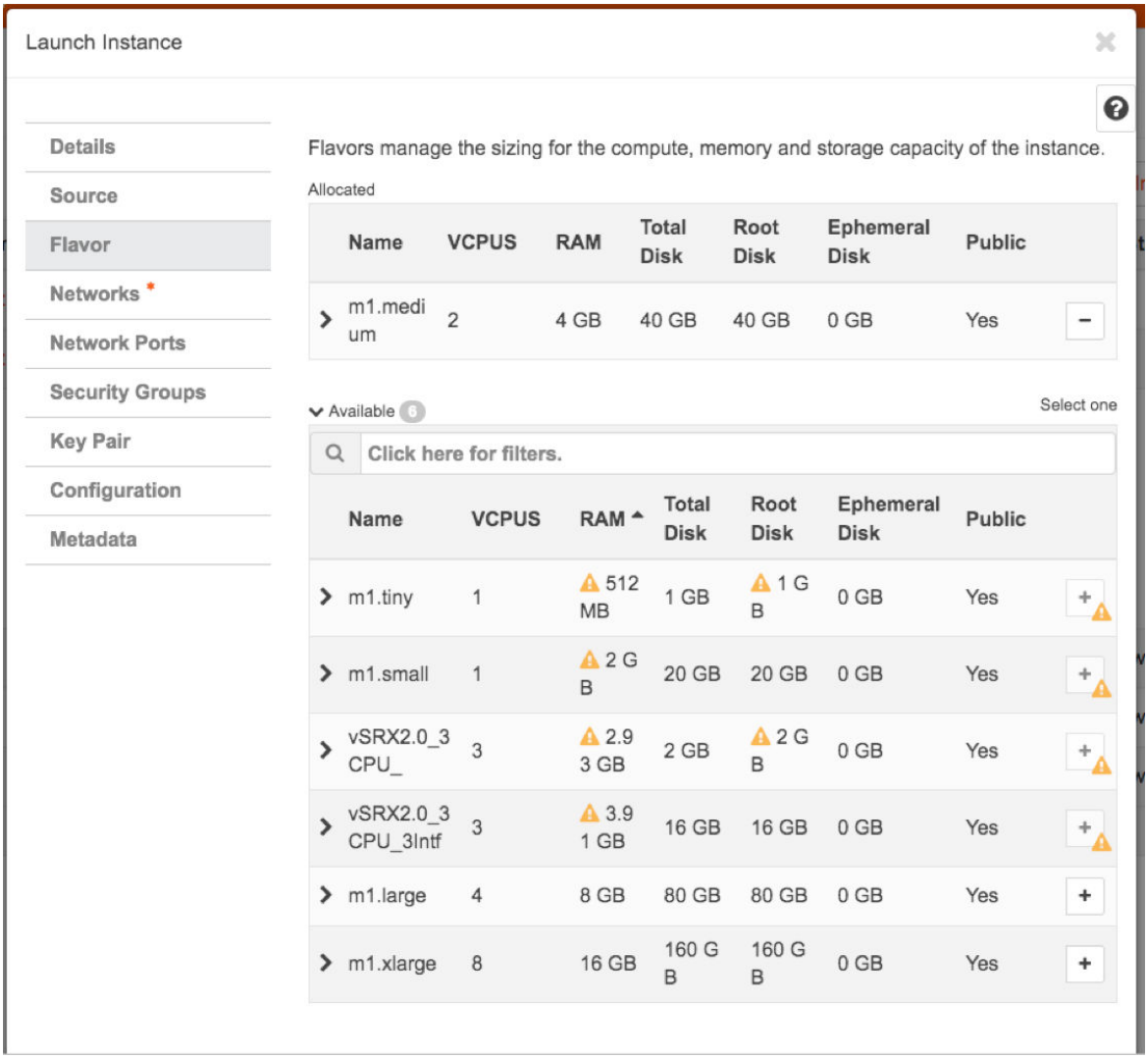
7. From the Source tab (see Figure 6 on page 55), select a vSRX VM image source file from the Available list and then click **+(Plus)**. The selected vSRX image appears under Allocated. Click **Next**.

Figure 59: Launch Instance Source Tab



8. From the Flavor tab (see Figure 7 on page 56), select a vSRX instance with a specific compute, memory, and storage capacity from the Available list and then click **+(plus sign)**. The selected vSRX flavor appears under Allocated. Click **Next**.

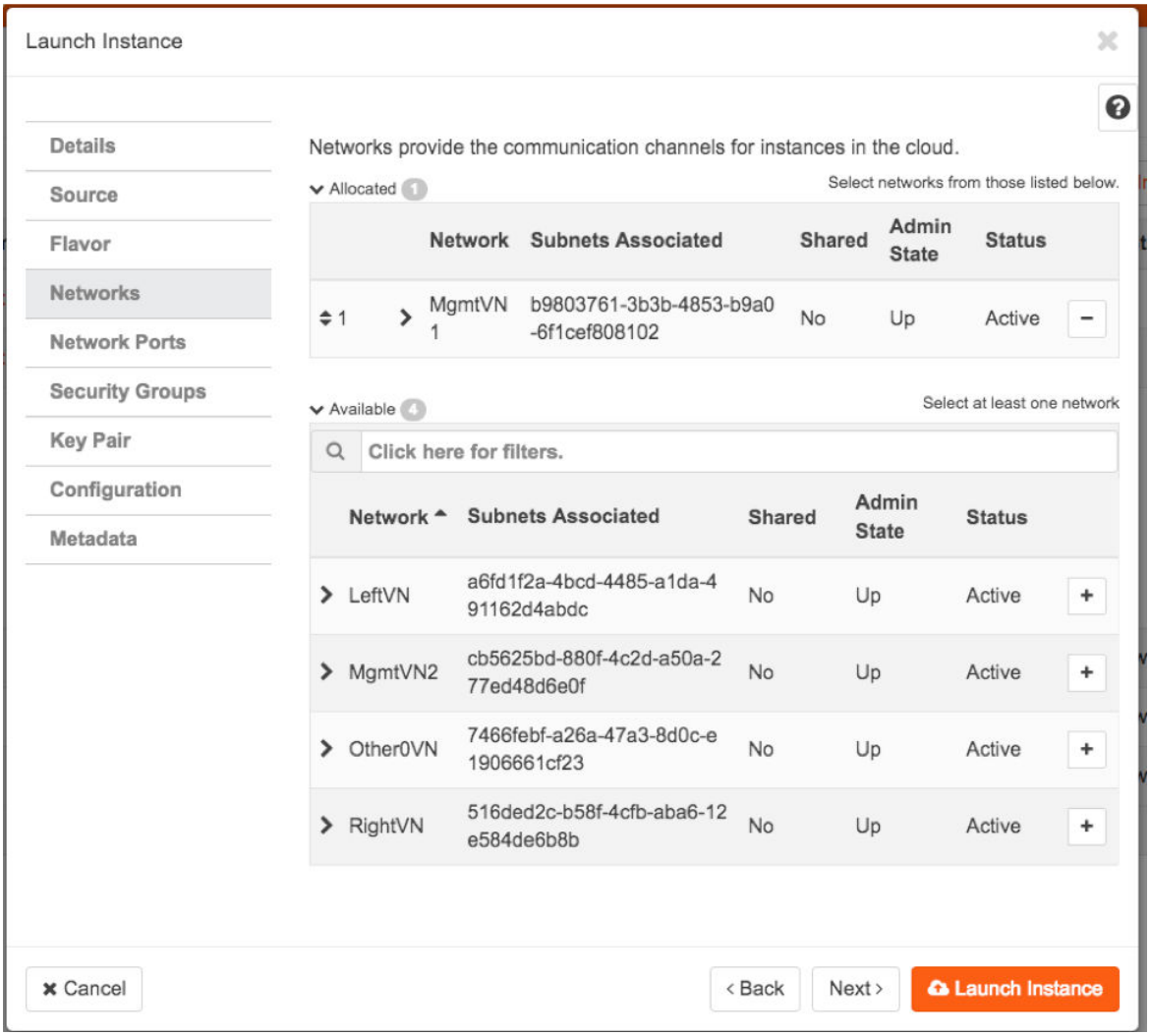
Figure 60: Launch Instance Flavor Tab



9. From the Networks tab (see Figure 8 on page 57), select the specific network of the vSRX instance from the Available list and then click **+(plus sign)**. The selected network appears under Allocated. Click **Next**.

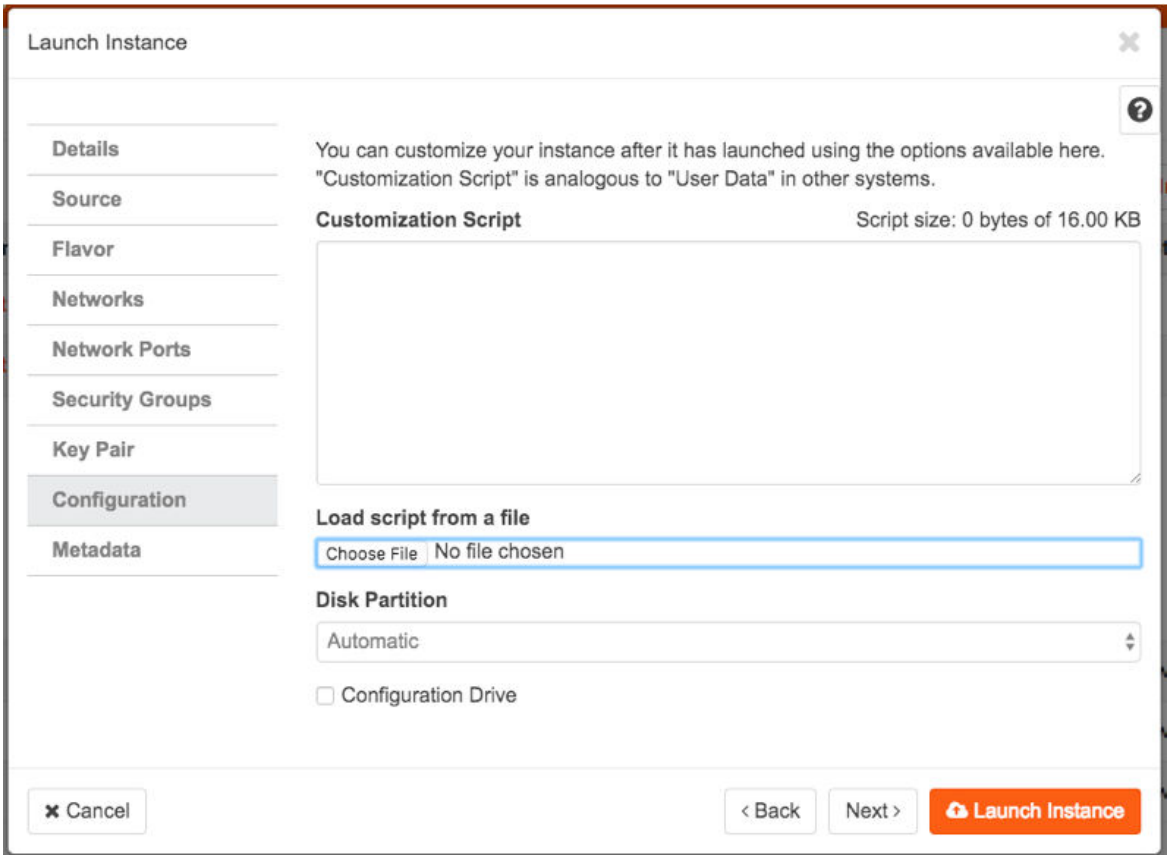
NOTE: Do not update any parameters in the Network Ports, Security Groups, or Key Pair tabs in the Launch Instance dialog box.

Figure 61: Launch Instance Networks Tab



10. From the Configuration tab (see Figure 9 on page 58), click **Browse** and navigate to the location of the validated Junos OS configuration file from your local directory that you want to use as the user-data file. Click **Next**.

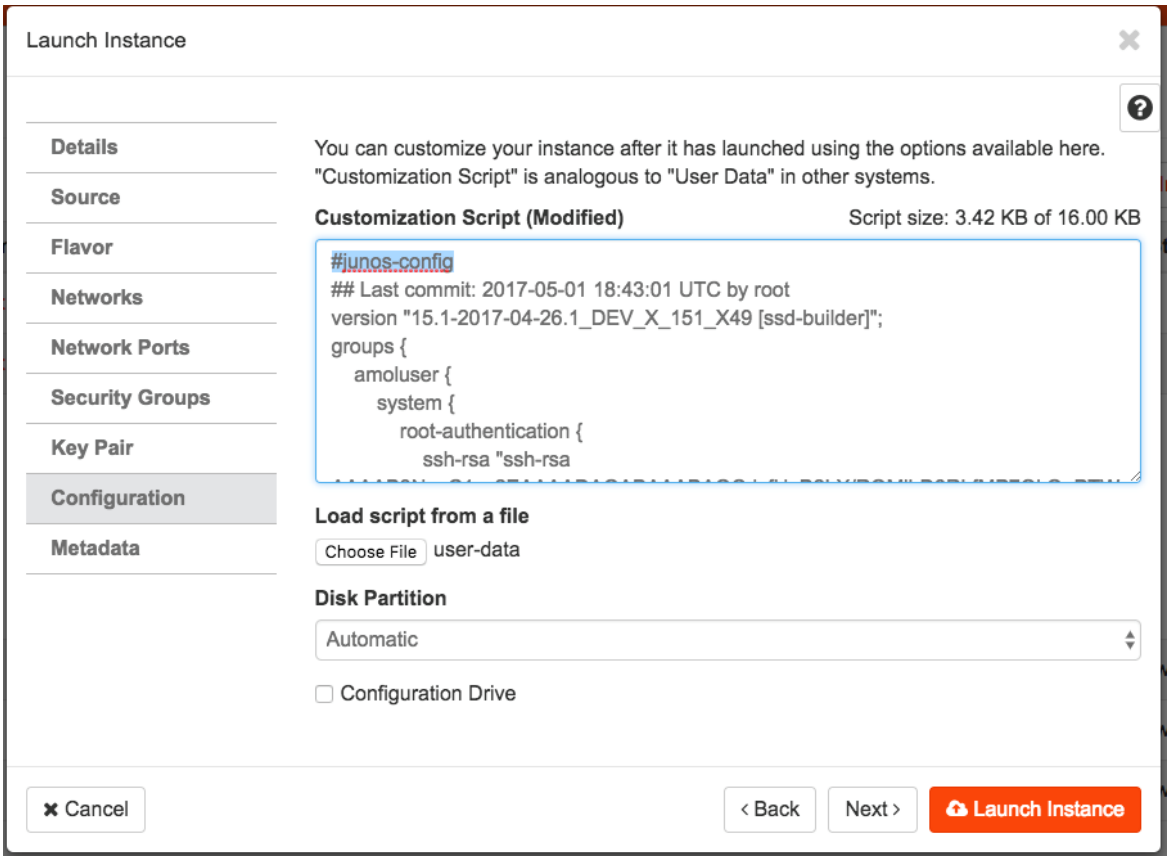
Figure 62: Launch Instance Configuration Tab



11. Confirm that the loaded Junos OS configuration contains the `#junos-config` string in the first line of the user-data configuration file (see Figure 10 on page 59) and then click **Next**.

NOTE: Do not update any parameters in the Metadata tab of the Launch Instance dialog box.

Figure 63: Launch Instance Configuration Tab with Loaded Junos OS Configuration



12. Click **Launch Instance**. During the initial boot-up sequence, the vSRX instance processes the cloud-init request.

NOTE: The boot time for the vSRX instance might increase with the use of the cloud-init package. This additional time in the initial boot sequence is due to the operations performed by the cloud-init package. During this operation, the cloud-init package halts the boot sequence and performs a lookup for the configuration data in each data source identified in the cloud.cfg. The time required to look up and populate the cloud data is directly proportional to the number of data sources defined. In the absence of a data source, the lookup process continues until it reaches a predefined timeout of 30 seconds for each data source.

13. When the initial boot-up sequence resumes, the user-data file replaces the original factory-default Junos OS configuration loaded on the vSRX instance. If the commit succeeds, the factory-default configuration will be permanently replaced. If the configuration is not supported or cannot be applied to the vSRX instance, the vSRX will boot using the default Junos OS configuration.

SEE ALSO

- Cloud-Init Documentation
- OpenStack Dashboard
- Launch and Manage Instances
- Horizon: The OpenStack Dashboard Project

Release History Table

Release	Description
15.1X49-D130	Starting in Junos OS Release 15.1X49-D130 and Junos OS Release 18.4R1, the cloud-init functionality in vSRX has been extended to support the use of a configuration drive data source in an OpenStack environment. The configuration drive uses the user-data attribute to pass a validated Junos OS configuration file to the vSRX instance.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX image to help simplify configuring new vSRX instances operating in an OpenStack environment according to a specified user-data file. Cloud-init is performed during the first-time boot of a vSRX instance.

CHAPTER 15

vSRX VM Management with Contrail

IN THIS CHAPTER

Connect to the vSRX Management Console

IN THIS SECTION

Ensure that you have launched the vSRX VM with Contrail.

You can connect to the vSRX console through OpenStack or Contrail.

Connect to the vSRX Management Console with Horizon

To connect to the vSRX console with OpenStack Horizon:

1. From the Horizon GUI, select your project, and select **Compute>Instances**. The list of existing instances appears.
2. From the Actions column, select **Console** from the More list. The vSRX console appears, and you can log in to the management port for the vSRX instance.

Connect to the vSRX Management Console with Contrail

To connect to the vSRX console of a vSRX VM with Contrail:

1. From the Contrail GUI, select your project, and select **Configure>Services>Service Instances**. The list of existing service instances appears.
2. Click on the left arrow next to the vSRX VM to expand to the Service Instance Details view.
3. Click the **View Console** link on the right. The vSRX console appears, and you can log in to the management port for the vSRX.

RELATED DOCUMENTATION

- OpenStack End User Guide
- Contrail - Creating an In-Network or In-Network-NAT Service Chain

Manage the vSRX VM

IN THIS SECTION

Each vSRX instance is an independent *virtual machine* (VM) that you can power on, pause, or shut down.

Power On the VM from OpenStack

To power on the VM:

1. From the OpenStack dashboard for your project, select **Compute>Instances**. The list of existing instances appears.
2. Check the VM you want to power on.
3. From the Actions column, select **Start Instance** from the list.

Pause the VM

To pause the VM:

1. From the Horizon GUI for your project, select **Compute>Instances**. The list of existing instances appears.
2. Check the VM that you want to pause.
3. From the Actions column, select **Pause Instance** from the list.

Restart the VM

To restart the VM:

1. From the Horizon GUI for your project, select **Compute>Instances**. The list of existing instances appears.
2. Check the VM that you want to reboot.
3. Select **Soft Reboot Instance** from the More list to restart the VM.

Power Off the VM from OpenStack

To power off the VM:

1. From the OpenStack dashboard for your project, select **Compute>Instances**. The list of existing instances appears.
2. Check the VM you want to power off.
3. From the Actions column, select **Console** from the list. The console opens.
4. From the console, power off the VM.

   ```
   user@host>request system power-off
   ```

Delete the vSRX VM from Contrail

BEST PRACTICE: We recommend that you use Contrail to delete any VMs used in service chains created by Contrail.

To delete the VM from Contrail:

1. From the Contrail GUI for your project, select **Configure>Services>Service Instances**. The list of existing service instances appears.
2. Select the VM that you want to delete.
3. Click trash icon on the upper right menu to delete the selected VMs.

RELATED DOCUMENTATION

Contrail - Creating an In-Network or In-Network-NAT Service Chain

Upgrade Multicore vSRX with Contrail

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, you can scale up the number of vCPUs or vRAM for a vSRX VM. You must gracefully power off the vSRX VM before you can scale up vSRX. See *Manage the vSRX VM* for details.

You can modify an existing flavor with the OpenStack Dashboard (Horizon). You cannot use the OpenStack CLI (`nova flavor`) commands to modify the CPU or RAM settings on an existing flavor. Instead, create a new flavor and modify the vSRX service template in Contrail to use this new flavor. See the *Create an Image Flavor with OpenStack* for details.

NOTE: You cannot scale down the number of vCPUs or vRAM for an existing vSRX VM.

Configure Multi-queue Virtio Interface for vSRX VM with OpenStack

Before you plan to scale up vSRX performance, enable network multi-queuing as a means to support an increased number of dataplane vCPUs for the vSRX VM. The default for vSRX in Contrail is 2 dataplane vCPUs, but you can scale that number to 4 vCPUs.

To use multiqueue virtio interfaces, ensure your system meets the following requirements:

OpenStack Liberty supports the ability to create VMs with multiple queues on their virtio interfaces. Virtio is a Linux platform for I/O virtualization, providing a common set of I/O virtualization drivers. Multiqueue virtio is an approach that enables the processing of packet sending and receiving to be scaled to the number of available virtual CPUs (vCPUs) of a guest, through the use of multiple queues

- The OpenStack version must be Liberty or greater.

- The maximum number of queues in the vSRX VM interface is set to the same value as the number of vCPUs in the guest.
- The vSRX VM image metadata property is set to enable multiple queues inside the VM.

Use the following command on the OpenStack node to enable multiple queues on a vSRX VM in Contrail:

```
source /etc/contrail/openstackrc
```

nova image-meta <image_name> set hw_vif_multiqueue_enabled="true"

After the vSRX VM is spawned, use the following command on the virtio interface in the guest to enable multiple queues inside the vSRX VM:

ethtool –L <interface_name> combined <#queues>

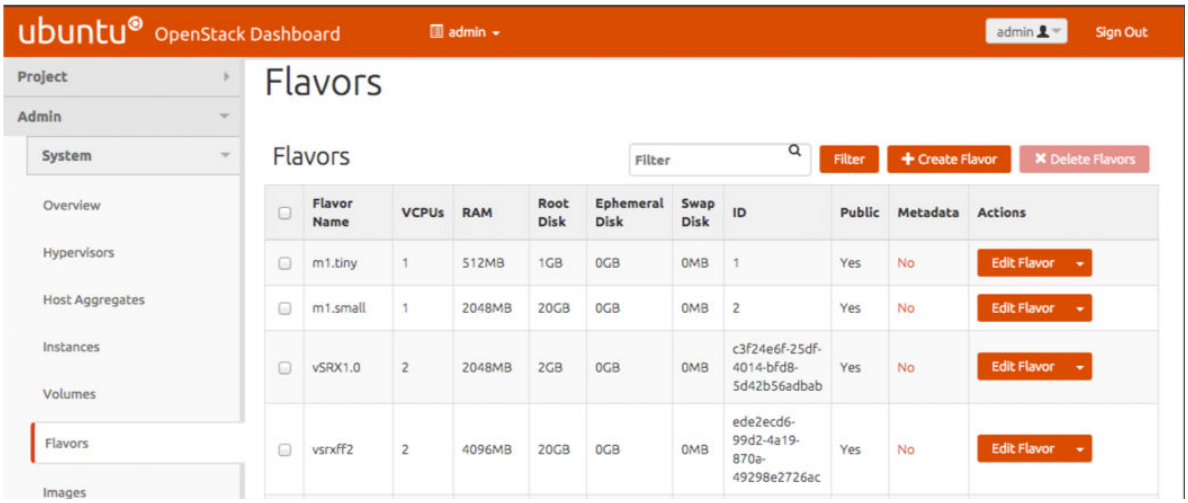
Modify an Image Flavor for vSRX with the Dashboard

OpenStack uses VM templates, or flavors, to set the memory, vCPU, and storage requirements for an image.

To Modify an image flavor for vSRX with the OpenStack dashboard:

1. From the dashboard select your project, and select **Admin>System Panel>Flavors**. The list of existing image flavors appears, as shown in Figure 64 on page 281.

Figure 64: OpenStack Flavors



2. Select the vSRX flavor and click **Edit Flavor**. The Edit Flavor dialog box appears.
3. Increase the number of vCPUs for your configuration. The minimum required for vSRX is 2 vCPUs.
4. Increase the RAM MB value. The minimum required for vSRX is 4096 MB.

5. Click **Create Flavor**. The flavor appears on the Flavors tab.

Update a Service Template

If you created a new image flavor for an existing vSRx instance, you need to update the service template to use this new image flavor before you relaunch the vSRX instance.

To update a service template:

1. From Contrail, select **Configure>Services>Service Templates**. The list of existing service templates appears.
2. Click on the vSRX service template and select edit.
3. Expand **Advanced Options** and select the new instance flavor from the Instance Flavor list.
4. Click **Save** to update this service template.
5. Power on the vSRX VM. See *Manage the vSRX VM* for details.

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, you can scale up the number of vCPUs or vRAM for a vSRX VM.

RELATED DOCUMENTATION

- OpenStack Installation Guide
- OpenStack End User Guide

Monitor vSRX with Contrail

To monitor basic statistics on the vSRX VM with Contrail:

1. On Contrail, select **Monitor>Networking>Instances**. The list of existing VMs appears.
2. Expand the row for the VM that you want to monitor. The CPU and memory statistics appear.
3. On Contrail, select **Monitor>Networking>Networks**. The list of existing virtual networks appears.
4. Expand the row for the virtual network that you want to monitor and select **Traffic Statistics**. The traffic and throughput statistics appear.

RELATED DOCUMENTATION

- Contrail - Monitor Networking

PART 5

vSRX Deployment for Nutanix

CHAPTER 16

Overview

IN THIS CHAPTER

Understand vSRX Deployment with Nutanix

IN THIS SECTION

Nutanix Platform Overview

IN THIS SECTION

The Nutanix Virtual Computing Platform is a converged, scale-out compute and storage system that is purpose-built to host and store virtual machines (VMs).

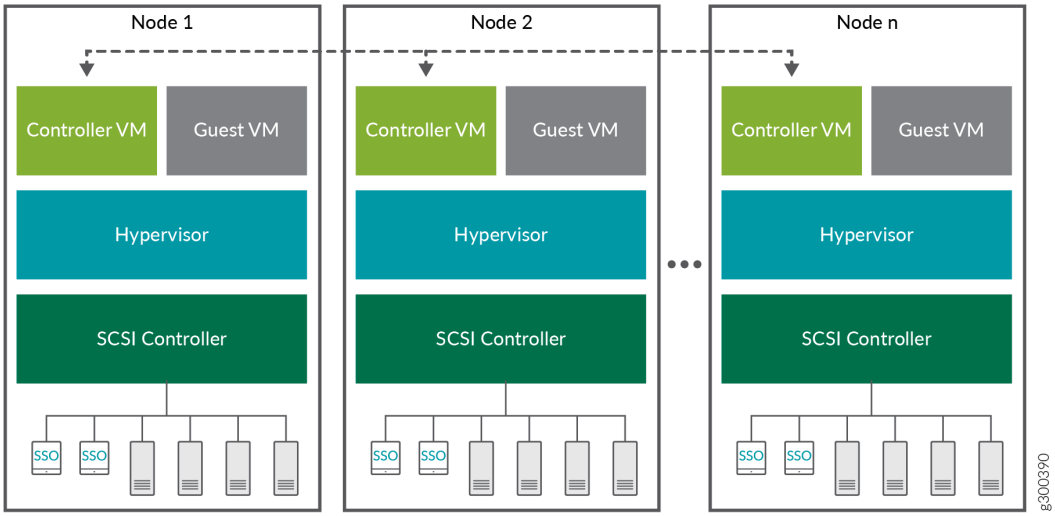
All nodes in a Nutanix cluster converge to deliver a unified pool of tiered storage and present resources to VMs for seamless access. A global data system architecture integrates each new node into the cluster,

allowing you to scale the solution to meet the needs of your infrastructure. Nutanix supports VMware vSphere (ESXi), Microsoft HyperV, Citrix XenServer, and Nutanix Acropolis hypervisor (AHV) (KVM-based).

The foundational unit for the cluster is a Nutanix node. Each node in the cluster runs a standard hypervisor and contains processors, memory, and local storage (SSDs and hard disks).

The Nutanix cluster has a distributed architecture, which means that each node in the cluster shares in the management of cluster resources and responsibilities. Within each node, there are software components that perform specific tasks during cluster operation. All components run on multiple nodes in the cluster, and depend on connectivity between their peers that also run the component. Most components also depend on other components for information.

A Nutanix Controller VM runs on each node, enabling the pooling of local storage from all nodes in the cluster.



Guest VM Data Management

VM data is stored locally, and replicated on other nodes for protection against hardware failure.

When a guest VM submits a write request through the hypervisor, that request is sent to the Controller VM on the host. To provide a rapid response to the guest VM, this data is first stored on the metadata drive, within a subset of storage. This cache is rapidly distributed across the 10-*Gigabit Ethernet* GbE network to other metadata drives in the cluster. Oplog data is periodically transferred to persistent storage within the cluster. Data is written locally for performance and replicated on multiple nodes for high availability.

When the guest VM sends a read request through the hypervisor, the Controller VM will read from the local copy first, if present. If the host does not contain a local copy, then the Controller VM will read

across the network from a host that does contain a copy. As remote data is accessed, it will be migrated to storage devices on the current host, so that future read requests can be local.

Guest VM data management includes the following features:

- **MapReduce tiering**—Nutanix cluster dynamically manages data based on how frequently it is accessed. New data is saved on the SSD tier. Frequently accessed data is kept on the SSD tier and old data is migrated to the HDD tier.

  Automated data migration also applies to read requests across the network. If a guest VM repeatedly accesses a block of data on a remote host, the local controller VM migrates that data to the SSD tier of the local host. This migration not only reduces network latency, but also ensures that frequently accessed data is stored on the fastest storage tier.

- **Live migration**—Live migration of VMs, whether it is initiated manually or through an automatic process like vSphere DRS, is fully supported by the Nutanix Virtual Computing Platform. All hosts within the cluster have visibility into shared Nutanix datastores through the Controller VMs. Guest VM data is written locally, and is also replicated on other nodes for high availability.

  If a VM is migrated to another host, future read requests are sent to a local copy of the data, if it exists. Otherwise, the request is sent across the network to a host that does contain the requested data. As remote data is accessed, the remote data is migrated to storage devices on the current host, so that future read requests are local.

- **High availability (HA)**—The built-in data redundancy in a Nutanix cluster supports high availability provided by the hypervisor. If a node fails, all high-availability-protected VMs can be automatically restarted on other nodes in the cluster. The hypervisor management system, such as vCenter, selects a new host for the VMs, which might or might not contain a copy of the VM data.

- **Virtualization management VM high availability**—In virtualization management VM high availability, when a node becomes unavailable, VMs that are running on that node are restarted on another node in the same cluster.

  Typically, an entity failure is detected by its isolation from the network (the failure to respond to heartbeats). Virtualization management ensures that at most one instance of the VM is running at any point during a failover. This property prevents concurrent network and storage I/O that could lead to corruption.

  Virtualization management VM high availability implements admission control to help ensure that in case of node failure, the rest of the cluster has enough resources to accommodate the other VMs.

- **Datapath redundancy**—The Nutanix cluster automatically selects the optimal path between a hypervisor host and its guest VM data. The Controller VM has multiple redundant paths available, which makes the cluster more resilient to failures.

When available, the optimal path is through the local Controller VM to local storage devices. In some situations, the data is not available on local storage, such as when a guest VM was recently migrated to another host. In those cases, the Controller VM directs the read request across the network to storage on another host through the Controller VM of that host.

Datapath redundancy also responds when a local Controller VM is unavailable. To maintain the storage path, the cluster automatically redirects the host to another Controller VM. When the local Controller VM comes back online, the datapath is returned to this VM.

vSRX Deployment with Nutanix Overview

IN THIS SECTION

This topic provides an overview of vSRX deployment on Nutanix Enterprise Cloud.

vSRX offers the same full-featured advanced security as the physical Juniper Networks SRX Series Services Gateways, but in a virtualized form factor. Handling speeds up to 100 Gbps, making it the industry's fastest virtual firewall. vSRX with Nutanix delivers:

- A single platform delivering high performance and predictable scale for any virtual workload.
- High-performance networking and security for scale-out virtual data centers.
- Flexibility with multi-hypervisor support (Hyper-V, ESXi, and Acropolis Hypervisor) and a full appliance portfolio for the right mix of compute and storage resources.
- VMs that keep running and are protected with VM-centric backups and integrated disaster recovery.
- Innovative Virtual Chassis Fabric architecture with automation capabilities for simplified management.

Manual, rigid, and static connectivity and security implementations might work in traditional network environments. In the multicloud era, however, where application requirements are highly dynamic, network security must be an agile and scalable partner to compute and storage.

Enterprise multiclouds typically employ perimeter security solutions like Nutanix Enterprise Cloud to block threats contained in north-south traffic entering or leaving the HCI. Effective as they are, these solutions cannot defend against threats introduced by compromised virtual machines (VMs) that infect east-west traffic flowing within the data center itself, between applications and services. If these threats are not identified and addressed in a timely manner, they could compromise mission-critical applications

and lead to the loss of sensitive data, causing irreparable harm to revenue and reputation of an organization.

vSRX works with Nutanix Enterprise Cloud to provide advanced security, consistent management, automated threat remediation, and effective microsegmentation—delivering a secure and automated solution for defending today's multicloud environments.

The joint Juniper Networks-Nutanix hyperconverged solution helps enterprises secure their multicloud environments with advanced security, consistent management, automated threat remediation, automation, and effective microsegmentation. Enterprises can now easily deploy a secure and automated multicloud without the overhead of operational and management complexity.

Nutanix provides on-demand services in the cloud. Services range from Infrastructure as a Service (IaaS) and Platform as a Service (SaaS), to Application and Database as a Service. Nutanix is a highly flexible, scalable, and reliable cloud platform. In Nutanix, you can host servers and services on the cloud as bring-your-own-license (BYOL) service.

Benefits of vSRX with Nutanix

- **Advanced security**—Protects the business by delivering advanced security services, including user and application firewall, advanced threat prevention, and intrusion prevention.
- **Microsegmentation**—Employs microsegmentation to secure applications and defend against lateral threat propagation in the enterprise multicloud. Protects virtual workloads through effective microsegmentation.

  Microsegmentation facilitates granular segmentation and control by applying security policies at the virtualized host level. From a security perspective, the more granular level at which a threat can be blocked, the more effective the defense will be in containing the threat's propagation. Administrators must augment their security solutions with microsegmentation and automated threat remediation, providing the visibility and control required to protect lateral data center traffic from common breaches.
- **Visibility**—Provides granular visibility and analytics into application, user, and IP behavior.
- **Automation**—Offers rich APIs and automation libraries from Nutanix and Juniper Networks to enable agile DevOps workflows; to deliver improved security response through unified automation of security and networking workflows.
- **Operational simplicity**—Streamlines and enables policy deployment and enforcement with single-pane management and simple, intuitive controls across multicloud deployments.

Understand vSRX Deployment with Nutanix AHV

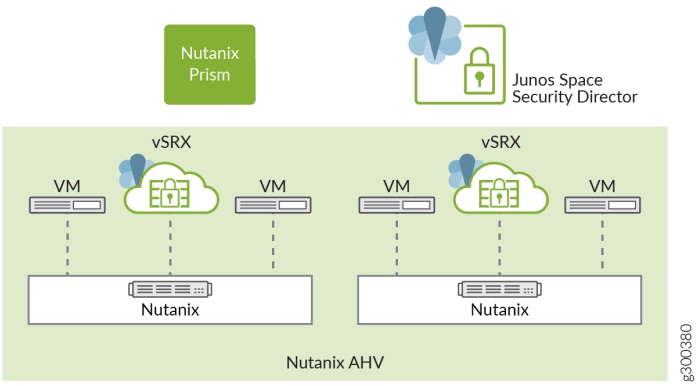
IN THIS SECTION

Nutanix Acropolis hyperconverged infrastructure (HCI) supports customer choice in virtualization solutions including VMware vSphere (ESXi), Microsoft HyperV, Citrix XenServer, and Nutanix AHV. AHV is a feature-rich Nutanix hypervisor. AHV is an enterprise-ready hypervisor based on proven open-source technology. Nutanix AHV is a license-free virtualization solution included with Acropolis that delivers enterprise virtualization ready for a multicloud world. With Acropolis and AHV, virtualization is tightly integrated into the Nutanix Enterprise Cloud OS rather than being layered on as a standalone product that needs to be licensed, deployed and managed separately.

Common tasks such as deploying, cloning, and protecting VMs are managed centrally through Nutanix Prism, rather than utilizing disparate products and policies in a piecemeal strategy.

Figure 65 on page 290 illustrates how security is provided for applications running in a private subnet of Nutanix Enterprise Cloud with AHV hypervisor.

Figure 65: vSRX Deployment in Nutanix Enterprise Cloud



The Nutanix AHV virtualization solution, including the tools you need to manage it, ships from the factory already installed and ready to go state so that you can have the system up and running as soon as you have racked the cluster and powered it on. When the system is up and running, you can maintain the environment through a simple HTML 5 Web UI. Prism Element, which is available on each cluster you deploy, integrates this UI with the overall Nutanix solution. You can access Prism Element through each individual Nutanix cluster through the cluster IP or any of the individual Nutanix Controller Virtual

Machine (CVM) IP addresses. Prism Element requires no additional software; it is built into every Nutanix cluster and incorporates support for AHV.

If you prefer a more centralized mechanism for managing your deployment, Prism Central is available from the Nutanix portal or can be deployed directly from the Nutanix cluster. Prism Central is a robust optional software appliance VM that can run on ESXi, Hyper-V, or AHV.

Prism Central is both a platform and a hypervisor-agnostic management interface, providing an aggregate view of your deployed Nutanix clusters. In addition to allowing you to view and manage the cluster, Prism Central provides insight into VMs, hosts, disks, and containers or pooled disks.

Prism Central provides a single pane of glass for managing not only multiple Nutanix clusters, but also the native Nutanix hypervisor, AHV. Unlike other hypervisors, AHV requires no additional back-end applications or database to maintain the data rendered in the UI.

Prism runs on every node in the cluster, but like other components, it elects a leader. All requests are forwarded from the followers to the leader using Linux iptables. This allows administrators to access Prism using any Controller VM IP address. If the Prism leader fails, a new leader is elected. The leader also communicates with the ESXi hosts for VM status and related information. Junos Space Security Director manages vSRX Virtual Firewalls deployed on each node of a Nutanix AHV cluster, and it acts as a unified security policy manager to apply consistent policies across all vSRX VMs in Nutanix-based private and public clouds (AWS/Azure).

Traffic between VMs and applications is redirected through the vSRX, allowing next-generation firewall security services with advanced threat prevention to be provisioned. Security policies enforced on traffic inside the Nutanix Enterprise Cloud augment the Nutanix HCI with microsegmentation, blocking sophisticated threats that propagate laterally while identifying and controlling application and user access. This enables security administrators to isolate and segment mission-critical applications and data using zero trust security principles.

Components of vSRX Deployment with Nutanix

Joint solution with vSRX and Nutanix includes the following key components:

- **vSRX Virtual Firewall**—vSRX offers the same full-featured advanced security as the physical Juniper Networks SRX Series Services Gateways, but in a virtualized form.
- **Junos Space Security Director**—Junos Space Security Director allows network operators to manage a distributed network of virtual and physical firewalls from a single location. Serving as the management interface for the vSRX Virtual Firewall, Security Director manages the firewall policies on all vSRX instances. It includes a customizable dashboard with details, threat maps, and event logs, providing unprecedented visibility into network security. Remote mobile monitoring is also possible through a mobile application for Google Android and Apple iOS systems.
- **Nutanix AHV**—Nutanix AHV is an enterprise-class virtualization solution included with the Nutanix Enterprise Cloud OS, with no additional software components to license, install, or manage. Starting

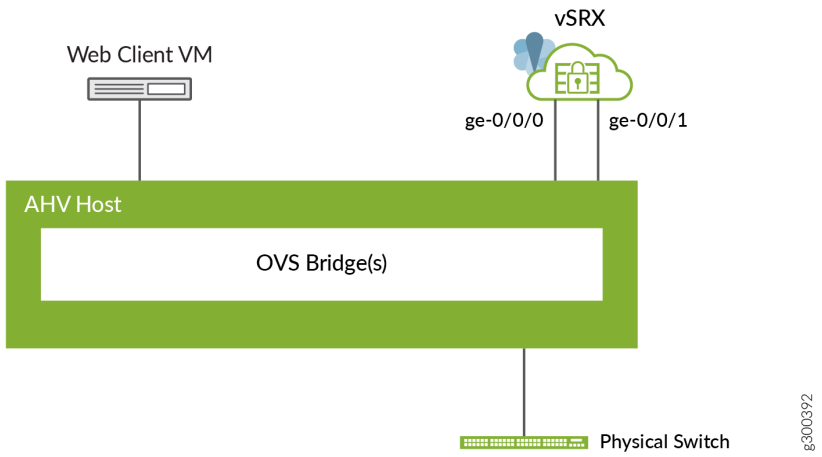
with proven open-source virtualization technology, AHV combines an enhanced datapath for optimal performance, security hardening, flow network virtualization, and complete management features to deliver a leaner yet more powerful virtualization stack, no costly shelfware, and lower virtualization costs.

- **Nutanix Manager (Nutanix Prism)**—Nutanix Prism is an end-to-end management tool for administrators to configure and monitor the Nutanix cluster and solutions for virtualized data center environments using the nCLI and the Web console. The end-to-end management capability streamlines and automates common workflows, eliminating the need for multiple management solutions across data center operations. Powered by advanced machine learning technology, Prism analyzes system data to generate actionable insights for optimizing virtualization and infrastructure management.

Sample vSRX Deployment Using Nutanix AHV

A Sample vSRX deployment to provide security for applications running in a private subnet of Nutanix Enterprise Cloud with AHV hypervisor is shown in Figure 66 on page 292.

Figure 66: Sample vSRX Deployment in Nutanix Enterprise Cloud Using AHV



A vSRX image is loaded into the Linux-based kernel with Nutanix AHV virtualization solution as the hypervisor. AHV-based VMs support multitenancy, allowing you to run multiple vSRX VMs on the host OS. AHV manages and shares the system resources between the host OS and the multiple vSRX VMs.

NOTE: vSRX requires you to enable hardware-based virtualization on a host OS that contains an Intel Virtualization Technology (VT) capable processor.

The basic components of this deployment include:

- **Linux bridge**—Used for CVM control traffic
- **Open vSwitch (OVS) bridge(s)**—Used form VM traffic and to connect to physical ports
- **Physical switch**—Transports in or out traffic to the physical network ports on the host

RELATED DOCUMENTATION

- Requirements for vSRX on KVM
- Upgrade a Multi-core vSRX
- Install vSRX with KVM

Requirements for vSRX on Nutanix

IN THIS SECTION

These topics provide an overview of requirements for deploying a vSRX 3.0 instance on Nutanix.

System Requirements for Nutanix

IN THIS SECTION

This topic provides the system requirement details.

Table 57 on page 294 lists the system requirements for a vSRX 3.0 instance deployed on Nutanix.

Table 57: System Requirements for vSRX 3.0

Component	Specification and Details
Hypervisor support	AHV 5.9
Memory	4 GB
Disk space	16 GB
vCPUs	2
vNICs	Up to 8
vNIC type	Virtio

Interface Mapping for vSRX 3.0 on Nutanix

Table 58 on page 294 shows the vSRX 3.0 and Nutanix interface names. The first network interface is used for the out-of-band management (fxp0) for vSRX 3.0.

Table 58: vSRX 3.0 and Nutanix Interface Names

Interface Number	vSRX 3.0 Interface	Nutanix Interface
1	fxp0	eth0
2	ge-0/0/0	eth1
3	ge-0/0/1	eth2
4	ge-0/0/2	eth3

Table 58: vSRX 3.0 and Nutanix Interface Names *(Continued)*

Interface Number	vSRX 3.0 Interface	Nutanix Interface
5	ge-0/0/3	eth4
6	ge-0/0/4	eth5
7	ge-0/0/5	eth6
8	ge-0/0/6	eth7

We recommend putting revenue interfaces in routing instances as a best practice to avoid asymmetric traffic/routing, because fxp0 is part of the default (inet.0) table by default. With fxp0 as part of the default routing table, there might be two default routes needed: one for the fxp0 interface for external management access, and the other for the revenue interfaces for traffic access. Putting the revenue interfaces in a separate routing instance avoids this situation of two default routes in a single routing instance.

NOTE: Ensure that interfaces belonging to the same security zone are in the same routing instance. See KB Article - Interface must be in the same routing instance as the other interfaces in the zone.

vSRX 3.0 Default Settings on Nutanix

vSRX 3.0 requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.

Table 59 on page 296 lists the factory-default settings for security policies on the vSRX 3.0.

Table 59: Factory-Default Settings for Security Policies

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

CAUTION: Do not use the `load factory-default` command on a vSRX 3.0 Nutanix instance. The factory-default configuration removes the Nutanix preconfiguration. If you must revert to factory default, ensure that you manually reconfigure Nutanix preconfiguration statements before you commit the configuration; otherwise, you will lose access to the vSRX 3.0 instance. See *Configure vSRX Using the CLI* for Nutanix preconfiguration details.

Best Practices for Improving vSRX 3.0 Performance

Refer the following deployment practices to improve vSRX 3.0 performance:

- Disable the source/destination check for all vSRX 3.0 interfaces.
- Limit public key access permissions to 400 for key pairs.
- Ensure that there are no contradictions between Nutanix security groups and your vSRX 3.0 configuration.
- Use vSRX 3.0 NAT to protect your instances from direct Internet traffic.

Reference Requirements

Requirements for vSRX 3.0 with different types of Hypervisors are:

- **Requirements for vSRX on VMware**—See *Requirements for vSRX on VMware*
- **Requirements for vSRX on KVM-Based Hypervisor**—See *Requirements for vSRX on KVM*
- **Requirements for vSRX with Hype-V-Based Hypervisor**—See *Requirements for vSRX on Microsoft Hyper-V*

CHAPTER 17

Install vSRX in Nutanix

IN THIS CHAPTER

Launch and Deploy vSRX in Nutanix AHV Cluster

IN THIS SECTION

Before you begin, you need a Nutanix account and an Identity and Access Management (IAM) role, with all required permissions to access, create, modify, and delete Nutanix cloud objects. You should also create access keys and corresponding secret access keys, X.509 certificates, and account identifiers. For better understanding of Nutanix terminologies and their use in vSRX deployments, see Understanding vSRX with Nutanix.

The topics in this section help you launch vSRX instances in a Nutanix AHV cluster.

Log In to Nutanix Setup

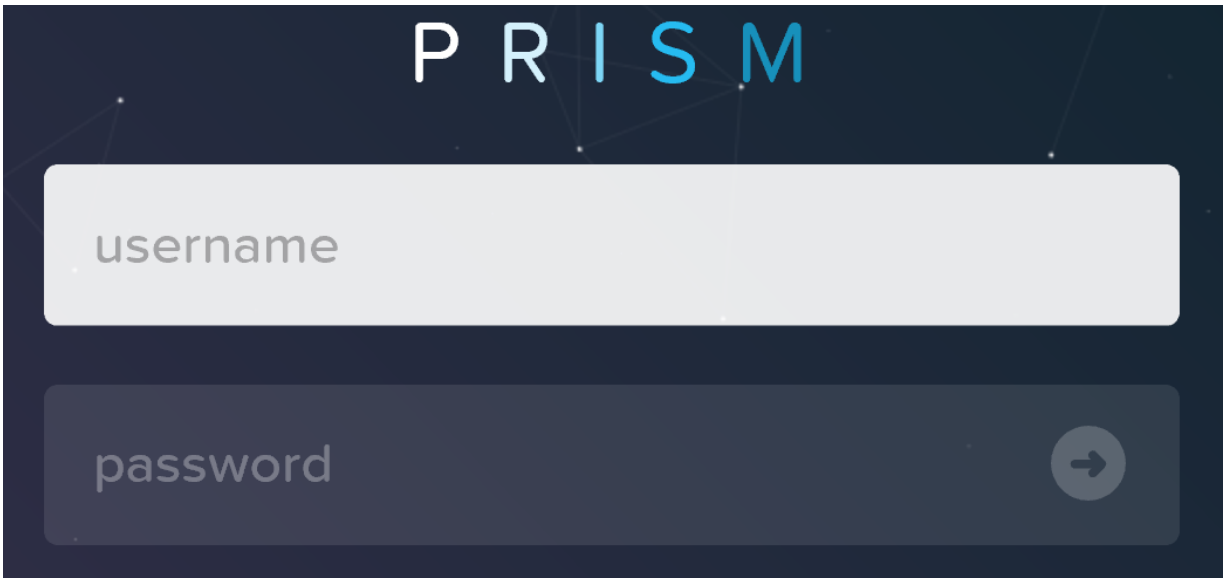
This topic provide details on how to log in to Nutanix setup.

Log in to the Nutanix Management Console.

NOTE: To access the Nutanix management console, remote access must be enabled on your local machine.

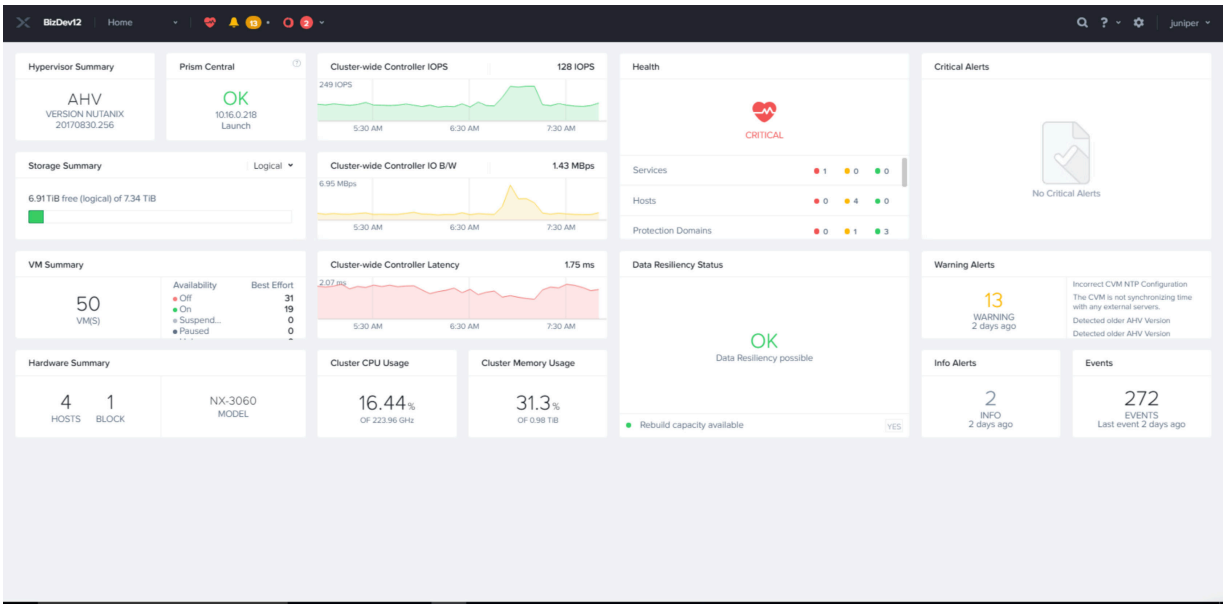
Once you have logged in to the remote Windows machine, you can access the Nutanix Prims Enable using your Web browser.

Figure 67: Prism Element Login Page



After you provide login details, the Nutanix Prism home page appears.

Figure 68: Initial Page of Prism Element



Adding a vSRX Image

Before you create a vSRX image, copy the image in the local machine from which the image can be accessed by Nutanix Prism Element. After copying, locally source the images from Prism GUI.

All the required vSRX images are available in the Juniper download page. After you copy the vSRX image on the local machine, complete the following steps to upload the image in Nutanix:

1. Click the **Image configuration** option from the **Tool** menu in the on top-right corner of the Prism home page.
2. Click the **Upload Image** tab.
3. Enter the required image details and provide a local file path under Image source. Wait for the image to be uploaded successfully.

Network Creation

This topic provides details on configuring the network for deploying vSRX VMs.

You can create a Routing Engine-FPC (RE-FPC) (or any other network) using the following steps:

1. At the top-right corner of the Nutanix Prism page, under Settings, click the **Network Configuration** option.

2. Click the **Create Network** button, add details for creating an internal network for RE-FPC communication, and click the **Save** button.

   A message appears, indicating that the RE-FPC internal network was successfully created.

> **NOTE**: In this deployment guide, all the the networks created on Nutanix setup are VLAN-based networks. Therefore, if you are deploying a Routing Engine and FPC on different hosts (compute nodes), the VLAN that is used by the RE-FPC internal networks must be part of the allowable VLAN range that is configured on the top-of-rack switch connecting the two machines.
>
> We tested the use case in which the Routing Engine and FPC were deployed on different hosts. However, for all our other tests, we deployed the Routing Engine and FPC on the same host.

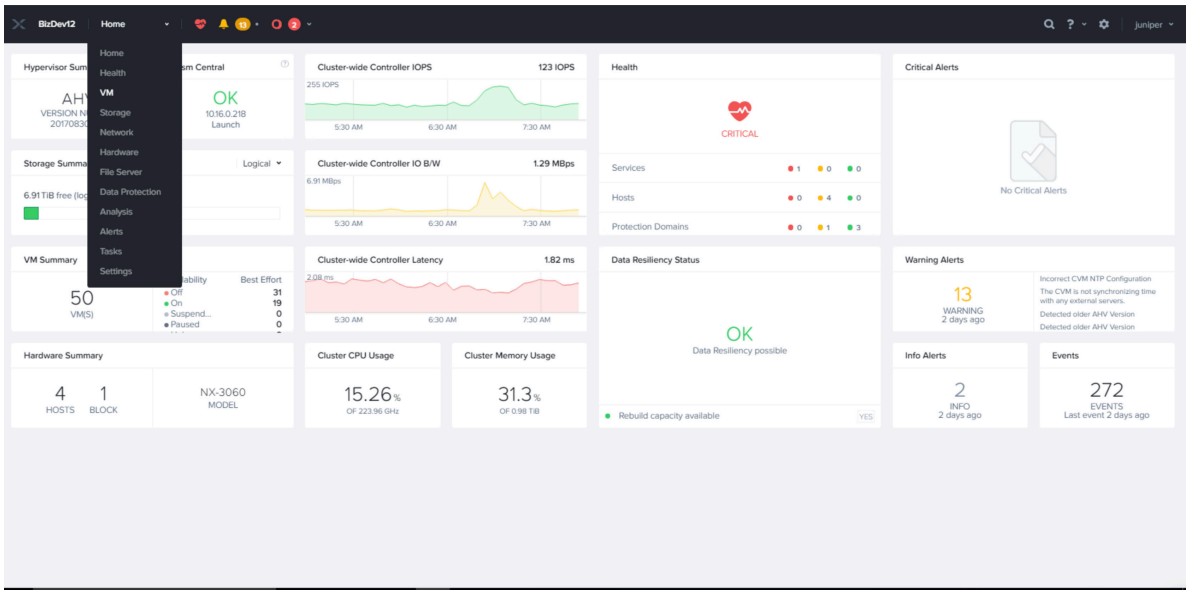
Create and Deploy a vSRX VM

This topic provides details on how to deploy a vSRX VM.

In Acropolis-managed clusters, you can create a new virtual machine (VM) through the Web console. When creating a VM, you can configure all of its components, such as number of vCPUs and memory, but you cannot attach a volume group to the VM. Attaching a volume group is possible only when you are modifying a VM.

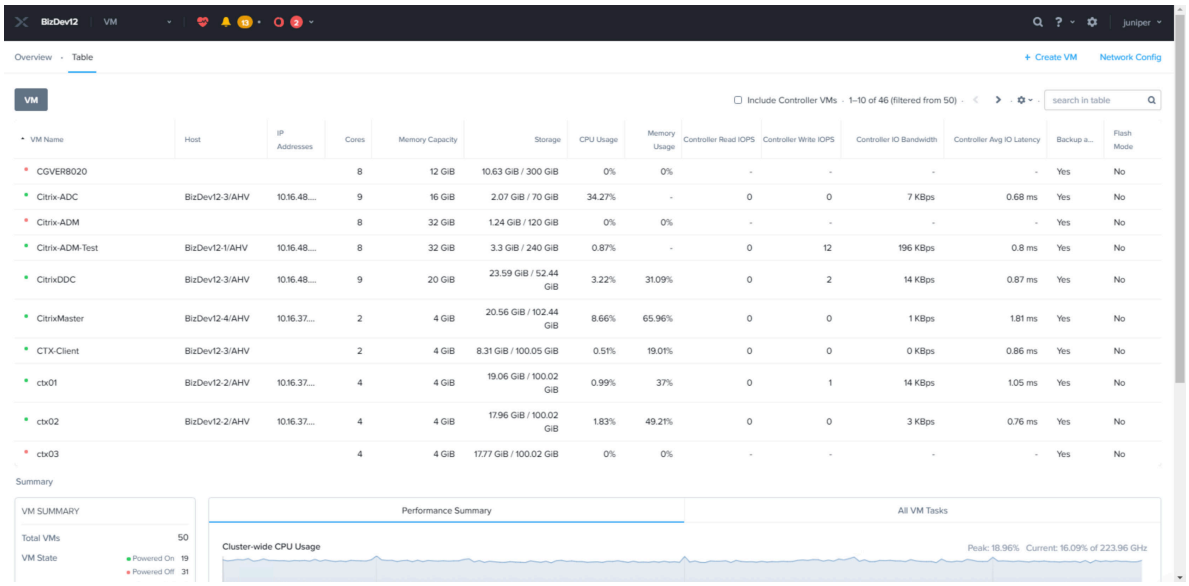
1. Click the **Home** menu at the top of the Prism home page and select the **VM** option from the drop-down list as as shown in Figure 69 on page 300.

Figure 69: VM Option Page



2. To create a VM, select the **VM** option under the Home tab (top-left corner) and click **+ Create VM** at the top-right side of the VM page as shown in Figure 70 on page 301.

Figure 70: VM Page



The Create VM page appears as shown in Figure 71 on page 302.

3. On the Create VM page, provide details of the indicated fields to create a vSRX VM as shown in Figure 71 on page 302 and click the **Save** button.

   - Name: Enter a name for the VM.
   - Description (optional): Enter a description for the VM.
   - vCPU(s): Enter the number of virtual CPUs to allocate to this VM.
   - Number of Cores per vCPU: Enter the number of cores assigned to each virtual CPU.
   - Memory: Enter the amount of memory to allocate to this VM.

- Select the time zone and update the compute details.

Figure 71: Create VM Page

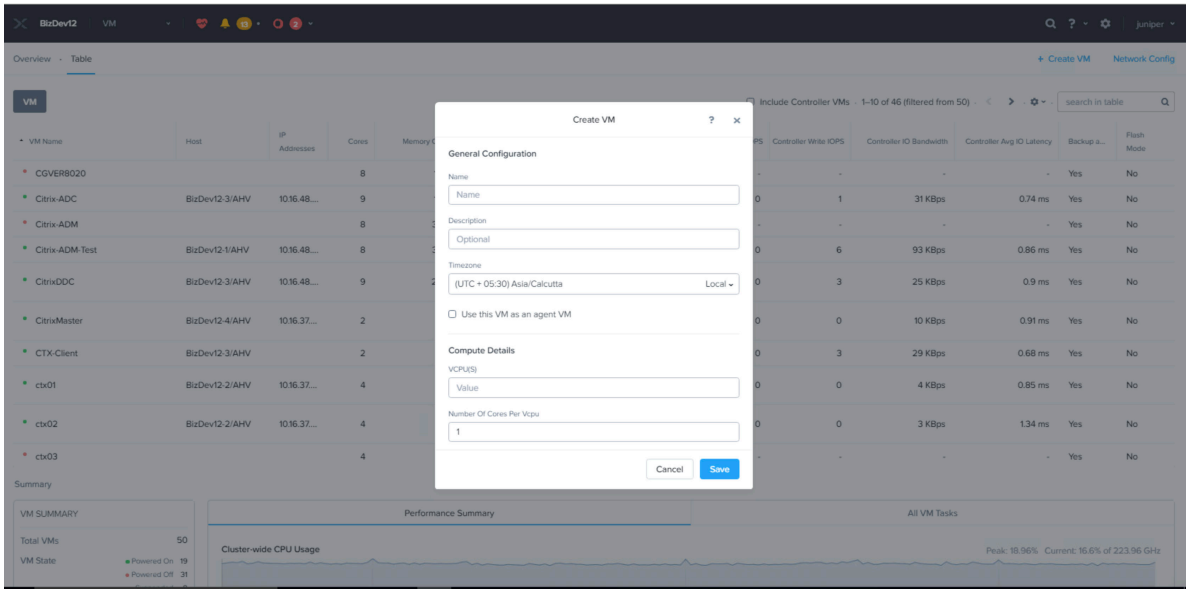
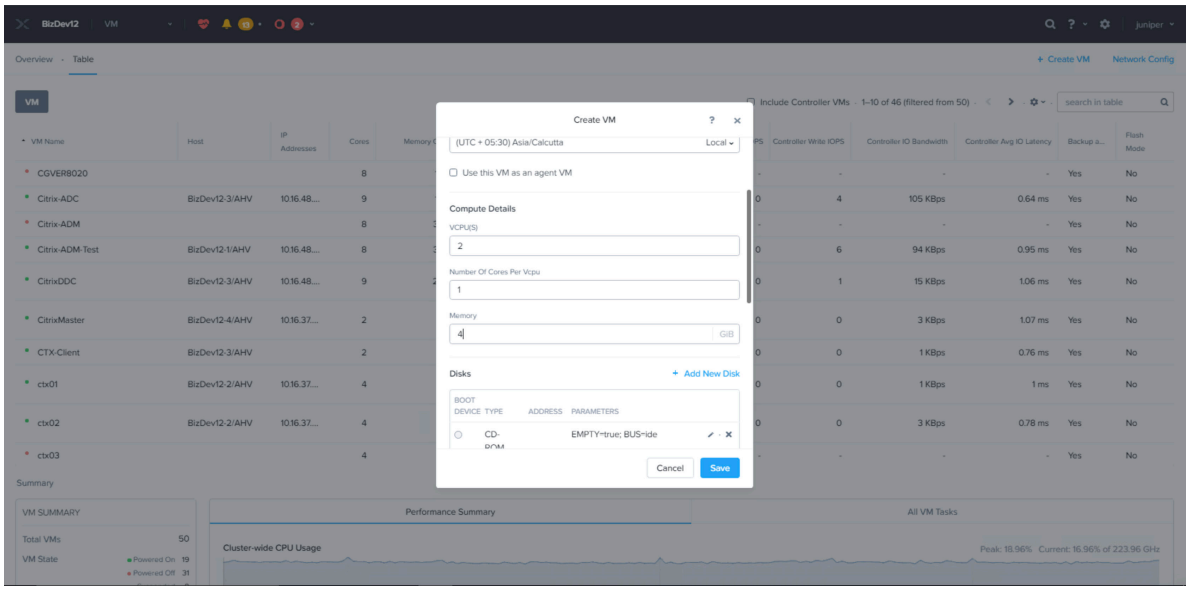
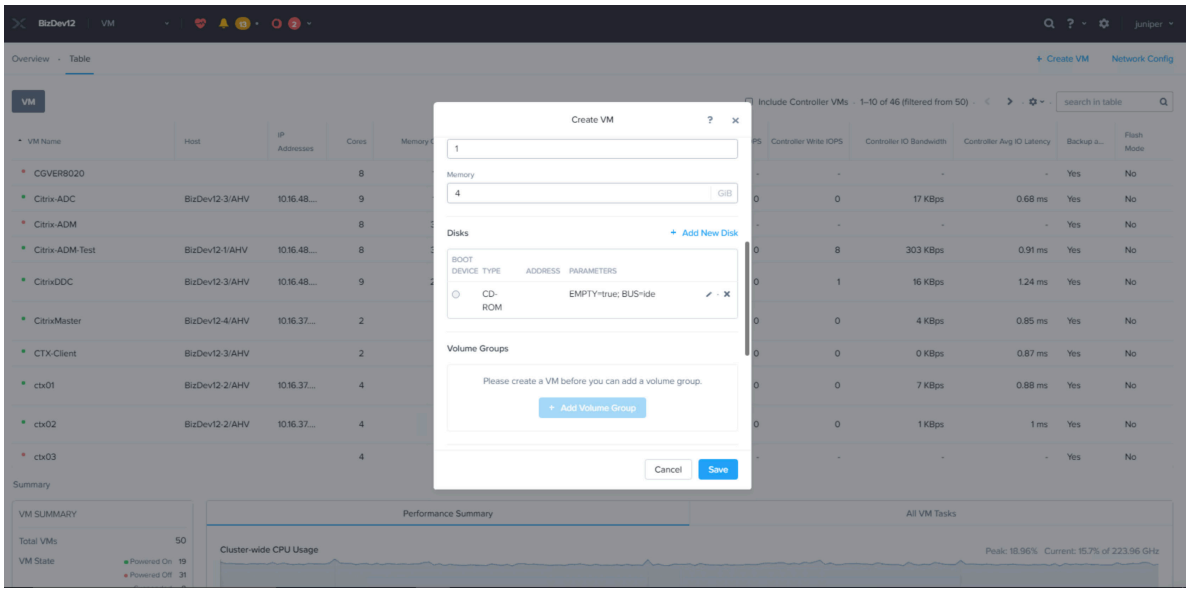


Figure 72: VM Compute Details Page



4. To attach a disk to the vSRX VM, click the **+ Add New Disk** option on the **Create VM** page as shown in Figure 73 on page 303.

Figure 73: VM Disk Details Page



5. The **Add Disk** page appears as shown in Figure 74 on page 304. Select the vSRX Junos Image.

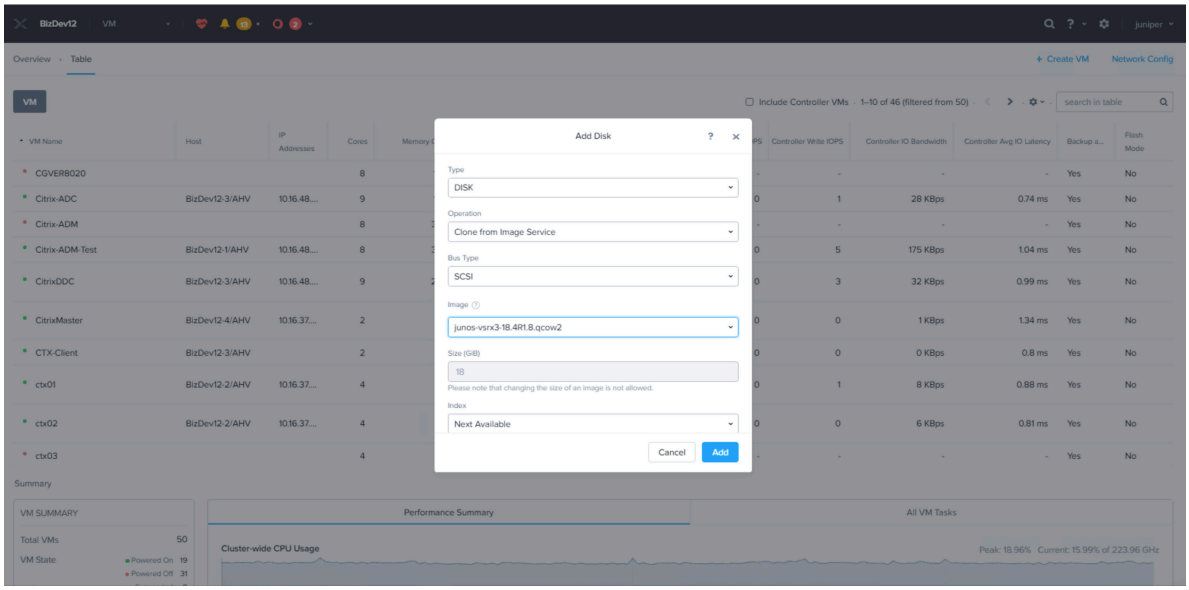
   Do the following in the indicated fields and click on the **Add** button:

   - Type: Select the type of storage device, **DISK** or **CDROM**, from the drop-down list. The following fields and options vary depending on whether you choose DISK or CDROM.
   - Operation: Specify the device contents from the drop-down list.
     - Select **Clone from ADSF file** to copy any file from the cluster that can be used as an image onto the disk.
     - Select **Empty CDROM** to create a blank CD device. (This option appears only when CD is selected in the previous field.) A CD device is needed.
     - Select **Allocate on Container** to allocate space without specifying an image. (This option appears only when DISK is selected in the previous field.) Selecting this option means you are allocating space only. You have to provide a system image later from a CD or other source.
     - Select **Clone from Image Service** to copy an image that you have imported by using the image service feature onto the disk.
   - Bus Type: Select the bus type from the drop-down list. The choices are IDE, SCSI, or SATA.
   - Path: Enter the path to the desired system image.

> **NOTE**: Field for entering the path appears only when Clone from ADSF file is selected. This file specifies the image to copy. For example, enter the pathname as /container_name/iso_name.iso. For example to clone an image from myos.iso in a container named crt1, enter /crt1/myos.iso. When a user types the container name (/container_name/), a list appears of the ISO files in that container (If one or more ISO files had previously been copied to that container).

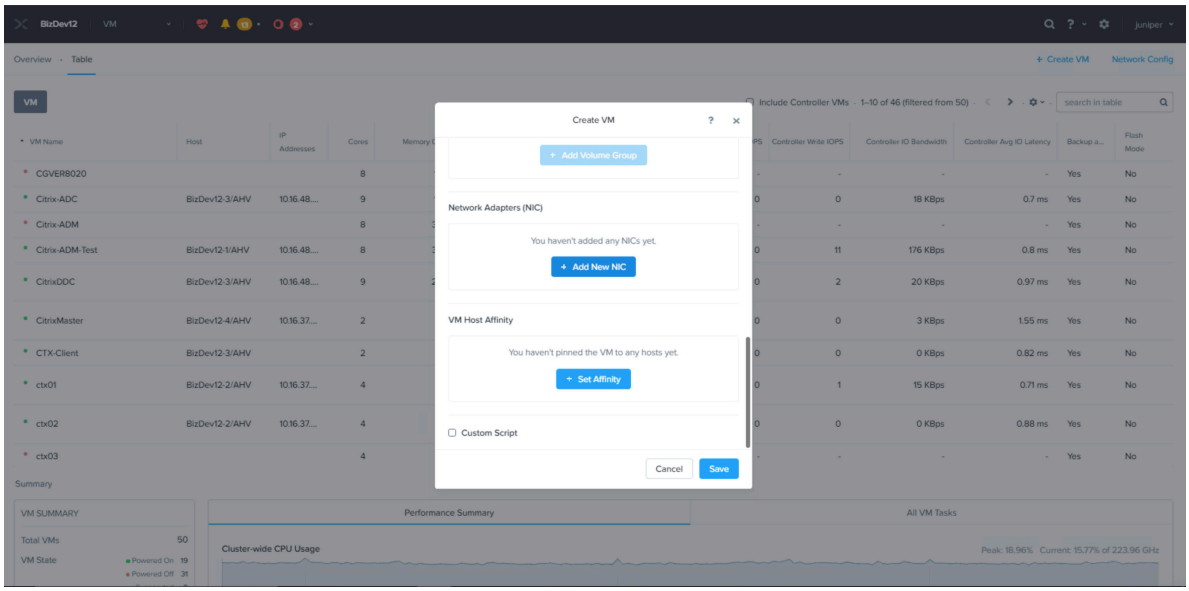
- Image: Select the image that you have created by using the image service feature. This field appears only when Clone from Image Service is selected. This field specifies the image to copy.
- Size: Enter the disk size in GiBs. This field appears only when Allocate on Container is selected.
- When all the field entries are correct, click the **Add** button to attach the disk to the VM and return to the Create VM page.
- Repeat Step 5 to attach additional devices to the VM.

Figure 74: Add Disk Details Page



6. To create a network interface for the vSRX VM, click the **+ Add New NIC** option in the Create VM page as shown in Figure 75 on page 305. Add the NICs required.

Figure 75: Add New NIC Option

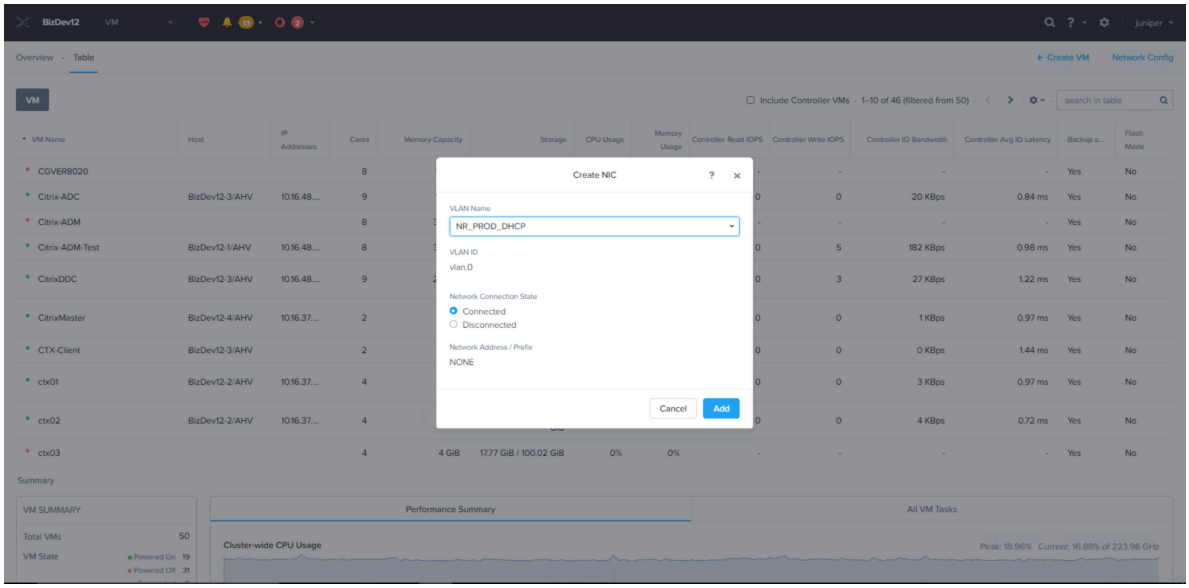


The Create NIC page appears as shown in Figure 76 on page 306. Do the following in the indicated fields:

- VLAN Name: Select the target virtual LAN from the drop-down list.
- VLAN ID: This is a read-only field that displays the VLAN ID.
- VLAN UUID: This is a read-only field that displays the VLAN UUID.
- Network Address/Prefix: This is a read-only field that displays the network IP address and prefix.
- IP Address: Enter an IP address for the VLAN. This field appears only if the NIC is placed in a managed network. Entering an IP address in this field is optional when the network configuration provides an IP pool. If the field is left blank, the NIC is assigned an IP address from the pool.
- When all the field entries are correct, click the **Add** button to create a network interface for the VM and return to the Create VM page.

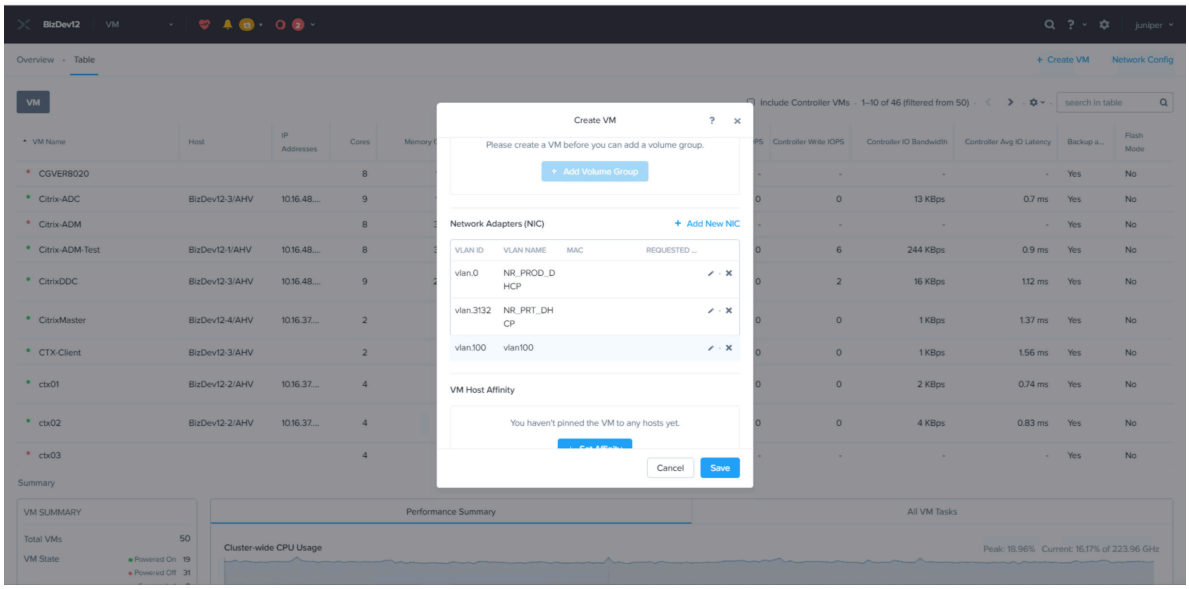
- Repeat this Step 6 to create additional network interfaces for the VM.

Figure 76: Create NIC Page



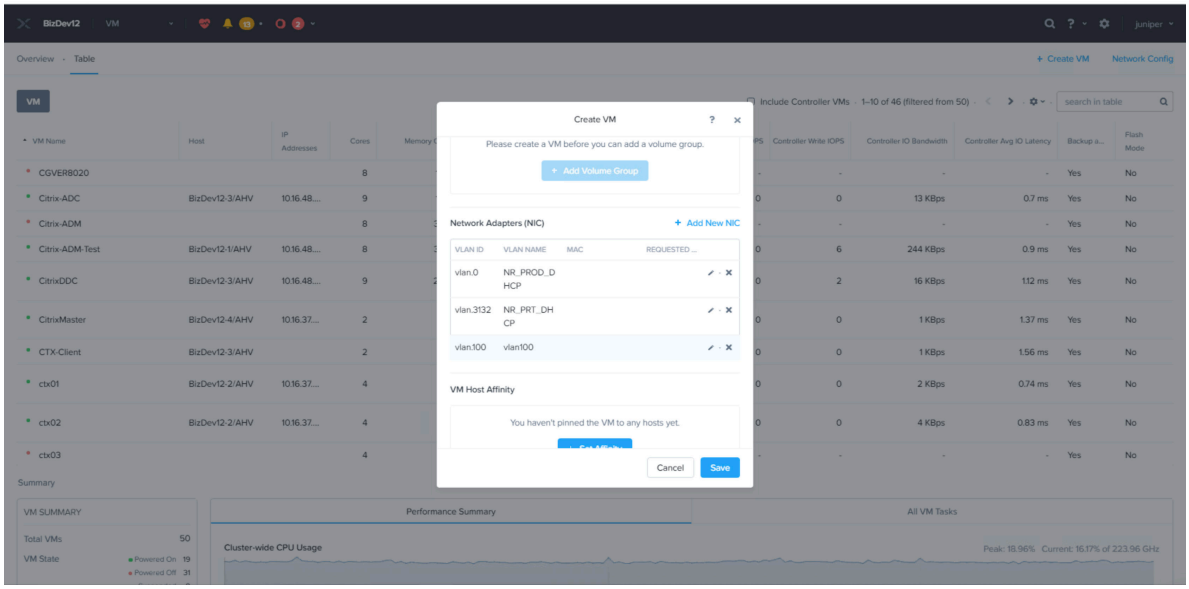
Repeat Step 6 and add more VLANs and NICs as needed.

Figure 77: Adding More VLANs and NICs



7. (Optional) If host affinity is needed, click **Set Affinity**..

Figure 78: VM Host Affinity Page



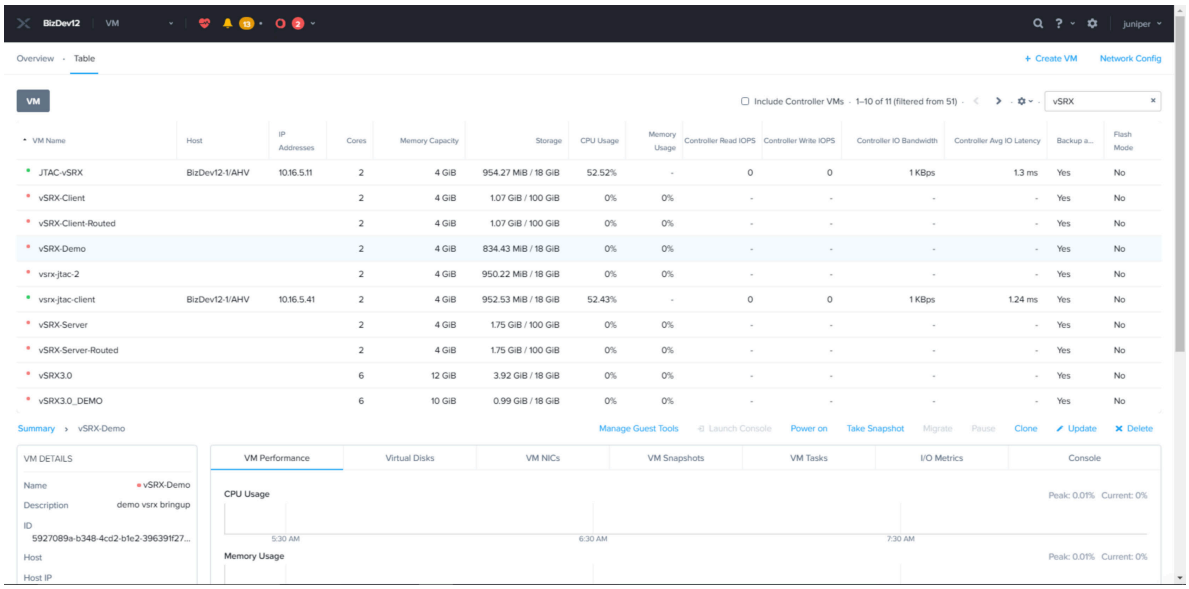
8. To customize the VM by using Cloud-init (for Linux VMs) or Sysprep (for Windows VMs), select the **Custom Script** check box.
9. When all the field entries are correct, click the **Save** button to create the VM and close the Create VM page.

Power on the vSRX VMs

This topic provides you details on how to power on vSRX VMs.

1. Use the Table drop-down list to search for VMs as shown in Figure 79 on page 308.

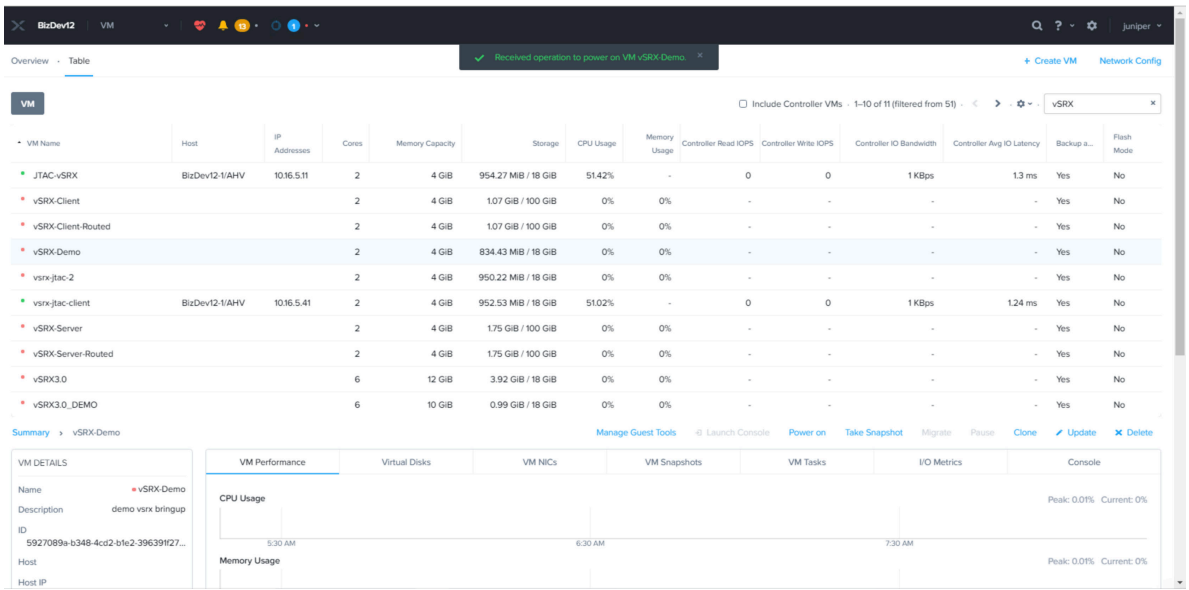
Figure 79: Powering on VMs



2. Click the **Power on** option (see Figure 79 on page 308) for each VM.

   All the VMs will turn on as shown in Figure 80 on page 308

Figure 80: Power on VM Confirmation Page

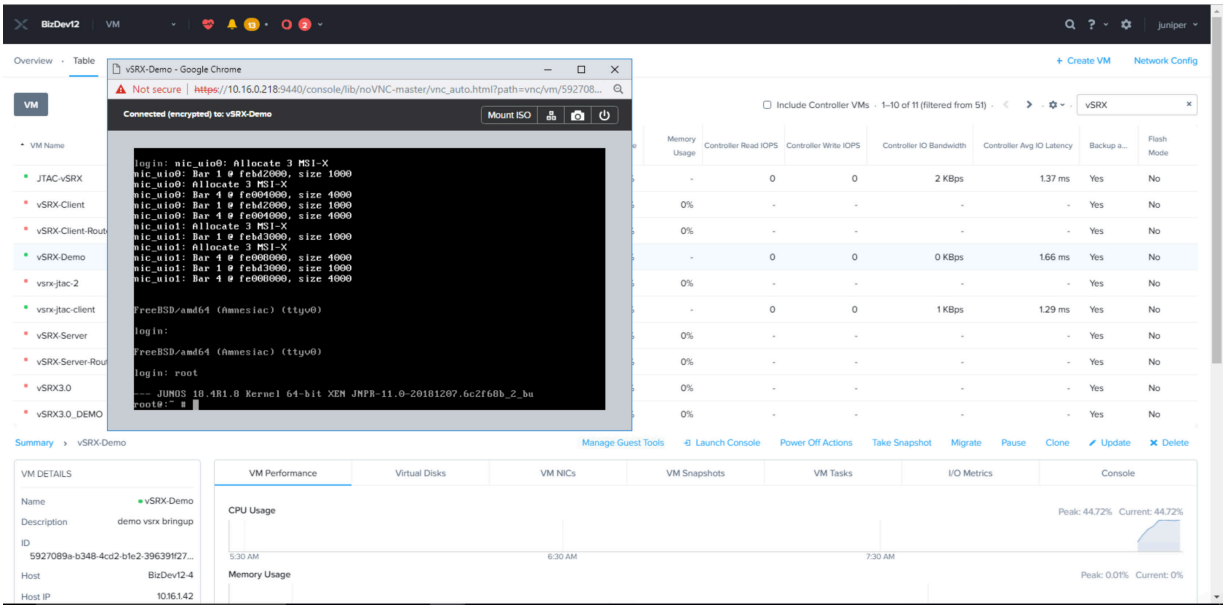


Launch vSRX VM Console

This topic explains how to launch the vSRX VM console.

Click the **Launch Console** option at the bottom of screenshot as shown in Figure 81 on page 309 to launch the VM console.

Figure 81: Launch Console Page



RELATED DOCUMENTATION

- Day One: vSRX on KVM

Upgrade the Junos OS for vSRX Software Release

You can upgrade the Junos OS for vSRX software using the CLI. Upgrading or downgrading Junos OS can take several hours, depending on the size and configuration of the network. Download the desired Junos OS Release for the vSRX 3.0 upgrade tgz file from the Juniper Networks website. Example filename is **junos-install-vsrx3-x86-64-xxxxx.tgz**.

You also can upgrade using J-Web (see J-Web) or the Junos Space Network Management Platform (see Junos Space).

For the procedure on upgrading a specific Junos OS for vSRX software release, see the *Migration, Upgrade, and Downgrade Instructions* topic in the release-specific *vSRX Release Notes* available on the vSRX TechLibrary webpage.

PART 6

vSRX Deployment for AWS

CHAPTER 18

Overview

IN THIS CHAPTER

Understand vSRX with AWS

IN THIS SECTION

This section presents an overview of vSRX on Amazon Web Services (AWS).

vSRX with AWS

AWS provides on-demand services in the cloud. Services range from Infrastructure as a Service (IaaS) and Platform as a Service (SaaS), to Application and Database as a Service. AWS is a highly flexible, scalable, and reliable cloud platform. In AWS, you can host servers and services on the cloud as a pay-as-you-go (PAYG) or bring-your-own-license (BYOL) service.

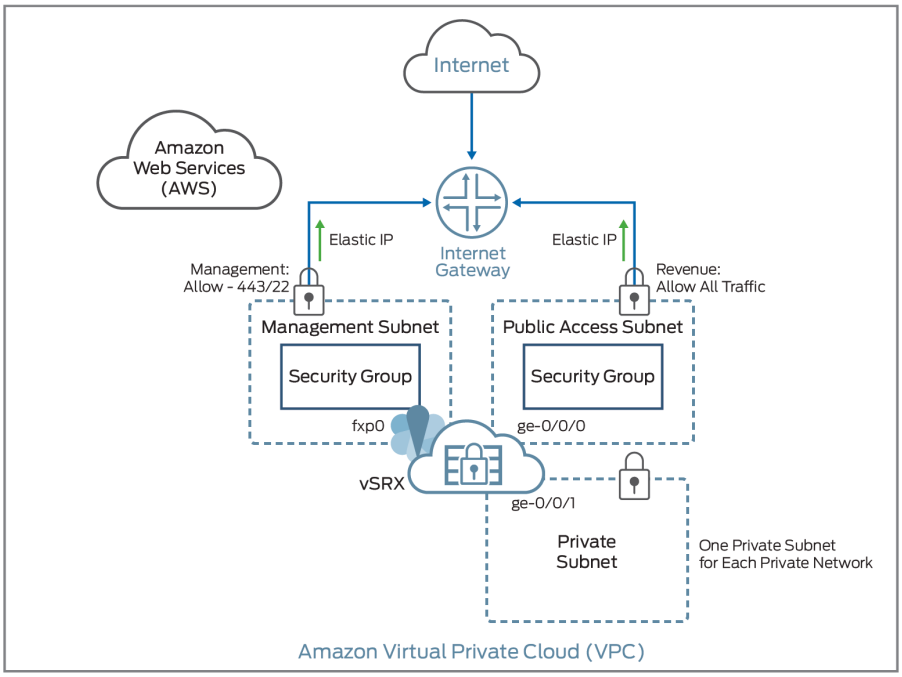
NOTE: vSRX PAYG images do not require any Juniper Networks licenses.

vSRX can be deployed in a virtual private cloud (VPC) in the Amazon Web Services (AWS) cloud. You can launch vSRX as an Amazon Elastic Compute Cloud (EC2) instance in an Amazon VPC dedicated to a specific user account. The vSRX Amazon Machine Image (AMI) uses hardware virtual machine (HVM) virtualization.

Figure 82 on page 313 shows an example of deploying a vSRX instance to provide security for applications running in a private subnet of an Amazon VPC.

In the Amazon VPC, public subnets have access to the Internet gateway, but private subnets do not. vSRX requires two public subnets and one or more private subnets for each individual instance group. The public subnets consist of one for the management interface (fxp0) and one for a revenue (data) interface. The private subnets, connected to the other vSRX interfaces, ensure that all traffic between applications on the private subnets and the Internet must pass through the vSRX instance.

Figure 82: vSRX in AWS Deployment



AWS Marketplace also enables you to discover and subscribe to software that supports regulated workloads through AWS Marketplace for AWS GovCloud (US).

Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, vSRX supports two bundles for PAYG that are available as 1-hour or 1-year subscriptions.

- vSRX Next Generation Firewall—Includes standard (STD) features of core security, including core firewall, IPsec VPN, NAT, CoS, and routing services, as well as advanced Layer 4 through 7 security services such as AppSecure features of AppID, AppFW, AppQoS, and AppTrack, IPS and rich routing capabilities.
- vSRX Premium-Next Generation Firewall with Anti-Virus Protection—Includes the features in the vSRX Next- Generation Firewall package, including the UTM antivirus feature.

You deploy vSRX in an Amazon Virtual Private Cloud (Amazon VPC) as an application instance in the Amazon Elastic Compute Cloud (Amazon EC2). Each Amazon EC2 instance is deployed, accessed, and configured over the Internet using the AWS Management Console, and the number of instances can be scaled up or down as needed.

NOTE: In the current release, each vSRX instance uses two vCPUs and 4 GB of memory, even if the instance type selected on AWS provides more resources.

vSRX uses hardware assisted virtual machines (HVM) for high performance (enhanced networking), and supports the following deployments on AWS cloud environments:

- As a firewall between other Amazon EC2 instances on your Amazon VPC and the Internet
- As a VPN endpoint between your corporate network and your Amazon VPC
- As a firewall between Amazon EC2 instances on different subnets

There are default limits for AWS services for an AWS account. For more information on AWS service limits, see https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html and https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html .

AWS Glossary

This section defines some common terms used in an AWS configuration. Table 60 on page 314 defines common terms used for Amazon Virtual Private Cloud (Amazon VPC) and Table 61 on page 316 defines common terms for Amazon Elastic Compute Cloud (Amazon EC2) services.

Table 60: Amazon VPC Related Terminology

Term	Description
Internet gateways	Amazon VPC components that allow communications between your instances in the Amazon VPC and the Internet.

Table 60: Amazon VPC Related Terminology *(Continued)*

Term	Description
IP addressing	AWS includes three types of IP address: • Public IP address–Addresses obtained from a public subnet that is publicly routable from the Internet. Public IP addresses are mapped to primary private IP addresses through AWS NAT. • Private IP address–IP addresses in the Amazon VPC Classless Interdomain Routing (CIDR) range, as specified in RFC 1918, that are not publicly routable. • Elastic IP address–A static IP address designed for dynamic cloud computing. When an Elastic IP address is associated with a public IP network interface, the public IP address associated is released until the Elastic IP address is disassociated from the network interface. Each network interface can be associated with multiple private IP addresses. Public subnets can have multiple private IP addresses, public addresses, and Elastic IP addresses associated with the private IP address of the network interface. Instances in private and public subnets can have multiple private IP addresses. One Elastic IP address can be associated with each private IP address for instances in public subnets. You can assign static private IP addresses in the subnet. The first five IP addresses and the last IP address in the subnet are reserved for Amazon VPC networking and routing. The first IP address is the gateway for the subnet.
Network ACL	AWS stateless virtual firewall operating at the subnet level.
Route tables	A set of routing rules used to determine where the network traffic is directed. Each subnet needs to be associated with a route table. Subnets not explicitly associated with a route table are associated with the main route table. Custom route tables can be created other than the default table.

Table 60: Amazon VPC Related Terminology *(Continued)*

Term	Description
Subnet	A virtual addressing space in the Amazon VPC CIDR block. The IP addresses for the Amazon EC2 instances are allocated from the subnet pool of IP addresses. You can create two types of subnets in the Amazon VPC: • Public subnets–Subnets that have traffic connections to the Internet gateway. • Private subnets–Subnets that do not have connections to the Internet gateway **NOTE**: With vSRX Network Address Translation (NAT) , you can launch all customer instances in private subnets and connect vSRX interfaces to the Internet. This protects customer instances from being directly exposed to Internet traffic.
VPC	Virtual private cloud.

Table 61: Amazon EC2 Related Terminology

Term	Description
Amazon Elastic Block Store (EBS)	Persistent block storage that can be attached to an Amazon EC2 instance. Block storage volumes can be formatted and mounted on an instance. Amazon EBS optimized instances provide dedicated throughput between Amazon EC2 and Amazon EBS.
Amazon Elastic Compute Cloud (EC2)	Amazon Web service that enables launch and management of elastic virtual servers or computers that run on the Amazon infrastructure.
Amazon Machine Image (AMI)	Amazon image format that contains the information, such as the template for root volume, launch permissions, and block device mapping, that is required to launch an Amazon EC2 instance.
Elastic IP	A static IP designed for dynamic cloud computing. The public IP is mapped to the privet subnet IP using NAT.
Enhanced networking	Provides high packet per second performance, low latency, higher I/O performance, and lower CPU utilization compared to traditional implementations. vSRX leverages this networking with hardware virtualized machine (HVM) Amazon Machine Images (AMIs).

Table 61: Amazon EC2 Related Terminology *(Continued)*

Term	Description
Instance	A virtual machine or server on Amazon EC2 that uses XEN or, XEN-HVM hypervisor types. Amazon EC2 provides a selection of instances optimized for different use cases.
Key pairs	Public key cryptography used by AWS to encrypt and decrypt login information. Create these key pairs using AWS-EC2 or import your own key pairs. **NOTE**: AWS does not accept DSA. Limit the public key access permissions to 400. For more information on key rotation, see https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html.
Network interfaces	Virtual network interfaces that you can attach to an instance in the Amazon VPC. An Elastic Network Interface (ENI) can have a primary private IP address, multiple secondary IP addresses, one Elastic IP address per private IP address, one public IP address, one or more security groups, one MAC address, and a source/destination check flag. For vSRX instances, disable the source/destination check for all interfaces. **NOTE**: ENIs use the IP addresses within the subnet range. So, the ENI IP addresses are not exhausted.
Network MTU	All Amazon instance types support an MTU of 1500. Some instance types support jumbo frames (9001 MTU). **NOTE**: Use C3, C4, C5, CC2, M3, M4, or T2 AWS instance types for vSRX instances with jumbo frames.
Placement Groups	Instances launched in a common cluster placement group. Instances within the cluster have networks with high bandwidth and low latency.
Security groups	An AWS-provided virtual firewall that controls the traffic for one or more instances. Security groups can be associated with an instance only at launch time. **NOTE**: Because vSRX manages your firewall settings, we recommend that you ensure there is no contradiction between rule sets on AWS security groups and rule sets in your vSRX configuration.

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, vSRX supports two bundles for PAYG that are available as 1-hour or 1-year subscriptions.

RELATED DOCUMENTATION

- AWS Tutorials
- Getting Started with AWS

Requirements for vSRX on AWS

IN THIS SECTION

This section presents an overview of requirements for deploying a vSRX instance on Amazon Web Services (AWS).

Minimum System Requirements for AWS

Table 62 on page 318 lists the minimum system requirements for vSRX instances to be deployed on AWS.

Table 62: Minimum System Requirements for vSRX

Component	Specification and Details
Hypervisor support	XEN-HVM

Table 62: Minimum System Requirements for vSRX *(Continued)*

Component	Specification and Details
Memory	4 GB
Disk space	16 GB
vCPUs	2
vNICs	3
vNIC type	SR-IOV

Interface Mapping for vSRX on AWS

vSRX on AWS supports up to a maximum of eight network interfaces, but the actual maximum number of interfaces that can be attached to a vSRX instance is dictated by the AWS instance type in which it is launched. For AWS instances that allow more than eight interfaces, vSRX will support up to a maximum of eight interfaces only.

For more information on maximum network interfaces by instance type, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html .

Table 63 on page 319 shows a mapping between vSRX interface names and their corresponding AWS interface names for up to eight network interfaces. The first network interface is used for the out-of-band management (fxp0) for vSRX.

Table 63: vSRX and AWS Interface Names

Interface Number	vSRX Interface	AWS Interface
1	fxp0	eth0
2	ge-0/0/0	eth1

Table 63: vSRX and AWS Interface Names *(Continued)*

Interface Number	vSRX Interface	AWS Interface
3	ge-0/0/1	eth2
4	ge-0/0/2	eth3
5	ge-0/0/3	eth4
6	ge-0/0/4	eth5
7	ge-0/0/5	eth6
8	ge-0/0/6	eth7

We recommend putting revenue interfaces in routing instances as a best practice to avoid asymmetric routing. Since fxp0 is part of the default (inet.0) routing table, there might be two default routes needed in the same routing instance: one for the fxp0 interface for external management access, and the other for the revenue interfaces for traffic access, resulting in asymmetric routing. Putting the revenue interfaces in a separate routing instance avoids this situation of two default routes in a single routing instance.

NOTE: Ensure that interfaces belonging to the same security zone are in the same routing instance. See KB Article - Interface must be in the same routing instance as the other interfaces in the zone.

vSRX Default Settings on AWS

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.
- The ENA driver-related component must be ready for vSRX.

Table 64 on page 321 lists the factory-default settings for security policies on the vSRX.

Table 64: Factory-Default Settings for Security Policies

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

CAUTION: Do not use the `load factory-default` command on a vSRX AWS instance. The factory-default configuration removes the AWS preconfiguration. If you must revert to factory default, ensure that you manually reconfigure AWS preconfiguration statements before you commit the configuration; otherwise, you will lose access to the vSRX instance. See *Configure vSRX Using the CLI* for AWS preconfiguration details.

Best Practices for Improving vSRX Performance

Review the following deployment practices to improve vSRX performance:

- Disable the source/destination check for all vSRX interfaces.
- Limit public key access permissions to 400 for key pairs.
- Ensure that there are no contradictions between AWS security groups and your vSRX configuration.
- Use the c5n instance types on AWS for best throughput on the vSRX.
- Ensure traffic flows through multiple interfaces of the vSRX for optimal usage of the vCPUs.
- Use vSRX NAT to protect your Amazon EC2 instances from direct Internet traffic.

CHAPTER 19

Configure and Manage vSRX in AWS

IN THIS CHAPTER

Configure an Amazon Virtual Private Cloud for vSRX

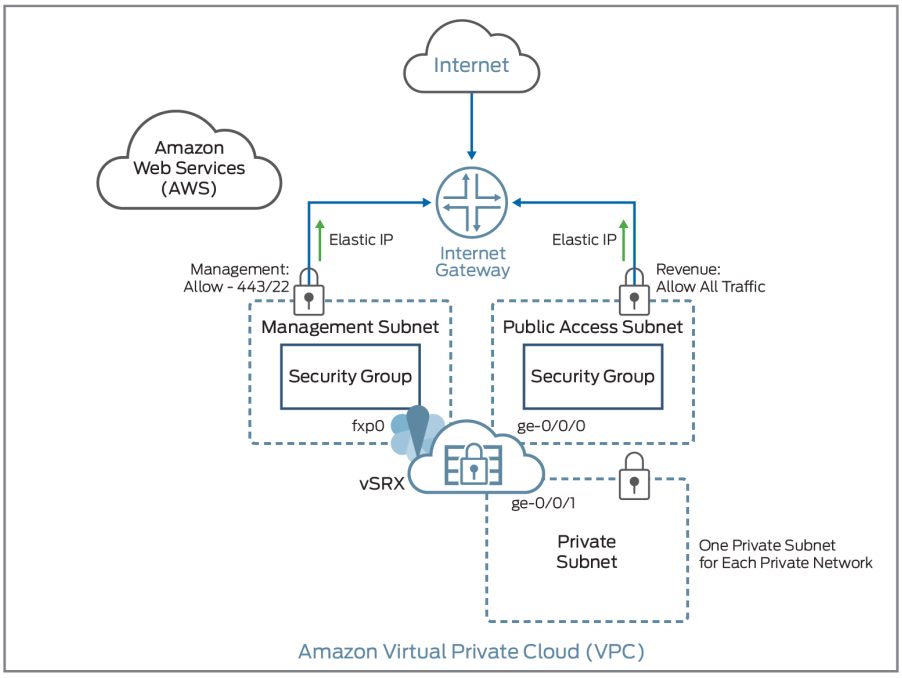
IN THIS SECTION

Before you begin, you need an Amazon Web Services (AWS) account and an Identity and Access Management (IAM) role, with all required permissions to access, create, modify, and delete Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (S3), and Amazon Virtual Private

Cloud (Amazon VPC) objects. You should also create access keys and corresponding secret access keys, X.509 certificates, and account identifiers. For better understanding of AWS terminologies and their use in vSRX AWS deployments, see *Understand vSRX with AWS*.

Figure 83 on page 323 shows an example of how you can deploy vSRX to provide security for applications running in a private subnet of an Amazon VPC.

Figure 83: Example of vSRX in AWS Deployment

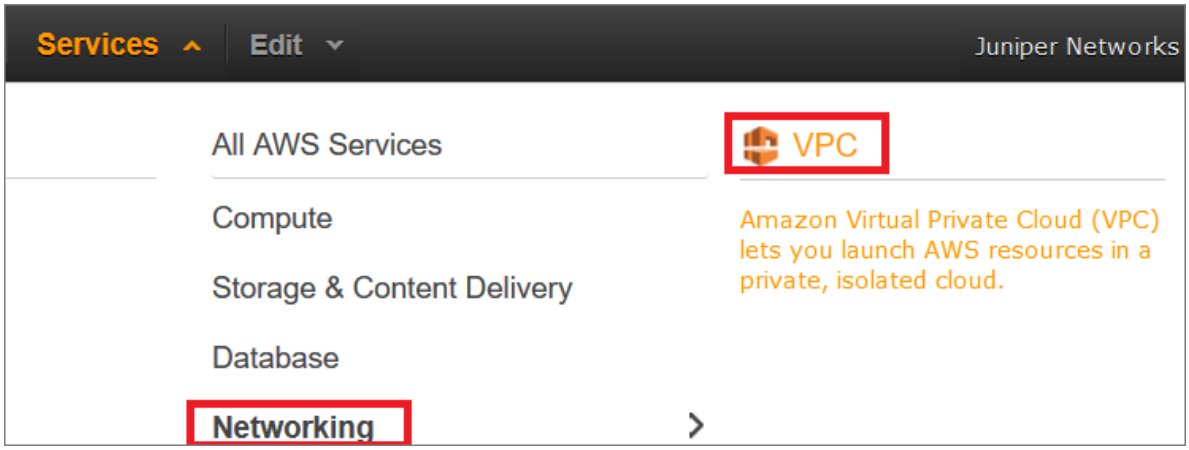


The following procedures outline how to create and prepare an Amazon VPC for vSRX. The procedures describe how to set up an Amazon VPC with its associated Internet gateway, subnets, route table, and security groups.

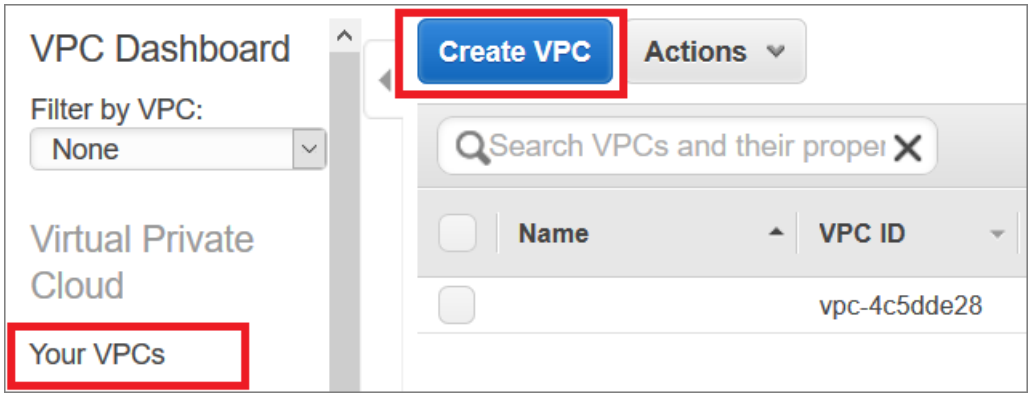
Step 1: Create an Amazon VPC and Internet Gateway

Use the following procedure to create an Amazon VPC and an Internet gateway. If you have already have a VPC and an Internet gateway, go to "Step 2: Add Subnets for vSRX" on page 325.

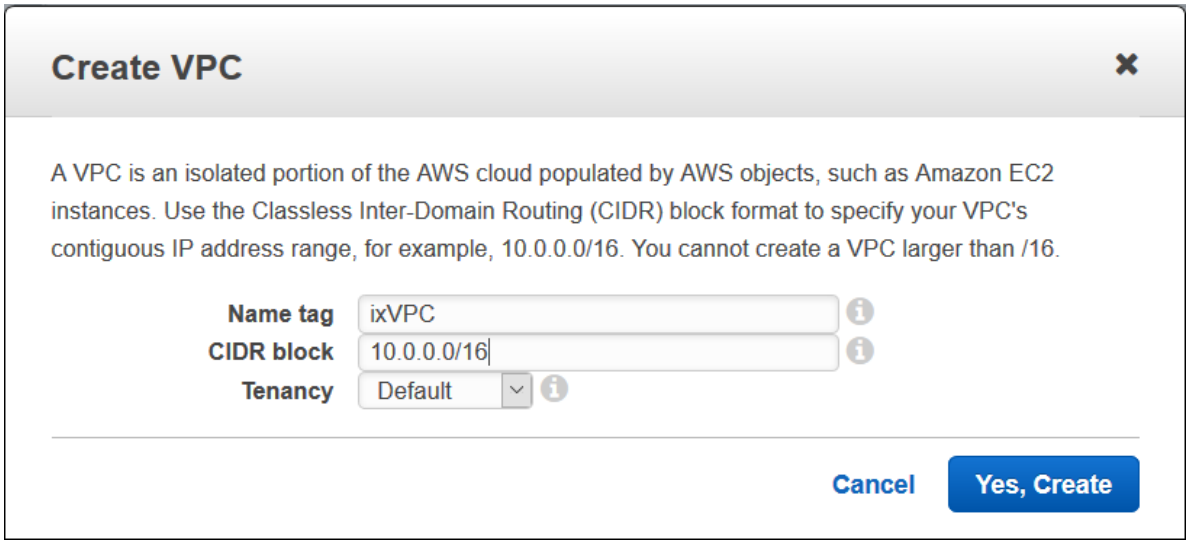
1. Log in to the AWS Management Console and select **Services > Networking > VPC**.



2. In the VPC Dashboard, select **Your VPCs** in the left pane, and click **Create VPC**.



3. Specify a VPC name and a range of private IP addresses in Classless Interdomain Routing (CIDR) format. Leave Default as the Tenancy.



4. Click **Yes, Create**.
5. Select **Internet Gateways** in the left pane, and click **Create Internet Gateway**.

6. Specify a gateway name and click **Yes, Create**.
7. Select the gateway you just created and click **Attach to VPC**.
8. Select the new Amazon VPC, and click **Yes, Attach**.

Step 2: Add Subnets for vSRX

In the Amazon VPC, public subnets have access to the Internet gateway, but private subnets do not. vSRX requires two public subnets and one or more private subnets for each individual instance group. The public subnets consist of one for the management interface (fxp0) and one for a revenue (data) interface. The private subnets, connected to the other vSRX interfaces, ensure that all traffic between applications on the private subnets and the Internet must pass through the vSRX instance.

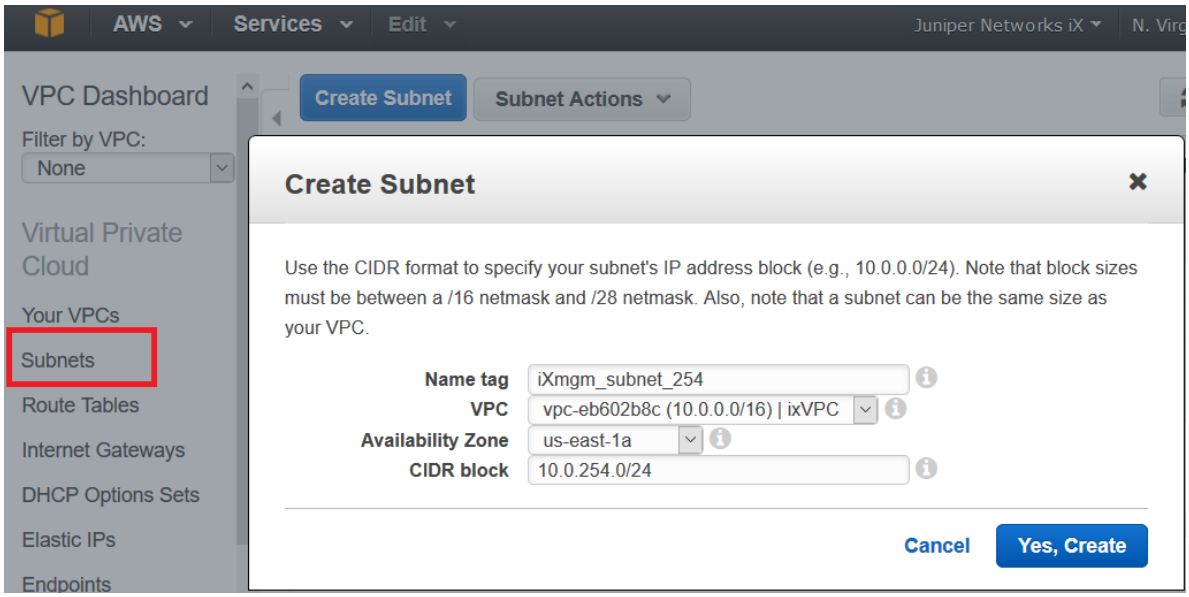
To create each vSRX subnet:

1. In the VPC Dashboard, select **Subnets** in the left pane, and click **Create Subnet**.
2. Specify a subnet name, select the Amazon VPC and availability zone, and specify the range of subnet IP addresses in CIDR format.

   TIP: As a naming convention best practice for subnets, we recommend including **private** or **public** in the name to make it easier to know which subnet is public or private.

   NOTE: All subnets for a vSRX instance must be in the same availability zone. Do not use **No Preference** for the availability zone.

3. Click **Yes, Create**.



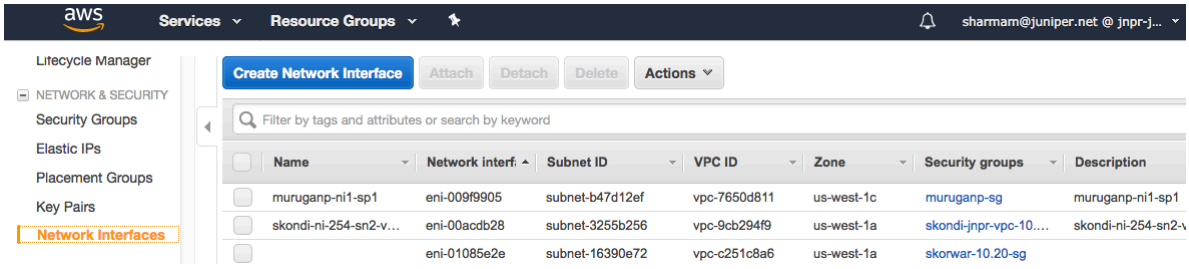
Repeat these steps for each subnet you want to create and attach to the vSRX instance.

Step 3: Attach an interface to a Subnet

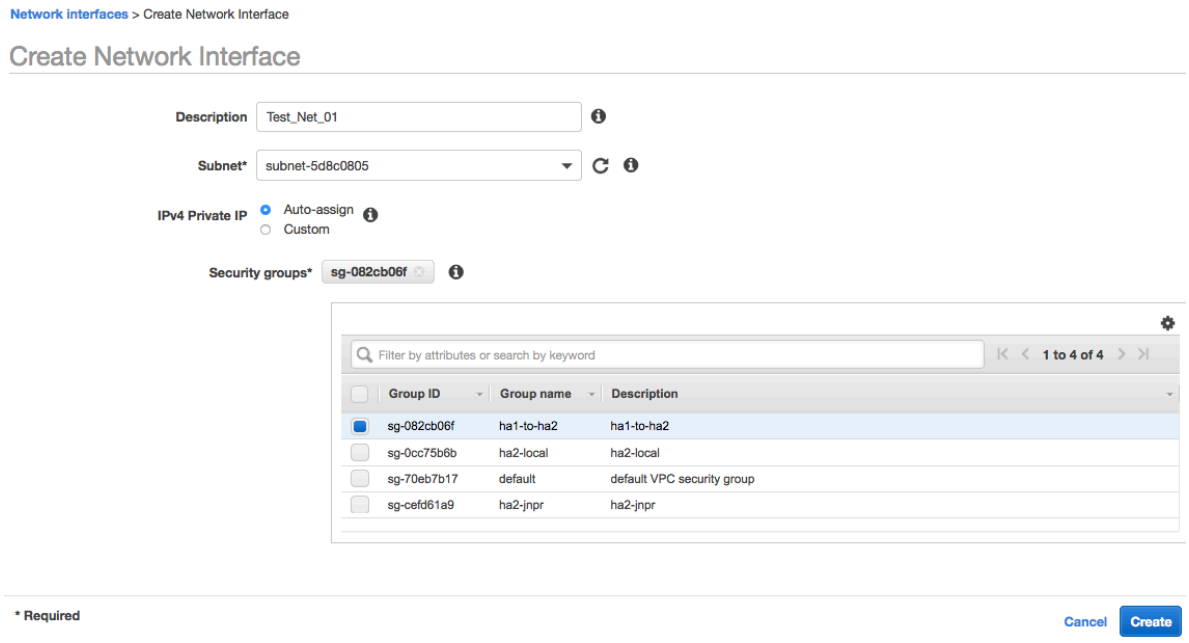
To attach an interface to a subnet:

1. Create a network interface from the Amazon EC2 home page.

   Click the Network Interface option on the EC2 home page and the **Create Network Interface** page opens.

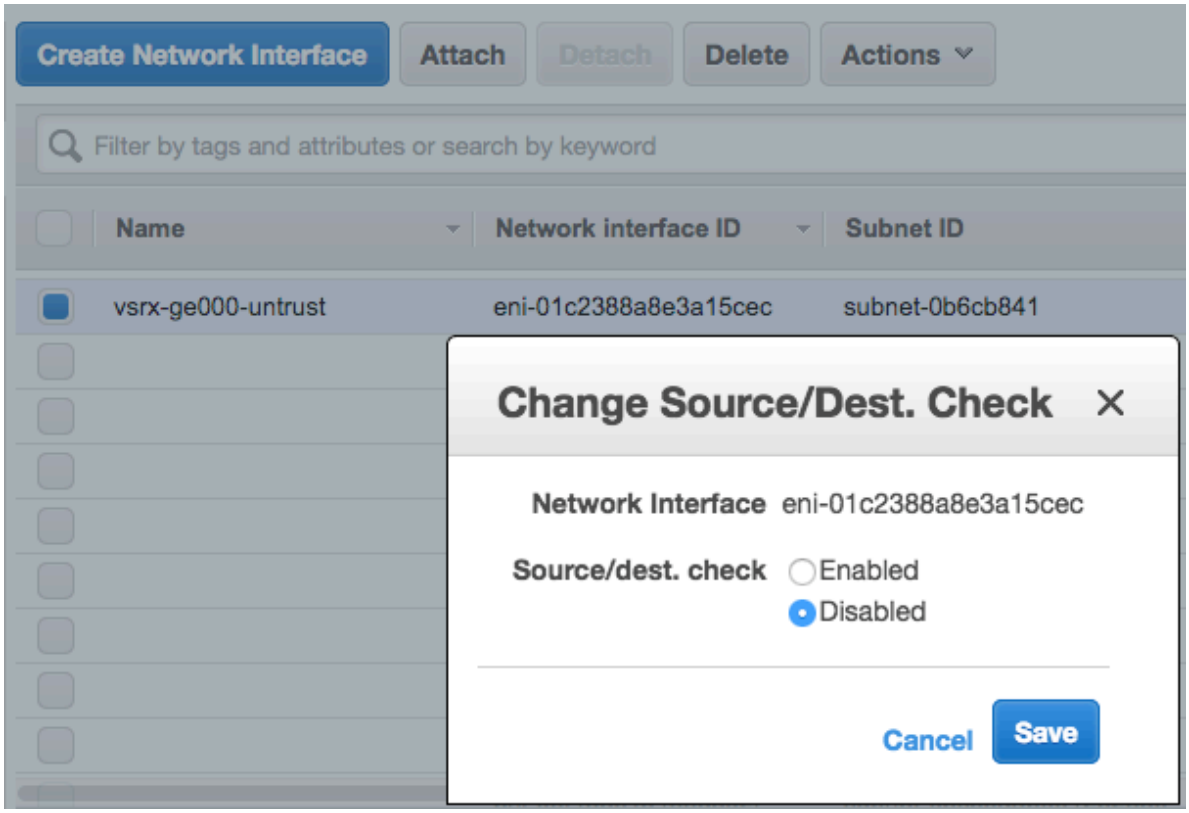


2. Click the **Create Network Interface** option, fill in the required information in the fields, and then click **Create**.

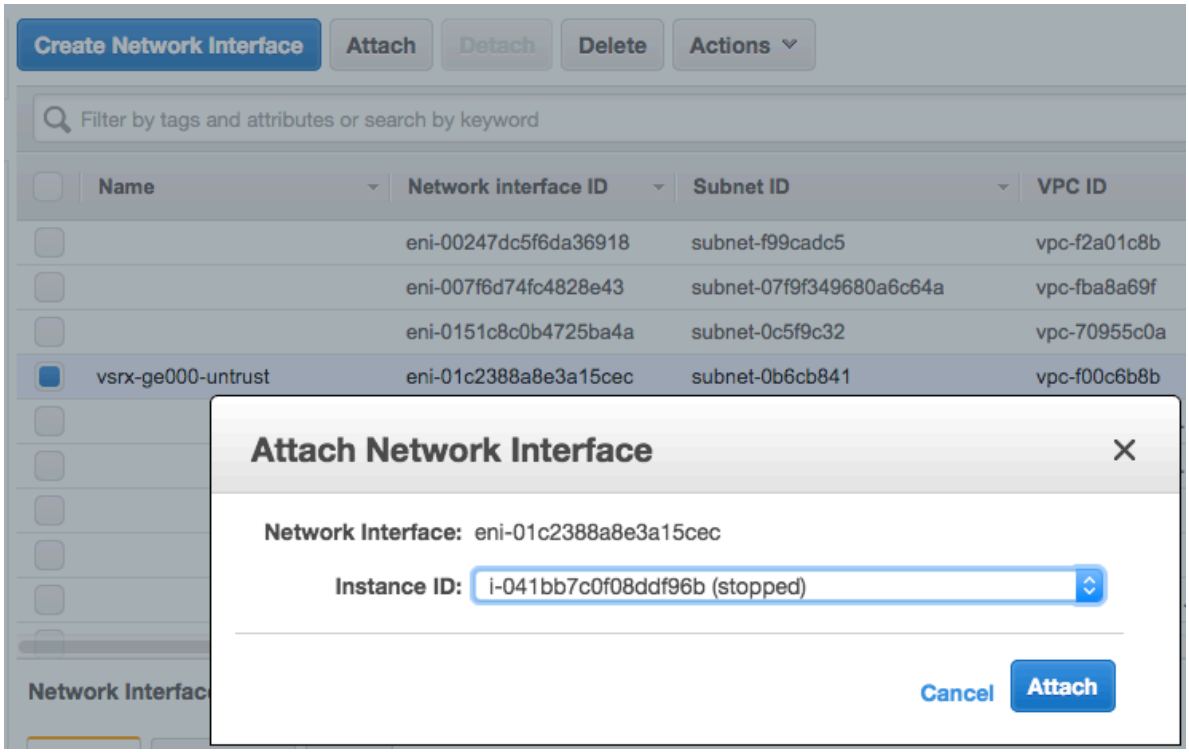


3. Find and select your newly created interface.

   If this interface is the revenue interface, then select **Change Source/Dest.Check** from the **Action** menu, choose **Disabled**, and click **Save**. If this interface is your fxp0 interface then skip this disabling step.



4. Click **Attach** from the menu on top of the screen, choose the **Instance ID** of your vSRX instance, and click **Attach**.



5. vSRX does not support interface hot plug-in. So, when you are done adding the interfaces, reboot the vSRX instances on which the interfaces were added, to apply the changes to take effect.

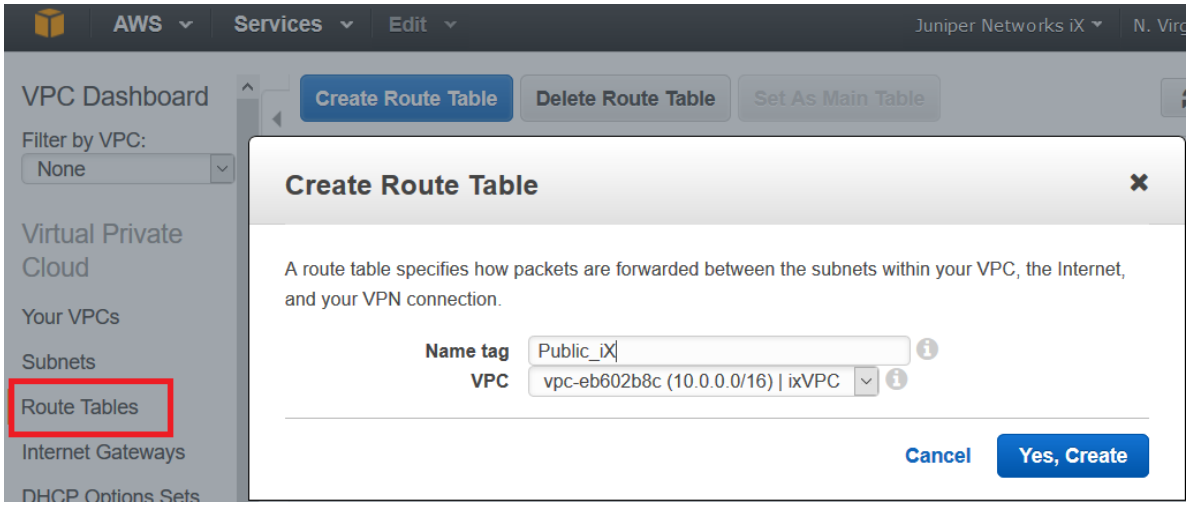
Step 4: Add Route Tables for vSRX

A main route table is created for each Amazon VPC by default. We recommend that you create a custom route table for the public subnets and a separate route table for each private subnet. All subnets that are not associated with a custom route table are associated with the main route table.

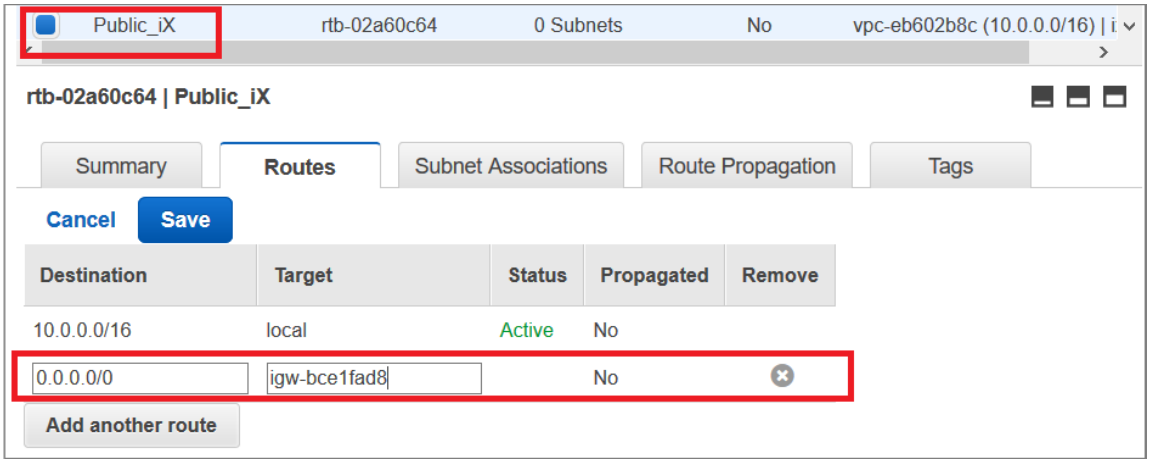
To create the route tables:

1. In the VPC Dashboard, select **Route Tables** in the left pane, and click **Create Route Table**.
2. Specify a route table name, select the VPC, and click **Yes, Create**.

> **TIP**: As a naming convention best practice for route tables, we recommend including **private** or **public** in the name to make it easier to know which route table is public or private.

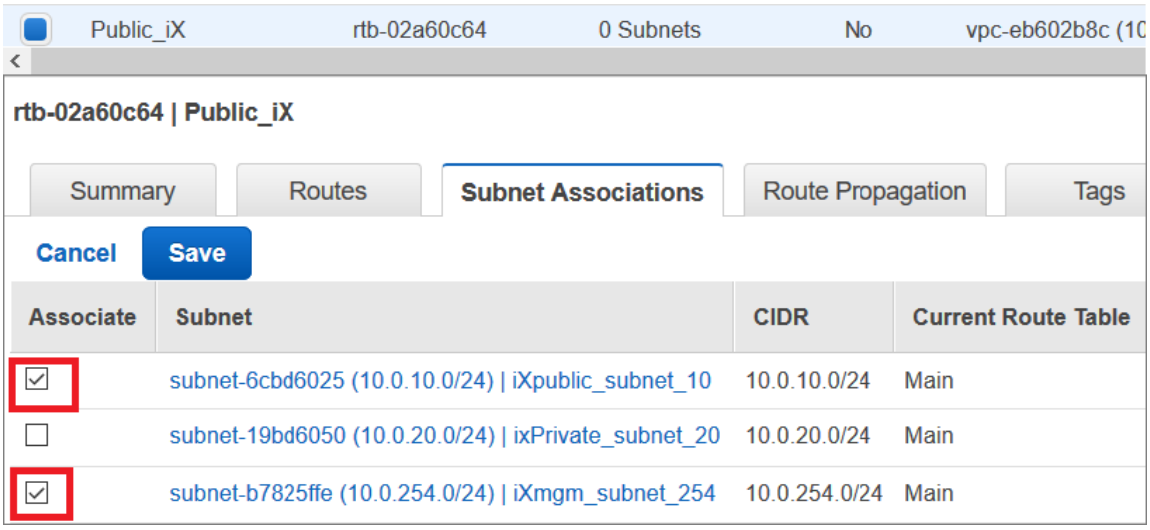


3. Repeat steps 1 and 2 to create all the route tables.
4. Select the route table you created for the public subnets and do the following:
   a. Select the **Routes** tab below the list of route tables.
   b. Click **Edit** and click **Add another route**.
   c. Enter **0.0.0.0/0** as the destination, select your VPC internet gateway as the target, and click **Save**.



d. Select the **Subnet Associations** tab, and click **Edit**.

e. Select the check boxes for the public subnets, and click **Save**.



5. Select each route table you created for a private subnet and do the following:

   a. Select the **Subnet Associations** tab, and click **Edit**.

   b. Select the check box for one private subnet, and click **Save**.

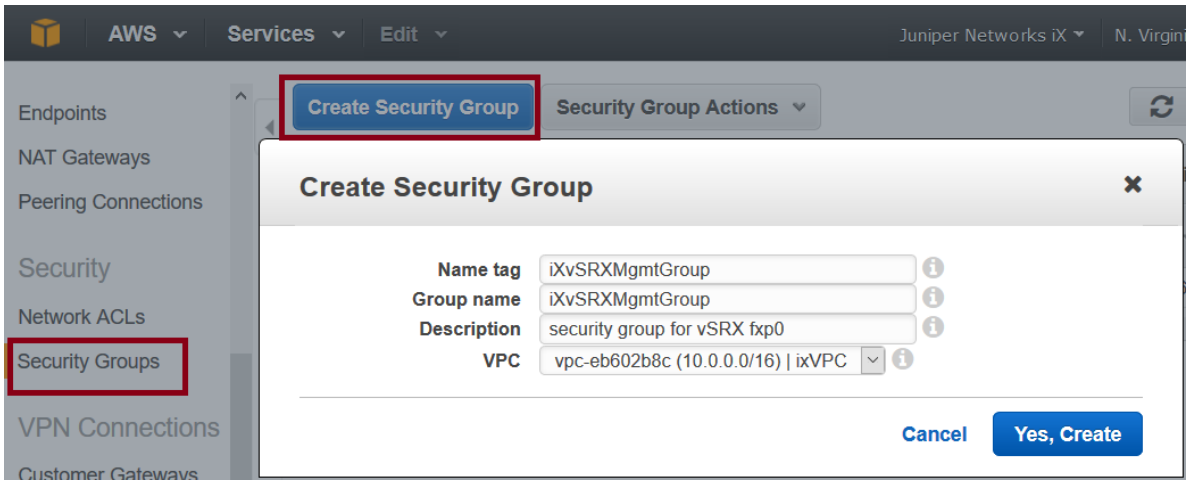
Step 5: Add Security Groups for vSRX

A default security group is created for each Amazon VPC. We recommend that you create a separate security group for the vSRX management interface (fxp0) and another security group for all other vSRX interfaces. The security groups are assigned when a vSRX instance is launched in the Amazon EC2 Dashboard, where you can also add and manage security groups.

To create the security groups:

1. In the VPC Dashboard, select **Security Groups** in the left pane, and click **Create Security Group**.

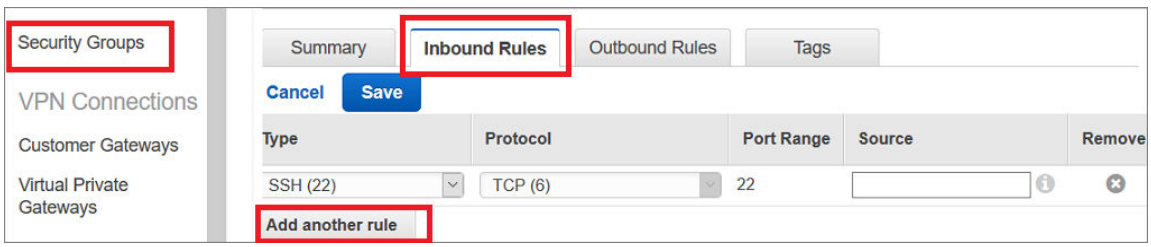
2. For the vSRX management interface, specify a security group name in the Name Tag field, edit the Group Name field (optional), enter a description of the group, and select the VPC.
3. Click **Yes, Create**.



4. Repeat Steps 1 through 3 to create a security group for the vSRX revenue interfaces.
5. Select the security group you created for the management interface and do the following:
   a. Select the **Inbound Rules** tab below the list of security groups.
   b. Click **Edit** and click **Add another rule** to create the following inbound rules:

Type	Protocol	Port	Source
Custom TCP rule	Default	20-21	Enter CIDR address format for each rule (0.0.0.0/0 allows any source).
SSH (22)	Default	Default	
HTTP (80)	Default	Default	
HTTPS (443)	Default	Default	

   c. Click **Save**.



d. Select the **Outbound Rules** tab to view the default rule that allows all outbound traffic. Use the default rule unless you need to restrict the outbound traffic.

6. Select the security group you created for all other vSRX interfaces and do the following:

> **NOTE**: The inbound and outbound rules should allow all traffic to avoid conflicts with the security settings on vSRX.

a. Select the **Inbound Rules** tab below the list of security groups.

b. Click **Edit** and create the following inbound rule:

Type	Protocol	Port	Source
All Traffic	All	All	• For webservers, enter 0.0.0.0/0 • For VPNs, enter a range of IPv4 addresses in the form of a Classless Inter-Domain Routing (CIDR) block (for example, 10.0.0.0/16).

c. Click **Save**.

d. Keep the default rule in the **Outbound Rules** tab. The default rule allows all outbound traffic.

RELATED DOCUMENTATION

- Day One: Amazon Web Services with vSRX Cookbook
- IAM Roles for Amazon EC2

Launch a vSRX Instance on an Amazon Virtual Private Cloud

IN THIS SECTION

The following procedures describe how to launch and configure a vSRX instance in the Amazon Virtual Private Cloud (Amazon VPC):

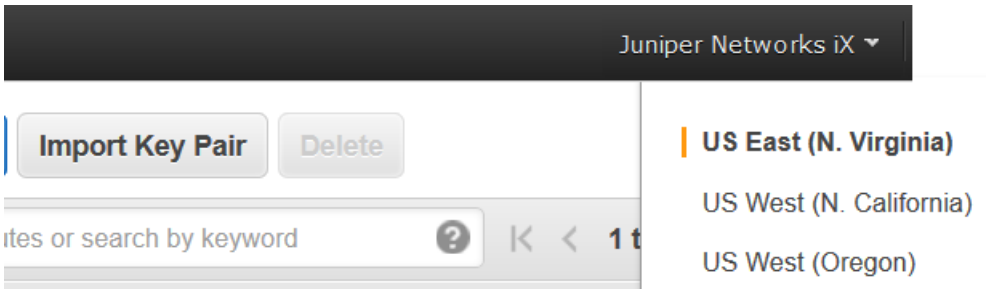
Step 1: Create an SSH Key Pair

An SSH key pair is required to remotely access a vSRX instance on AWS. You can create a new key pair in the Amazon EC2 Dashboard or import a key pair created by another tool.

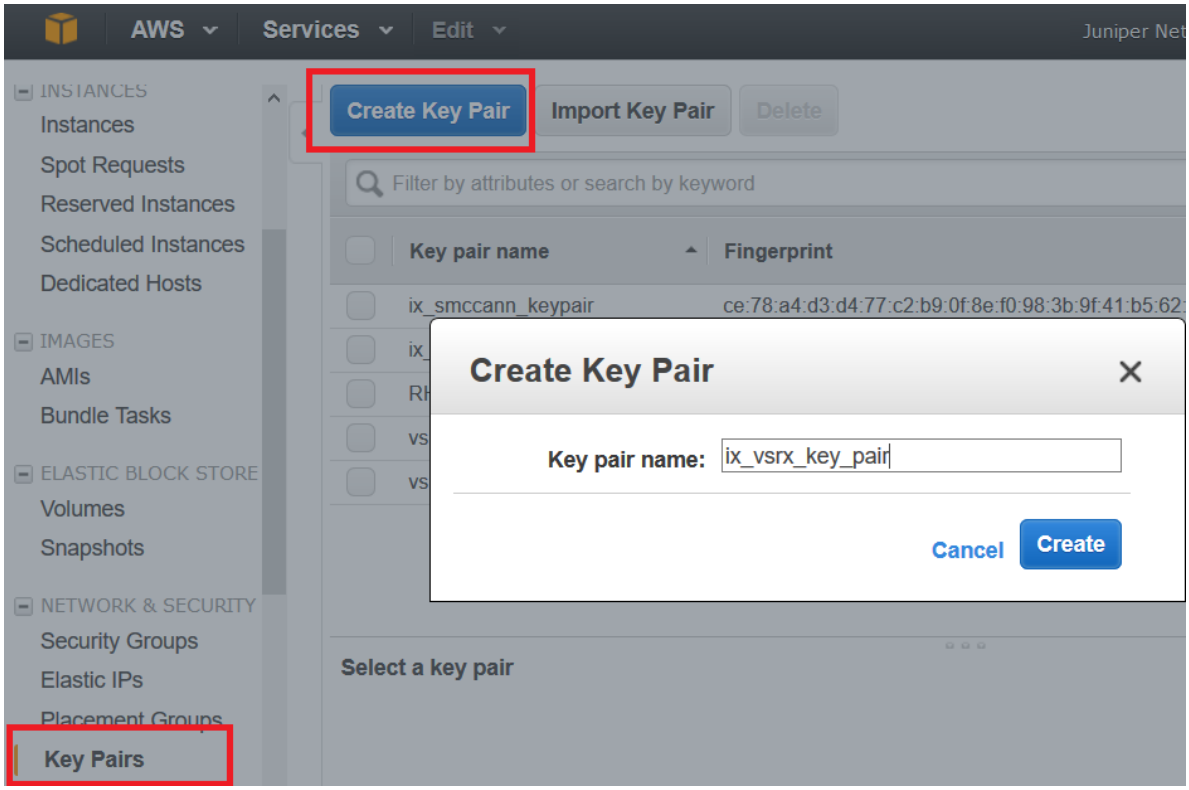
To create an SSH key pair:

1. Log in to the AWS Management Console and select **Services > Compute > EC2**.
2. In the Amazon EC2 Dashboard, select **Key Pairs** in the left pane. Verify that the region name shown in the toolbar is the same as the region where you created the Amazon Virtual Private Cloud (Amazon VPC).

Figure 84: Verify Region



3. Click **Create Key Pair**, specify a key pair name, and click **Create**.



4. The private key file (.pem) is automatically downloaded to your computer. Move the downloaded private key file to a secure location.
5. To use an SSH client on a Mac or Linux computer to connect to the vSRX instance, use the following command to set the permissions of the private key file so that only you can read it:

```
host# chmod 400 <key-pair-name>.pem
```

6. To access the vSRX instance from a shell prompt, use the `ssh -i <full path to your keyfile.pem>/<ssh-key-pair-name>.pem ec2-user@<public-ip-of-vsrx>` command. If the key file is in your current directory, then you can use the file name instead of the full path as `ssh -i <keyfile.pem>/<ssh-key-pair-name>.pem ec2-user@<public-ip-of-vsrx>`.

NOTE: Alternately, use **Import Key Pair** to import a different key pair you generated with a third-party tool.

For more information on key rotation, see https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html.

Step 2: Launch a vSRX Instance

The AWS instance types supported for vSRX are listed in Table 65 on page 335.

vSRX does not support M and C3 instances types. If you have spun your vSRX using any of these instances types, then you must change the instance type to either C4 or C5 instances type.

Table 65: Supported AWS Instance Types for vSRX

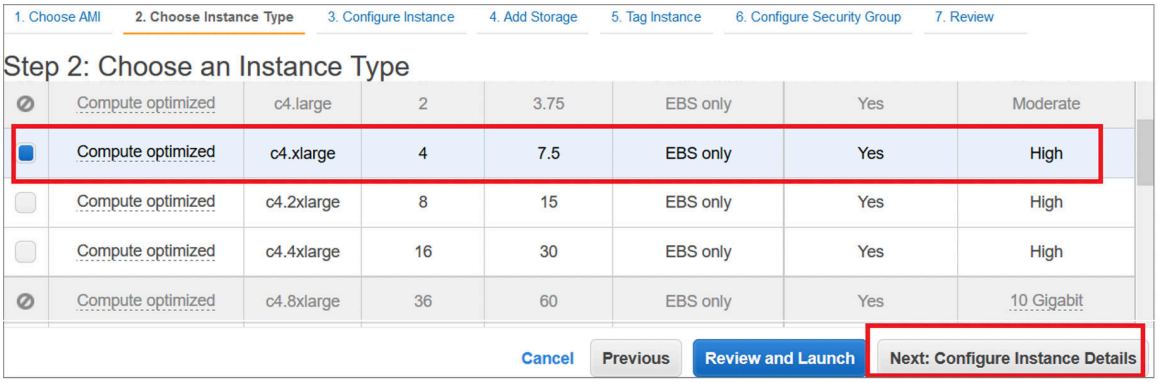
Instance Type	vSRX Type	vCPUs	Memory (GB)	RSS Type
c4.xlarge	VSRX-4CPU-7G memory	4	7.5	SW RSS
c4.2xlarge	VSRX-8CPU-15G memory	8	15	SW RSS
c4.4xlarge	VSRX-16CPU-30G memory	16	30	SW RSS
c4.8xlarge	VSRX-36CPU-60G memory	36	60	SW RSS
c5.large	VSRX-2CPU-3G memory	2	4	HW RSS
c5.2xlarge	VSRX-8CPU-15G memory	8	16	HW RSS
c5.4xlarge	VSRX-16CPU-31G memory	16	32	SW RSS
c5n.2xlarge	VSRX-8CPU-20G memory	8	21	HW RSS
c5n.4xlarge	VSRX-16CPU-41G memory	16	42	HW RSS
c5n.9xlarge	VSRX-36CPU-93G memory	36	96	HW RSS

BEST PRACTICE: **Instance Type Selection**—Based on the changes that your require for your network, you might find that your instance is overutilized, (such as the instance type is too small) or underutilized, (such as the instance type is too large). If this is the case, you can change the size of your instance. For example, if your instance is too small for its workload, you can change it to another instance type that is appropriate for the workload. You might also want to migrate from a previous generation instance type to a current generation instance type to take advantage of some features; for example, support for IPv6. Consider change of instances for better performance and throughputs.

Starting with Junos OS Release 18.4R1, c5.large vSRX instances are supported. These are cost effective and provide better performance and throughput.

To launch a vSRX instance in the Amazon VPC:

1. In the Amazon EC2 Dashboard, select **Instances** in the left pane.
2. Click **Launch Instance**, search for the vSRX on AWS Marketplace, and click **Select** next to the vSRX AMI.
3. Select a supported instance type. See Table 65 on page 335 for details.



4. Click **Next: Configure Instance Details**, and specify the fields in Table 66 on page 336.Expand **Advanced Details** to see all settings.

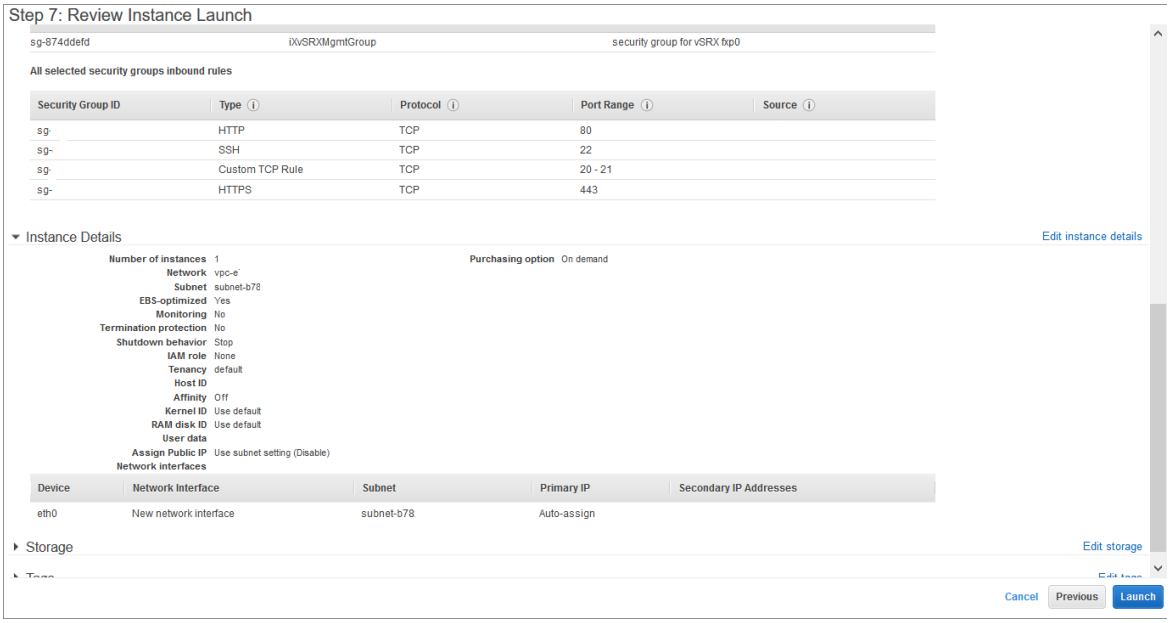
Table 66: AWS Instance Details

Field	Setting
Network	Select the Amazon VPC configured for vSRX.
Subnet	Select the public subnet for the vSRX management interface (fxp0).
Auto-assign Public IP	Select **Disable** (you will assign an Elastic IP address later).
Placement group	Use the default.
Shutdown behavior	Select **Stop** (the default).

Table 66: AWS Instance Details *(Continued)*

Field	Setting
• Enable terminal protection • Monitoring	Use your IT policy.
Network Interfaces	Use the default or assign a public IP address for the **Primary IP** field.
User data	Starting in Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX for AWS image to help simplify configuring new vSRX instances operating on AWS according to a specified user-data file. In the User data section on the Configure Instance Details page, select **As File** and attach the user-data file. The selected file is used for the initial launch of the instance. During the initial boot-up sequence, the vSRX instance processes the cloud-init request. See *Using Cloud-Init to Automate the Initialization of vSRX Instances in AWS* for information about how to create the user-data file. **NOTE**: The Junos OS configuration that is passed as user data is only imported at initial launch. If the instance is stopped and restarted, the user-data file is not imported again.

5. Click **Next: Add Storage**, and use the default settings or change the Volume Type and IOPS as needed.
6. Click **Next: Tag Instance**, and specify a name for the vSRX instance.
7. Click **Next: Configure Security Group**, select **Select an existing security group**, and select the security group created for the vSRX management interface (fxp0).
8. Click **Review and Launch**, review the settings for the vSRX instance, and click **Launch**.



9. Select the SSH key pair you created, select the acknowledgment check box, and click **Launch Instance**.
10. Click **View Instances** to display the Instances list in the Amazon EC2 Dashboard. It might take several minutes to launch a vSRX instance.

Step 3: View the AWS System Logs

To debug launch time errors, you can view the AWS system logs, as follows:

1. In the Amazon EC2 Dashboard, select **Instances**.
2. Select the vSRX instance, and select **Actions > Instance Settings > Get System Logs**.

Step 4: AddNetwork Interfaces for vSRX

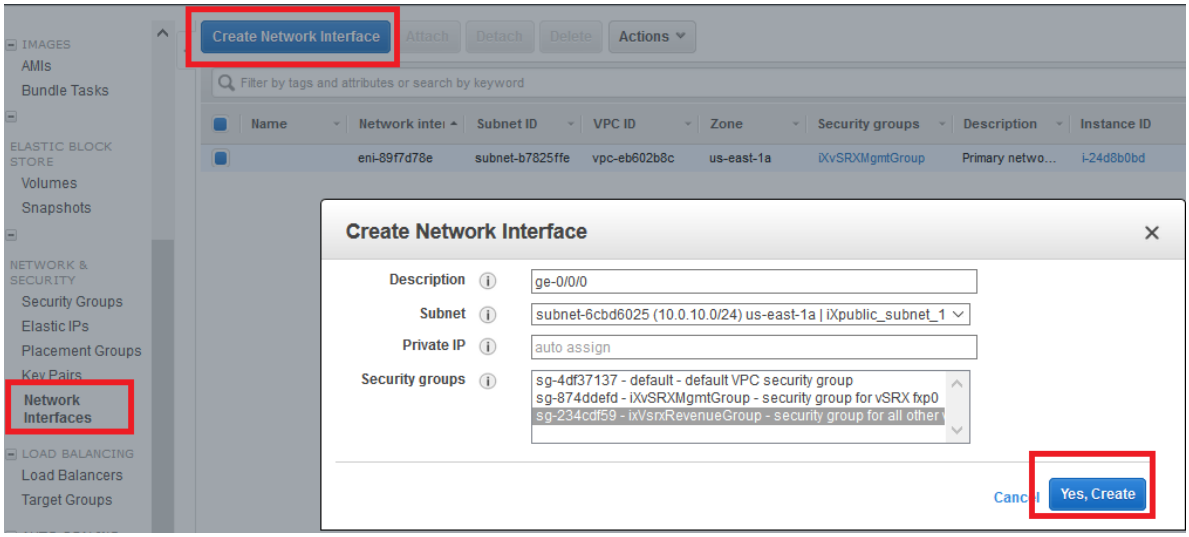
AWS supports up to eight interfaces for an instance, depending on the AWS instance type selected. Use the following procedure for each of the revenue interfaces you want to add to vSRX (up to seven). The first revenue interface is ge-0/0/0, the second is ge-0/0/1, and so on (see *Requirements for vSRX on AWS*).

To add a vSRX revenue interface:

1. In the Amazon EC2 Dashboard, select **Network Interfaces** in the left pane, and click **Create Network Interface**.
2. Specify the interface settings as shown in Table 67 on page 339, and click **Yes, Create**.

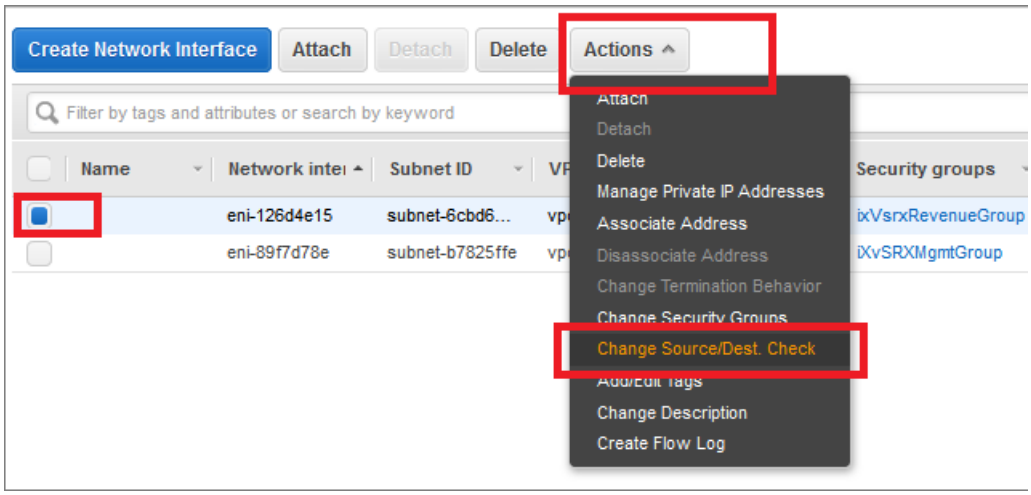
Table 67: Network Interface Settings

Field	Setting
Description	Enter an interface description for each of the revenue interfaces.
Subnet	Select the public subnet created for the first revenue interface (ge-0/0/0) or the private subnet created for all the other revenue interfaces.
Private IP	Enter an IP address from the selected subnet or allow the address to be assigned automatically.
Security Groups	Select the security group created for the vSRX revenue interfaces.



3. Select the new interface, select **Actions > Change Source/Dest. Check**, select **Disabled**, and click **Save**.

Figure 85: Disable Source/Dest. Check



4. Select the new interface, select **Attach**, select the vSRX instance, and click **Attach**.
5. Click the pencil icon in the new interface Name column and give the interface a name (for example, ix-fxp0.0).

NOTE: For a private revenue interface (ge-0/0/1 through ge-0/0/7), make a note of the network name you created or the network interface ID. You will add the name or interface ID later to the route table created for the private subnet.

Step 5: Allocate Elastic IP Addresses

For public interfaces, AWS does a NAT translation of the public IP address to a private IP address. The public IP address is called an *Elastic IP address*. We recommend that you assign an Elastic IP address to the public vSRX interfaces (fxp0 and ge-0/0/0). Note that when a vSRX instance is restarted, the Elastic IPs are retained, but public subnet IPs are released.

To create and allocate Elastic IPs:

1. In the Amazon EC2 Dashboard, select **Elastic IPs** in the left pane, click **Allocate New Address**, and click **Yes, Allocate**. (If your account supports EC2-Classic, you must first select **EC2-VPC** from the Network platform list.)
2. Select the new Elastic IP address, and select **Actions > Associate Address**.
3. Specify the settings in Table 68 on page 341, and click **Allocate**.

Table 68: Elastic IP Settings

Field	Setting
Network Interface	Select the vSRX management interface (fxp0) or the first revenue interface (ge-0/0/0).
Private IP Address	Enter the private IP address to be associated with the Elastic IP address.

Step 6: Add the vSRX Private Interfaces to the Route Tables

For each private revenue interface you created for vSRX, you must add the interface ID to the route table you created for the associated private subnet.

To add a private interface ID to a route table:

1. In the VPC Dashboard, select **Route Tables** in the left pane.
2. Select the route table you created for the private subnet.
3. Select the **Routes** tab below the list of route tables.
4. Click **Edit** and click **Add another route**.
5. Specify the settings in Table 69 on page 341, and click **Save**.

   Table 69: Private Route Settings

Field	Setting
Destination	Enter 0.0.0.0/0 for Internet traffic.
Target	Type the network name or the network interface ID for the associated private subnet. The network interface must be in the private subnet shown in the **Subnet Associations** tab. **NOTE**: Do not select the Internet gateway (igw-nnnnnnnn).

Repeat this procedure for each private network interface. You must reboot the vSRX instance to complete this configuration.

Step 7: Reboot the vSRX Instance

To incorporate the interface changes and complete the Amazon EC2 configuration, you must reboot the vSRX instance. Interfaces attached while the vSRX instance is running do not take effect until the instance is rebooted.

NOTE: Always use AWS to reboot the vSRX instance. Do not use the vSRX CLI to reboot.

To reboot a vSRX instance:

1. In the Amazon EC2 Dashboard, select **Instances** in the left pane.
2. Select the vSRX instance, and select **Actions > Instance State > Reboot**.

It might take several minutes to reboot a vSRX instance.

Step 8: Log in to a vSRX Instance

In AWS deployments, vSRX instances provide the following capabilities by default to enhance security:

- Allows you to login only through SSH.
- cloud-init is used to setup SSH key login.
- SSH password login is disabled for root account.

vSRX instances launched on Amazon's AWS cloud infrastructure uses the cloud-init services provided by Amazon to copy the SSH public-key associated with your account that is used to launch the instance. You will then be able to login to the instance using the corresponding private-key.

NOTE: Root login using SSH password is be disabled by default.

Use an SSH client to log in to a vSRX instance for the first time. To log in, specify the location where you saved the SSH key pair **.pem** file for the user account, and the Elastic IP address assigned to the vSRX management interface (fxp0).

NOTE: Starting in Junos OS Release 17.4R1, the default user name has changed from `root@` to `ec2-user@`.

```
ssh -i <path>/<ssh-key-pair-name>.pem ec2-user@<fxpo-elastic-IP-address>
```

NOTE: Root login using a Junos OS password is disabled by default. You can configure other users after the initial Junos OS setup phase.

If you do not have the key pair filename and Elastic IP address, use these steps to view the key pair name and Elastic IP for a vSRX instance:

1. In the Amazon EC2 Dashboard, select **Instances**.
2. Select the vSRX instance, and select **eth0** in the Description tab to view the Elastic IP address for the fxp0 management interface.
3. Click **Connect** above the list of instances to view the SSH key pair filename.

To configure the basic settings for the vSRX instance, see *Configure vSRX Using the CLI*.

NOTE: vSRX pay-as-you-go images do not require any separate licenses.

Release History Table

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX for AWS image to help simplify configuring new vSRX instances operating on AWS according to a specified user-data file.

RELATED DOCUMENTATION

Day One: Amazon Web Services with vSRX Cookbook

Using Cloud-Init to Automate the Initialization of vSRX Instances in AWS

Starting in Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX for AWS image to help simplify configuring new vSRX instances operating on AWS according to a specified user-data file. Cloud-init is performed during the first-time boot of a vSRX instance.

Cloud-init is an open source application for automating the initialization of a cloud instance at boot-up. Cloud-init is designed to support multiple different cloud environments, such as Amazon EC2, so that the same virtual machine (VM) image can be directly used in multiple cloud instances without any modification. Cloud-init support in a VM instance runs at boot time (first-time boot) and initializes the VM instance according to the specified user-data file.

A user-data file is a special key in the metadata service that contains a file that cloud-aware applications in the VM instance can access upon a first-time boot. In this case, it is the validated Junos OS configuration file that you intend to upload to a vSRX instance as the active configuration. This file uses

the standard Junos OS command syntax to define configuration details, such as root password, management IP address, default gateway, and other configuration statements.

When you create a vSRX instance, you can use `cloud-init` services on AWS to pass a valid Junos OS configuration file as user data to initialize new vSRX instances. The user-data file uses the standard Junos OS syntax to define all the configuration details for your vSRX instance. The default Junos OS configuration is replaced during the vSRX instance launch with a validated Junos OS configuration that you supply in the form of a user-data file.

NOTE: The user-data file cannot exceed 16 KB. If your user-data file exceeds this limit, you must compress the file using gzip and use the compressed file. For example, the gzip junos.conf command results in the junos.conf.gz file.

The configuration must be validated and include details for the fxp0 interface, login, and authentication. It must also have a default route for traffic on fxp0. This information must match the details of the AWS VPC and subnet into which the instance is launched. If any of this information is missing or incorrect, the instance is inaccessible and you must launch a new one.

WARNING: Ensure that the user-data configuration file is not configured to perform autoinstallation on interfaces using Dynamic Host Configuration Protocol (DHCP) to assign an IP address to the vSRX. Autoinstallation with DHCP will result in a "commit fail" for the user-data configuration file.

To initiate the automatic setup of a vSRX instance from AWS:

1. If you have not done so already, create a configuration file with the Junos OS command syntax and save the file. The configuration file can be plain text or MIME file type text/plain.

   The user-data configuration file must contain the full vSRX configuration that is to be used as the active configuration on each vSRX instance, and the string `#junos-config` must be the first line of the user-data configuration file before the Junos OS configuration.

   NOTE: The `#junos-config` string is mandatory in the user-data configuration file; if it is not included, the configuration will not be applied to the vSRX instance as the active configuration.

2. Copy the Junos OS configuration file to an accessible location from where it can be retrieved to launch the vSRX instance.
3. To specify the user-data file for configuring the vSRX instance, select **As File** in the User data section on the Configure Instance Details page and attach the file (as described in *Launch a vSRX Instance on an Amazon Virtual Private Cloud*). The selected configuration file is used for the initial launch of the

vSRX instance. During the initial boot-up sequence, the vSRX instance processes the cloud-init request.

> **NOTE**: The boot time for the vSRX instance might increase with the use of the cloud-init package. This additional time in the initial boot sequence is due to the operations performed by the cloud-init package. During this operation, the cloud-init package halts the boot sequence and performs a lookup for the configuration data in each data source identified in the cloud.cfg. The time required to look up and populate the cloud data is directly proportional to the number of data sources defined. In the absence of a data source, the lookup process continues until it reaches a predefined timeout of 30 seconds for each data source.

4. When the initial boot-up sequence resumes, the user-data file replaces the original factory-default Junos OS configuration loaded on the vSRX instance. If the commit succeeds, the factory-default configuration will be permanently replaced. If the configuration is not supported or cannot be applied to the vSRX instance, the vSRX will boot using the default Junos OS configuration.

Release History Table

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX for AWS image to help simplify configuring new vSRX instances operating on AWS according to a specified user-data file. Cloud-init is performed during the first-time boot of a vSRX instance.

RELATED DOCUMENTATION

- Cloud-Init Documentation
- cloud-init
- Launching an Instance

AWS Elastic Load Balancing and Elastic Network Adapter

IN THIS SECTION

This section provides an overview of the AWS ELB and ENA features and also describes how these features are deployed on vSRX instances.

Overview of AWS Elastic Load Balancing

IN THIS SECTION

This section provides information about AWS ELB.

Elastic Load Balancing (ELB) is a load-balancing service for Amazon Web Services (AWS) deployments.

ELB distributes incoming application or network traffic across ntra availability zones, such as Amazon EC2 instances, containers, and IP addresses. ELB scales your load balancer as traffic to your application changes over time, and can scale to the vast majority of workloads automatically.

AWS ELB using application load balancers enables automation by using certain AWS services:

- **Amazon Simple Notification Service**—For more information, see https://docs.aws.amazon.com/sns/latest/dg/welcome.html.
- **AWS Lambda**—For more information, see https://docs.aws.amazon.com/lambda/latest/dg/welcome.html.
- **AWS Auto Scale Group**—For more information, see https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html.

Benefits of AWS Elastic Load Balancing

- Ensures elastic load balancing for intra available zone by automatically distributing the incoming traffic.
- Provides flexibility to virtualize your application targets by allowing you to host more applications on the same instance and to centrally manage Transport Layer Security (TLS) settings and offload CPU-intensive workloads from your applications.
- Provides robust security features such as integrated certificate management, user authentication, and SSL/TLS decryption.
- Supports auto-scaling a sufficient number of applications to meet varying levels of application load without requiring manual intervention.
- Enables you to monitor your applications and their performance in real time with Amazon CloudWatch metrics, logging, and request tracing.
- Offers load balancing across AWS and on-premises resources using the same load balancer.

AWS Elastic Load Balancing Components

AWS Elastic Load Balancing (ELB) components include:

- **Load balancers**—A load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple availability zones (AZs), thereby increasing the availability of your application. You add one or more listeners to your load balancer.
- **Listeners or vSRX instances**—A listener is a process for checking connection requests, using the protocol and port that you configure. vSRX instances as listeners check for connection requests from clients, using the protocol and port that you configure, and forward requests to one or more target groups, based on the rules that you define. Each rule specifies a target group, condition, and priority. When the condition is met, the traffic is forwarded to the target group. You must define a default rule for each vSRX instance, and you can add rules that specify different target groups based on the content of the request (also known as content-based routing).
- **Target groups or vSRX application workloads**—Each vSRX application as target group is used to route requests to one or more registered targets. When you create each vSRX instance as a listener rule, you specify a vSRX application and conditions. When a rule condition is met, traffic is forwarded to the corresponding vSRX application. You can create different vSRX applications for different types of requests. For example, create one vSRX application for general requests and other vSRX applications for requests to the microservices for your application.

AWS ELB supports three types of load balancers: application load balancers, network load balancers, and classic load balancers. You can select a load balancer based on your application needs. For more information about the types of AWS ELB load balancers, see AWS Elastic Load Balancing.

Overview of Application Load Balancer

Starting in Junos OS Release 18.4R1, vSRX instances support AWS Elastic Load Balancing (ELB) using the application load balancer to provide scalable security to the Internet-facing traffic using native AWS services. An application load balancer automatically distributes incoming application traffic and scales resources to meet traffic demands.

You can also configure health checks to monitor the health of the registered targets so that the load balancer can send requests only to the healthy targets.

The key features of an application load balancer are:

- Layer-7 load balancing
- HTTPS support
- High availability
- Security features
- Containerized application support
- HTTP/2 support
- WebSockets support
- Native IPv6 support
- Sticky sessions
- Health checks with operational monitoring, logging, request tracing
- Web Application Firewall (WAF)

When the application load balancer receives a request, it evaluates the rules of the vSRX instance in order of priority to determine which rule to apply, and then selects a target from the vSRX application for the rule action. You can configure a vSRX instance rule to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.

You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application. ELB scales your load balancer as traffic to your application changes over time. ELB can scale majority of workloads automatically.

The application load balancer launch sequence and current screen can be viewed using the vSRX instance properties. When running vSRX as an AWS instance, logging in to the instance through SSH starts a session on Junos OS. Standard Junos OS CLI can be used to monitor health and statistics of the vSRX instance. If the #load_balancer=true tag is sent in user data, then boot-up messages mention that the vSRX interfaces are configured for ELB and auto-scaling support. Interfaces eth0 and eth1 are then swapped.

If an unsupported Junos OS configuration is sent to the vSRX instance in user data, then the vSRX instance reverts to its factory-default configuration. If the #load_balancer=true tag is missing, then interfaces are not swapped.

Deployment of AWS Application Load Balancer

IN THIS SECTION

AWS ELB application load balancer can be deployed in two ways:

- vSRX behind AWS ELB application load balancer
- ELB sandwich

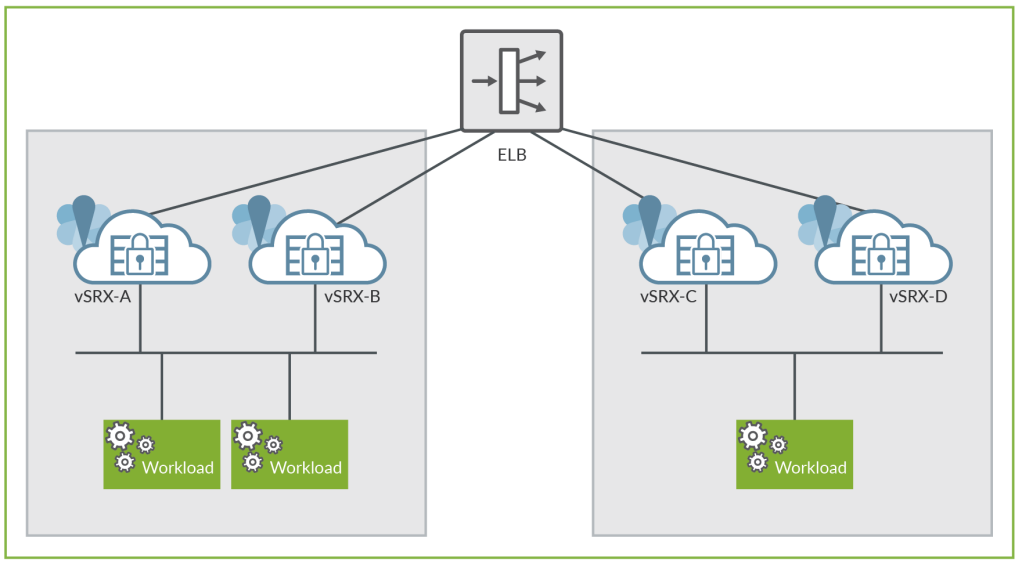
vSRX Behind AWS ELB Application Load Balancer Deployment

In this type of deployment, the vSRX instances are attached to the application load balancer, in one or more availability zones (AZs), and the application workloads are behind the vSRX instances. The application load balancer sends traffic only to the primary interface of the instance. For a vSRX instance, the primary interface is the management interface fxp0.

To enable ELB in this deployment, you have to swap the management and the first revenue interface.

Figure 86 on page 350 illustrates the vSRX behind AWS ELB application load balancer deployment.

Figure 86: vSRX Behind AWS ELB Application Load Balancer Deployment



Enabling AWS ELB with vSRX Behind AWS ELB Application Load Balancer Deployment

The following are the prerequisites for enabling AWS ELB with the vSRX behind AWS ELB application load balancer type of deployment:

- All incoming and outgoing traffic to ELB are monitored from the ge-0/0/0 interface associated with the vSRX instance.
- The vSRX instance at launch has two interfaces in which the subnets containing the interfaces are connected to the internet gateway (IGW). The two interface limit is set by the AWS auto scaling group deployment. You need to define at least one interface in the same subnet as the AWS ELB. The additional interfaces can be attached by the lambda function.
- Source or destination check is disabled on the eth1 interface of the vSRX instance.

For deploying an AWS ELB application load balancer using the vSRX behind AWS ELB application load balancer method:

The vSRX instance contains:

- Cloud initialization (cloud-init) user data with ELB tag as #load_balancer=true.

- The user data configuration with #junos-config tag, fxp0 (dhcp), ge-0/0/0 (dhcp) (must be DHCP any security group that it needs to define)
- Cloud-Watch triggers an Simple Notification Service (SNS), which in turn triggers a Lambda function that creates and attaches an Elastic Network Interface (ENI) with Elastic IP address (EIP) to the vSRX instance. Multiple new ENIs (maximum of 8) can be attached to this instance.
- The vSRX Instance must be rebooted. A reboot must be performed for all subsequent times the vSRX instance launches with swapped interfaces.

NOTE: Chassis cluster is not supported if you try to swap the ENI between instances and IP monitoring.

NOTE: You can also launch the vSRX instance in an Auto Scaling Group (ASG). This launch can be automated using a cloud formation template (CFT).

Sandwich Deployment of AWS ELB Application Load Balancer

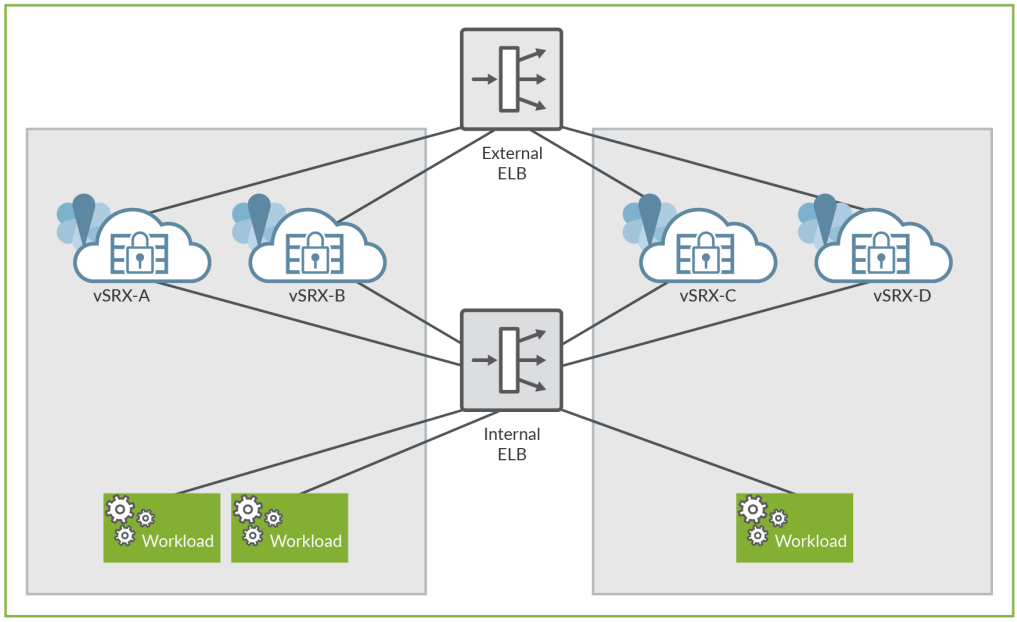
In this deployment model, you can scale both, security and applications. vSRX instances and the applications are in different ASGs and each of these ASGs is attached to a different application load balancer. This type of ELB deployment is elegant and simplified way to manually scale vSRX deployments to address planned or projected traffic increases while also delivering multi-AZ high availability. The deployment ensures inbound high availability and scaling for AWS deployments.

Because the load balancer scales dynamically, its virtual IP address (VIP) is a fully qualified domain name (FQDN). This FQDN resolves to multiple IP addresses according to the availability zone. To enable this resolution, the vSRX instance should be able to send and receive traffic from the FQDN (or the multiple addresses that it resolves to).

You configure this FQDN by using the `set security zones security-zone ELB-TRAFFIC address-book address ELB dns-name FQDN_OF_ELB` command.

Figure 87 on page 352 illustrates the AWS ELB application load balancer sandwich deployment for vSRX.

Figure 87: Sandwich Deployment of AWS ELB Application Load Balancer



Enabling Sandwich Deployment of AWS Application Load Balancer for vSRX

For AWS ELB application load balancer sandwich deployment for vSRX:

- vSRX receives the #load_balancer=true tag in cloud-init user data.
- In Junos OS, the initial boot process scans the mounted disk for the presence of the flag file in the **setup_vsrx** file. If the file is present, it indicates that the two interfaces with DHCP in two different virtual references must be configured. This scan and configuration update is performed in the default configuration and on top of the user data if the flag file is present.

  NOTE: If user data is present, then the boot time after the second or the third mgd process commit increases.

- You must reboot the vSRX instance. Perform reboot for all the subsequent times the vSRX instance is launched with swapped interfaces.

NOTE: Chassis cluster support for swapping the Elastic Network Interfaces (ENIs) between instances and IP monitoring does not work.

NOTE: You can also launch vSRX instance in an ASG and automate the deployment using a cloud formation template (CFT).

Invoking Cloud Formation Template (CFT) Stack Creation for vSRX Behind AWS Application Load Balancer Deployment

This topic provide details on how to invoke cloud formation template (CFT) stack creation for the non-sandwich deployment (with vSRX Behind AWS Application Load Balancer) which contains only one load balancer.

Before you invoke the CFT stack creation, ensure you have the following already available within AWS environment:

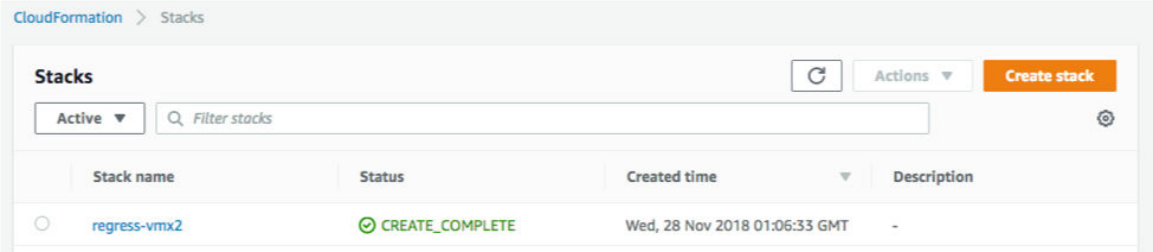
- VPC created and ready to use.
- A management subnet
- An external subnet (subnet for vSRX interface receiving traffic from the ELB).
- An internal subnet (subnet for vSRX interface sending traffic to the workload).
- An AMI ID of the vSRX instance that you want to launch.
- User data (the vSRX configuration that has to be committed before the traffic is forwarded to the workload. This is a base 64 encoded data not more than 4096 characters in length; you may use up to three user data fields if a single field data exceeds 4096 characters).
- EC2 key file.
- Get the lambda function file add_eni.zip from Juniper vSRX GitHub repository and upload it to your instances S3 bucket. Use this information in the **Lambda S3 Location** field of the template.
- Your AWS account should have permissions to create Lambda functions on various resources in your region.

Follow the following steps to invoke CFT stack creation for AWS ELB with vSRX behind AWS ELB application load balancer deployment.

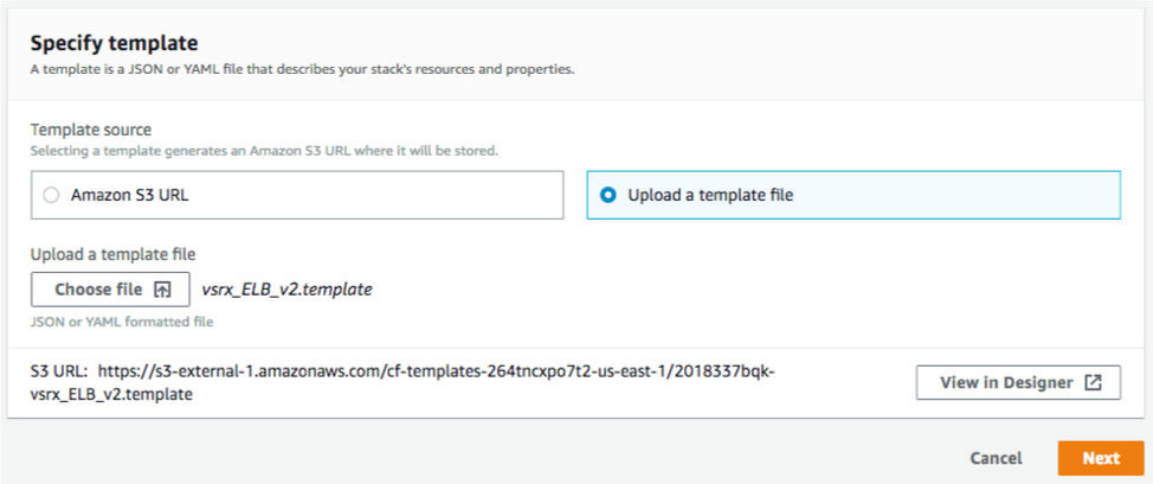
1. Log into your AWS account and make sure the region on the top right is the one you want to use.

Go to AWS console home page and under **All Services** look for **Management & Governance** section and click **CloudFormation** option.

2. Click the **Create Stack** button on the top right side of the CloudFormation page.



3. On the new page, select **Upload a template file radio** button, then click **Choose file** button, and then select your template file and click **Next**.



4. The next page that opens is a form created from the template. Some fields might already have a default value, that you might change if you want to.

   Enter a **Stack Name**, select the **VPC ID**, **InstanceType**, **MgtSubnetID**, **ExternalSubnetID**, **InternalSubnetID**, **ImageID**. Paste the Base64 encoded user data (which is the vSRX configuration to be committed and is provided in a separate text file). If your Base64 encoded vSRX configuration exceeds 4096 bytes, you may use UserData2 and UserData3 fields as needed.

5. Set **MinASGInstances** as 1 and **MaxASGInstances** as 3

6. Select your Amazon EC2 Key Pair file and click **Next**.

Create stack

Stack name

Stack name

vSRX3-CFT-Test1

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

vSRX configuration

VPCID
VPC ID

vpc-fba8a69f (10.0.0.0/16) (DC-01)

InstanceType
Select the instance type you want to use

c5.large

MgtSubnetID
Subnet ID used for the mgmt interface

subnet-ee5fe5c4 (10.0.254.0/24) (DC01a_MgmtSub_fxp0)

ExternalSubnetID
Subnet ID used for the revenue interface

subnet-335ce619 (10.0.0.0/24) (DC01a_PubSub_ge0)

InternalSubnetID
Subnet ID used for the internal interface

subnet-115ce63b (10.0.1.0/24) (DC01a_PvtSub_ge1)

ImageID

ami-048457f49622be514

UserData
User Data configuration. Maximum 4096 bytes.For better readability, encode it with base64. Final userdata will be the combination of UserData, UserData2 and UserData3

IwZSB2aXJ0dWFsLXJvdXRlcjsKICAgICAgICBpbnRlcmZhY2UgZ2UtMC8wLzAuMDsKICAgICAgICBpbnRlcmZhY2UgZ2UtMC8wLzEuMDsKICAgIH0KfQ==

UserData2
User Data configuration (Optional). Maximum 4096 bytes.

UserData3
User Data configuration (Optional). Maximum 4096 bytes.

Auto scaling group configuration

MinASGInstances
Minimum number of vSRX in the auto scaling group

1

MaxASGInstances
Maximum number of vSRX in the auto scaling group

3

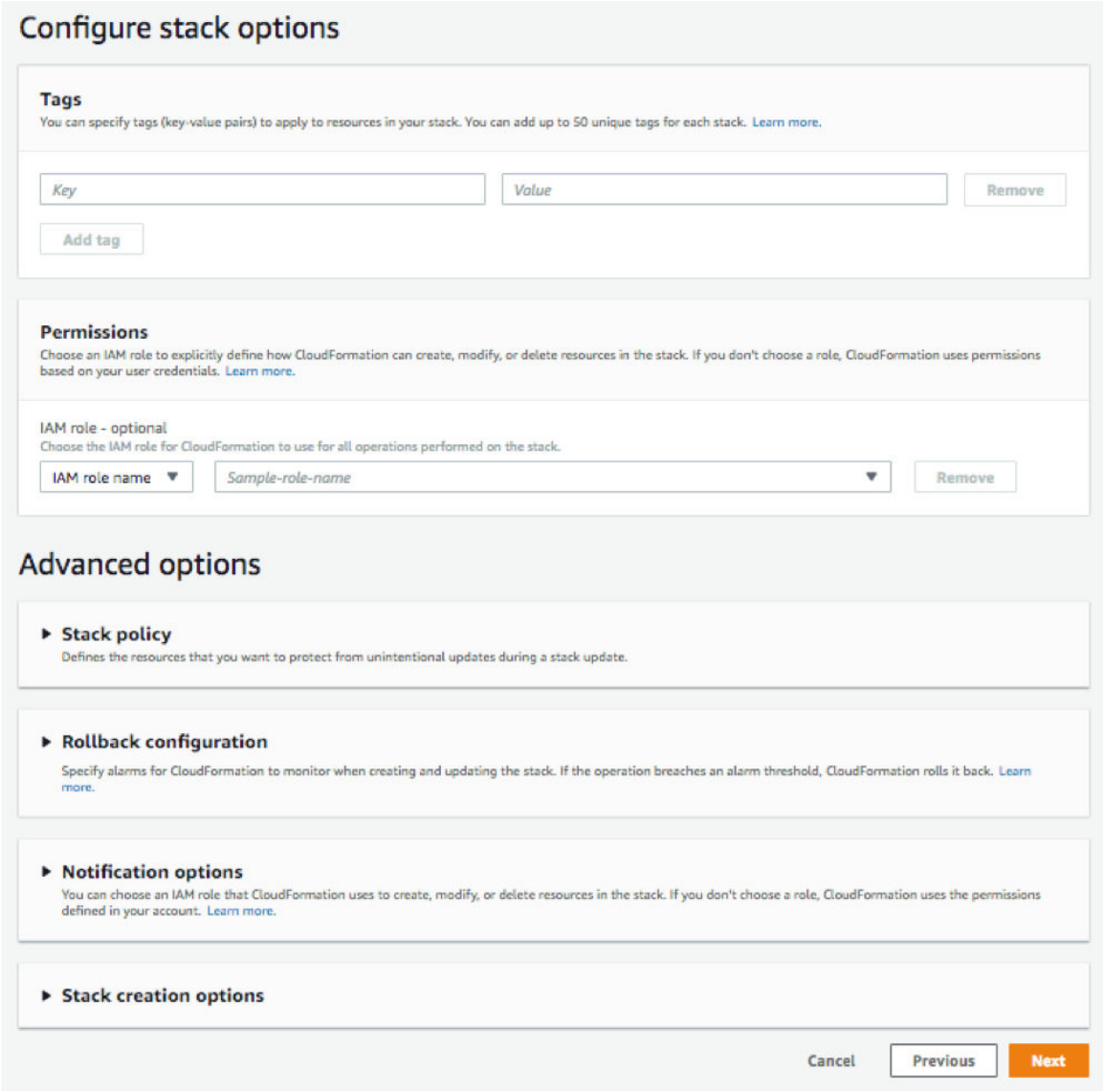
Other parameters

EC2KeyPair
Amazon EC2 Key Pair

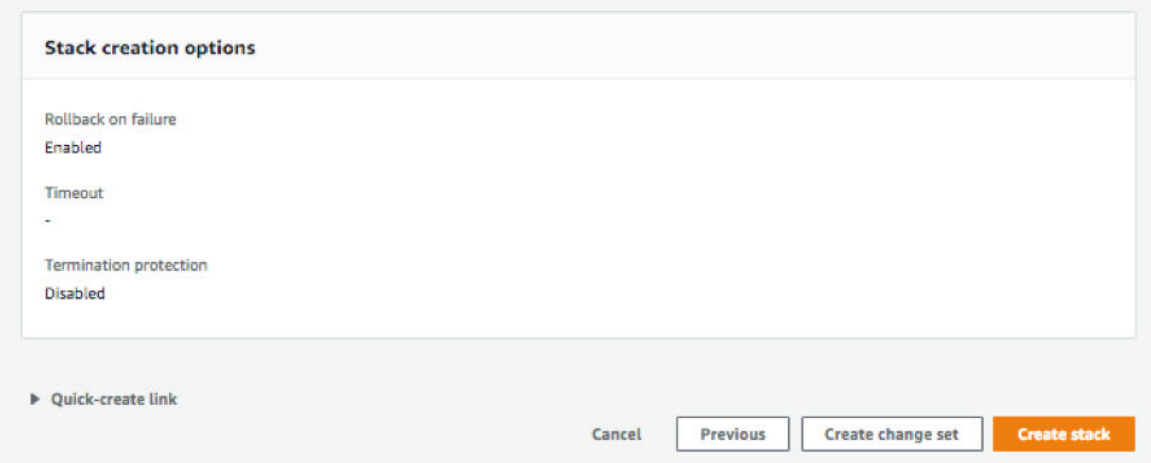
vsrx_test

Cancel | Previous | Next

7. Skip the next page with **Configure stack** options and **Advanced** option and click **Next**.



8. On the next page, you will be able to review and edit your stack creation details. Once you are done reviewing, click **Create stack** button on the bottom right of the page.



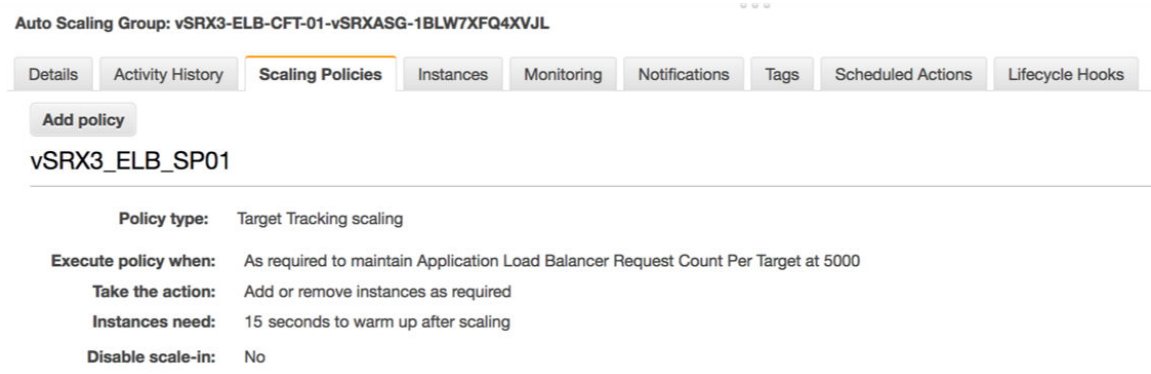
9. On the next page, wait for the stack creation to be completed. If there are any errors in the stack creation, then the errors are displayed on this page. You have to rectify the errors and recreate the stack using the above steps.
10. Once the stack is created successfully , click **Services**>**EC2** and then click **Auto Scaling Groups** on the left-hand side menu.

    On the right-hand side of the page, you should see an auto-scaling group (ASG) with the stack name that you created.

    When you select the ASG you created then that ASG details are displayed at the bottom of the page.

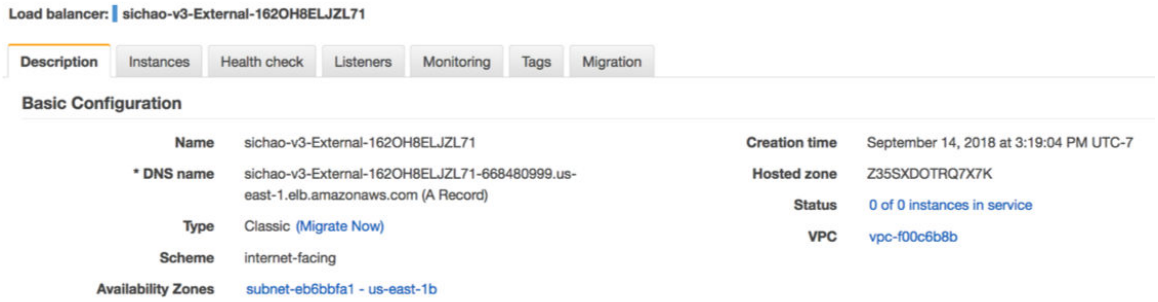
    Click the **Scaling Policies** tab to create a scaling policy for this ASG, to maintain a certain number of vSRXs in the ASG and to cater to various requests, as per your requirements. Refer to 'Scaling policy example' under the 'Sample Data' in this topic below.

    Auto Scaling Group monitors the state of the vSRX instances. It will automatically re spawn a new instance if any vSRX instance failure is detected. You can find more information in the **Activity History** tab of the ASG and in the Cloudwatch logs.



11. Click **Services**>**EC2** and then **Load Balancers** on the left-hand side menu. On the right-hand side of the page, you should see a load balancer (LB) with the stack name that you created. You can select this load balancer and view the load balancer details at the bottom of the page.

    The **instances** tab above will show the vSRX instances being load-balanced by this LB. This LB will be assigned a DNS name as show above. Any HTTP traffic sent to that host will be forwarded by the vSRX to the web server workload being protected by the vSRX. The number of vSRX instances can vary between **MinASGInstances** and **MaxASGInstances** used during setup, depending upon the scaling criteria.



12. **For Scaling a Policy**:

    - As mentioned in Step 11, click on **Add policy** on the **Scaling Policies** tab of your Auto Scaling Group (ASG) and name the policy.
    - Select a **Metric type** from the drop down list, for example: for Average CPU Utilization, enter a **Target Value** as 75. Add 30 seconds warm-up time the vSRX instances need and leave **Disable scale-in** unchecked.
    - Click **Create** to add this policy to the ASG. The ASG executes the policy as required to maintain average CPU utilization at 75.

Sample Configuration of AWS Elastic Load Balancer with vSRX instance for HTTP Traffic

- You need to have your DNS server IP and your Web Server IP (or if your web server is behind a load balancer, then use that load balancer's IP address below instead of the Web Server IP).
- After using your IP addresses in the below configuration, convert this configuration into Base 64 format (refer to: https://www.base64encode.org/) and then paste the converted configuration into the UserData field. By doing so, applies the below configuration to the existing default configuration on a vSRX launched in AWS, during the stack creation process.

```
#load_balancer=true
#junos-config
system {
    name-server {
```

```
<Your DNS Server IP>
}
    syslog {
        file messages {
            any any;
        }
    }
}
security {
    address-book {
        global {
            address websrv <Your Web Server IP>/32>;
}
    }
    nat {
        source {
            rule-set src-nat {
                from interface ge-0/0/0.0;
                to zone trust;
                rule rule1 {
                    match {
                        source-address 0.0.0.0/0;
                        destination-port {
                            80;
                        }
                    }
                    then {
                        source-nat {
                            interface;
                        }
                    }
                }
            }
        }
        destination {
            pool pool1 {
                address <Your Web Server IP>/32;
}
            rule-set dst-nat {
                from interface ge-0/0/0.0;
                rule rule1 {
                    match {
                        destination-address 0.0.0.0/0;
```

```
                    destination-port {
                        80;
                    }
                }
                then {
                    destination-nat {
                        pool {
                            pool1;
                        }
                    }
                }
            }
        }
    }
    policies {
        from-zone untrust to-zone trust {
            policy mypol {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone trust {
            host-inbound-traffic {
                system-services {
                    any-service;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                ge-0/0/1.0;
            }
        }
```

```
        security-zone untrust {
            host-inbound-traffic {
                system-services {
                    any-service;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                ge-0/0/0.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                dhcp;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                dhcp;
            }
        }
    }
}
routing-instances {
    ELB_RI {
        instance-type virtual-router;
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
    }
}
```

Overview of AWS Elastic Network Adapter (ENA) for vSRX Instances

IN THIS SECTION

Amazon Elastic Compute Cloud (EC2) provides the Elastic Network Adapter (ENA), the next-generation network interface and accompanying drivers that provide enhanced networking on EC2 vSRX instances.

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA).

Benefits

- Supports multiqueue device interfaces. ENA makes uses of multiple transmit and receive queues to reduce internal overhead and to increase scalability. The presence of multiple queues simplifies and accelerates the process of mapping incoming and outgoing packets to a particular vCPU.
- The ENA driver supports industry-standard TCP/IP offload features such as checksum offload and TCP transmit segmentation offload (TSO).
- Supports receive-side scaling (RSS) network driver technology that enables the efficient distribution of network receive processing across multiple CPUs in multiprocessor systems, for multicore scaling. Some of the ENA devices support a working mode called low-latency queue (LLQ), which saves several microseconds.

Understanding AWS Elastic Network Adapter

Enhanced networking uses single-root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types. SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. Enhanced networking provides higher bandwidth, higher packet per second (pps) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking.

ENA is a custom network interface optimized to deliver high throughput and packet per second (pps) performance, and consistently low latencies on EC2 vSRX instances. Using ENA for vSRX C5.large instances (with 2 vCPUs and 4-GB memory), you can utilize up to 20 Gbps of network bandwidth. ENA-based enhanced networking is supported on vSRX instances.

The ENA driver exposes a lightweight management interface with a minimal set of memory-mapped registers and an extendable command set through an admin queue. The driver supports a wide range of ENA adapters, is link-speed independent (that is, the same driver is used for 10 Gbps, 25 Gbps, 40 Gbps, and so on), and negotiates and supports various features. The ENA enables high-speed and low-overhead Ethernet traffic processing by providing a dedicated Tx/Rx queue pair per CPU core.

The DPDK drivers for ENA are available at https://github.com/amzn/amzn-drivers/tree/master/userspace/dpdk.

NOTE: When AWS ELB application load balancers are used, the eth0 (first) and eth1 (second) interfaces are swapped for the vSRX instance. The AWS ENA detects and rebinds the interface with its corresponding kernel driver.

Multi-Core Scaling Support on AWS with SWRSS and ENA

EC2 instance types are predefined by AWS. You cannot launch an instance with an arbitrary number of vCPUs. This scenario leads to a gap between the resource AWS provides and the resource that vSRX 3.0 can use.

As an example: For AWS C5.4xlarge without software RSS, vSRX 3.0 will be launched with 9 vCPUs. Whereas we have 16 vCPUs that can be used. So, the remaining 7 vCPUs offered by AWS are wasted. With Software RSS, the hardware RSS queue limitation is removed. With more software queue available, more vCPUs can be deployed as data vCPUs.

Starting in Junos OS Release 19.4R1, vSRX 3.0 instances with the Software Receive Side Scaling (SWRSS) feature can scale up the number of vCPUs on instances with ENA support in AWS. The ENA enabled instances allow for more RSS queues. With the SWRSS feature, the dynamic ratio between number of vCPUs and RSS queues allows for the scale up of vSRX with larger AWS EC2 instances.

Software RSS supports up to 32 vCPUs. Launching vSRX into EC2 instance with more than 32 vCPUs will not provide further benefits. To support multi-core scaling you need to ensure SWRSS is enabled on vSRX instances.

With this feature support the AWS instances type supported by vSRX are c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge, and c5.9xlarge. For more information, see Amazon EC2 Instance Types.

Centralized Monitoring and Troubleshooting using AWS Features

IN THIS SECTION

This topic provides you details on how you can perform monitoring and troubleshooting of your vSRX instances on the AWS console by integrating vSRX with CloudWatch, IAM, and Security Hub.

Understanding Centralized Monitoring Using Cloudwatch

IN THIS SECTION

AWS provides a comprehensive view of various metrics, logs, security events from third-party services across AWS accounts. With the support of CloudWatch, vSRX can publish native metrics and logs to cloud, which you can use to monitor vSRX running status. Security Hub is the single place that aggregates, organizes and prioritizes security alerts.

The CloudWatch logs agent provides an automated way to send log data to CloudWatch Logs from Amazon EC2 instances. The agent pushes log data to CloudWatch Logs.

The cloudagent daemon that runs on the vSRX allows integration of AWS CloudWatch and Security Hub. The cloudagent:

- Collects device metrics and send metrics to AWS CloudWatch
- Collects system and security logs and sends the logs to AWS CloudWatchLog

  Any event type (component or log level) that can be collected by the cloudagent under vSRX event log mode is supported for CloudWatch log collection. Events supported for CloudWatchLog are:

  - System activities such as Interfaces status (up/down), configuration changes, user login logout and so on.
  - Security events such as IDP, SkyATP, and security logs such as UTM logs, and Screen, SkyATP and so on.
- Collects security alerts and import those alerts to Security Hub in security finding format.

  To import security events to Security Hub, you need to configure CloudWatch log collection and import the security events based on the log messages.

  Security Hub collects security data from across AWS accounts, services, and supported third-party partners and helps you analyze your security trends and identify the highest priority security issues. After the AWS security hub support is added on vSRX, it helps administrator reduces the effort of collecting and prioritizing security findings across accounts. With the help of Security hub, you can run automated, continuous account-level configuration and compliance checks based on vSRX security output.

For the list of events and metrics that are imported, see Table 70 on page 365 and Table 71 on page 368.

For more information on the events and their purpose, see Juniper System Log Explorer.

Table 70: Events Imported to Security Hub

Metric	Description
AV_MANY_MSGS_NOT_SCANNED_MT	Skip antivirus scanning due to excessive traffic
WEBFILTER_URL_BLOCKED	Web request blocked
AAMW_CONTENT_FALLBACK_LOG	AAMW content fallback info
AV_MANY_MSGS_DROPPED_MT	Drop the received file due to excessive traffic
PFE_SCREEN_MT_CFG_ERROR	screen config failure
WEBFILTER_URL_REDIRECTED	Web request redirected

Table 70: Events Imported to Security Hub *(Continued)*

Metric	Description
AV_FILE_NOT_SCANNED_PASSED_MT	The antivirus scanner passed the received traffic without scanning because of exceeding the maximum content size
RT_SCREEN_TCP_SRC_IP	TCP source IP attack
RT_SCREEN_SESSION_LIMIT	Session limit
SECINTEL_ACTION_LOG	Secintel action info
IDP_APPDDOS_APP_STATE_EVENT	IDP: DDOS application state transition event
AAMW_HOST_INFECTED_EVENT_LOG	AAMW cloud host status event info
IDP_ATTACK_LOG_EVENT	IDP attack log
IDP_SESSION_LOG_EVENT	IDP session event log
AAMW_MALWARE_EVENT_LOG	AAMW cloud malware event info
WEBFILTER_URL_PERMITTED	Web request permitted
IDP_PACKET_CAPTURE_LOG_EVENT	IDP packet captutre event log
RT_SCREEN_WHITE_LIST	Screen white list
RT_SCREEN_IP	IP attack
AAMW_SMTP_ACTION_LOG	AAMW SMTP action info

Table 70: Events Imported to Security Hub *(Continued)*

Metric	Description
RT_SCREEN_TCP_DST_IP	TCP destination IP attack
IDP_APPDDOS_APP_ATTACK_EVENT	IDP: DDOS attack on application
RT_SCREEN_ICMP	ICMP attack
IDP_TCP_ERROR_LOG_EVENT	IDP TCP error log
AV_FILE_NOT_SCANNED_DROPPED_MT	The antivirus scanner dropped the received traffic without scanning because of exceeding the maximum content size
AAMW_ACTION_LOG	AAMW action info
PFE_SCREEN_MT_ZONE_BINDING_ERROR	screen config failure
AV_VIRUS_DETECTED_MT	The antivirus scanner detected a virus
PFE_SCREEN_MT_CFG_EVENT	screen config
RT_SCREEN_TCP	TCP attack
RT_SCREEN_UDP	UDP attack
AV_SCANNER_DROP_FILE_MT	The antivirus scanner dropped the received traffic because of an internal error
AAMW_IMAP_ACTION_LOG	AAMW IMAP action info
AV_SCANNER_ERROR_SKIPPED_MT	Skip antivirus scanning due to an internal error

Table 70: Events Imported to Security Hub *(Continued)*

Metric	Description
AV_MEMORY_INSUFFICIENT_MT	The DRAM size is too small to support antivirus

Table 71: Supported vSRX Metrics Published on CloudWatch by Coudagent

Metric	Unit	Description
ControlPlaneCPUUtil	Percent	Utilization of the CPU on which control plane tasks are running
DataPlaneCPUUtil	Percent	Utilization of each CPU on which data plane tasks are running
DiskUtil	Percent	Disk storage utilization
ControlPlaneMemoryUtil	Percent	Memory utilization of control plane tasks
DataPlaneMemoryUtil	Percent	Memory utilization of data plane task
FlowSessionInUse	Count	Monitors the number of flow session in use, including all those sessions are allocated in valid, invalid, pending and other states.
FlowSessionUtil	Percent	Flow session utilization
RunningProcesses	Count	Number of processes in running state.
Ge00XInputKBPS	Kilobits/Second	Interfaces input statistics on Kilobits per second. Each GE interface will be monitored separately.
Ge00XInputPPS	Count/Second	Interfaces input statistics on packets per second. Each GE interface will be monitored separately.
Ge00XOutputKBPS	Kilobits/Second	Interfaces output statistics on Kilobits per second. Each GE interface will be monitored separately.

Table 71: Supported vSRX Metrics Published on CloudWatch by Coudagent *(Continued)*

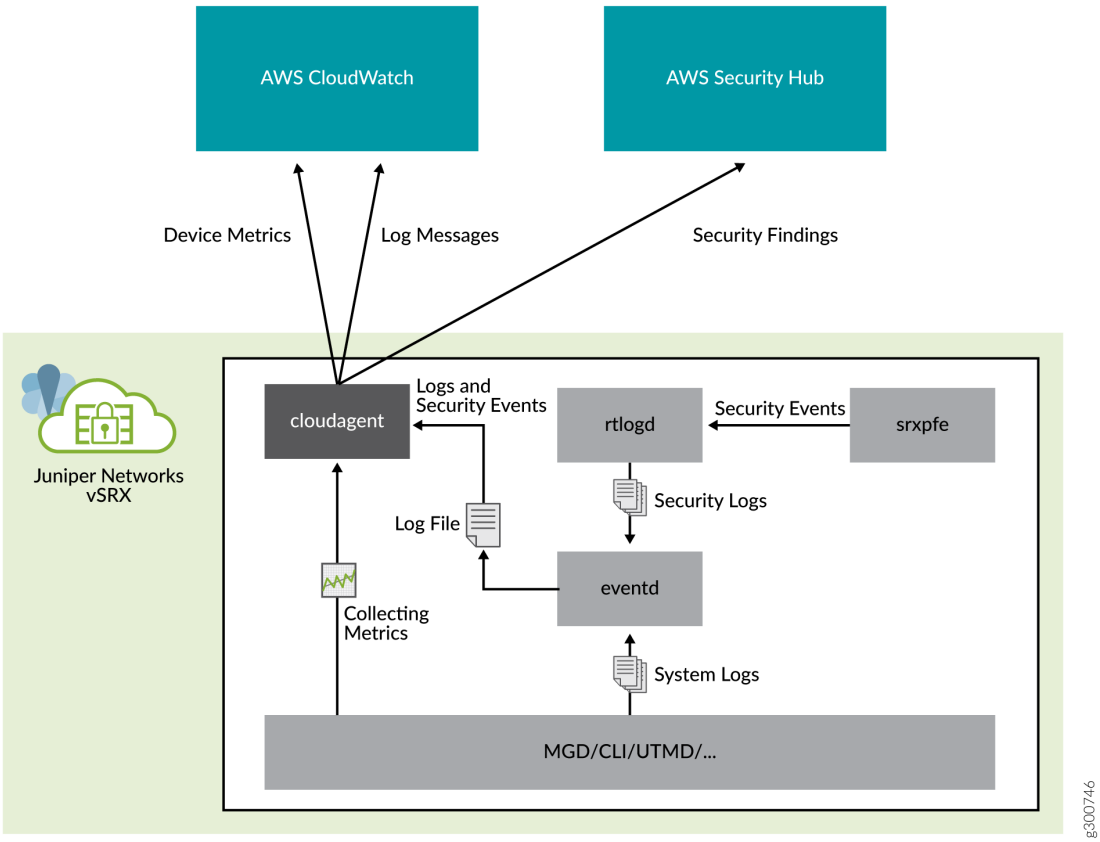
Metric	Unit	Description
Ge00XOutputPPS	Count/Second	Interfaces output statistics on packets per second. Each GE interface will be monitored separately.

Besides the agent running in vSRX, you must configure the AWS console to enable CloudWatch and Security Hub service for vSRX, including:

- Grant privileges for vSRX to post data to CloudWatch and Security Hub
- Create a role with corresponding permission in AWS Identity and Access Management (IAM) console
- Attach the role to vSRX instances in AWS EC2 console
- Configure CloudWatch dashboard to display metric items with chart widget

Figure 88 on page 370 shows how a cloudagent collects data from vSRX and posts to AWS services.

Figure 88: Integration of AWS Cloudwatch on vSRX 3.0



Benefits

- Observability of events and data on a single platform across applications and infrastructure
- Easiest way to collect metric in AWS and on-premises
- Improve operational performance and resource optimization
- Get operational visibility and insight
- Derive actionable insights from logs

CloudWatch Overview

Amazon CloudWatch is a monitoring and management service built for developers, system operators, site reliability engineers (SRE), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your vSRX 3.0 instances running smoothly.

CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, and visualizes it using automated dashboards so you can get a unified view of your AWS resources, applications, and services that run in AWS and on-premises. You can correlate your metrics and logs to better understand the health and performance of your resources. You can also create alarms based on metric value thresholds you specify, or that can watch for anomalous metric behavior based on machine learning algorithms. To take action quickly, you can set up automated actions to notify you if an alarm is triggered and automatically start auto scaling, for example, to help reduce mean-time-to-resolution. You can also dive deep and analyze your metrics, logs, and traces, to better understand how to improve application performance.

Security Hub Overview

AWS Security Hub gives you a comprehensive view of your high-priority security alerts and compliance status across AWS accounts. Security Hub is the single place that aggregates, organizes, and prioritizes security alerts. vSRX supports Security Hub with authentication to post security finding data to Security Hub.

Various security alerts from your vSRX instances are collected by Security Hub. With the integration of Security Hub, you now have a single place that aggregates, organizes, and prioritizes your security alerts, or findings, from your vSRX instances. Your findings are visually summarized on integrated dashboards with actionable graphs and tables. You can also continuously monitor your environment using automated compliance checks based on the AWS best practices and Juniper standards. Enable Security Hub using the management console and once enabled, Security Hub will begin aggregating and prioritizing the findings.

Identity and Access Management Console

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

IAM is a feature of your AWS account offered at no additional charge.

Integration of vSRX with AWS Monitoring and Troubleshooting Features

IN THIS SECTION

This topic provides details on how to integrate CloudWatch and Security Hub with vSRX 3.0 for centralized monitoring and troubleshooting on the AWS console.

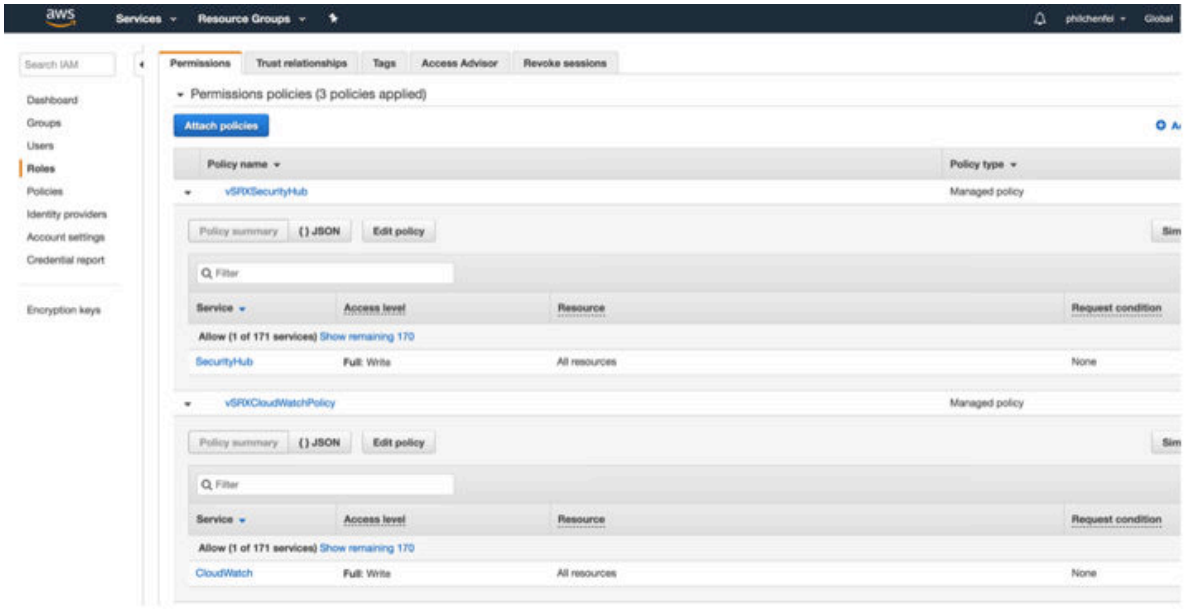
Grant Permission for vSRX to access AWS CloudWatch and Security Hub

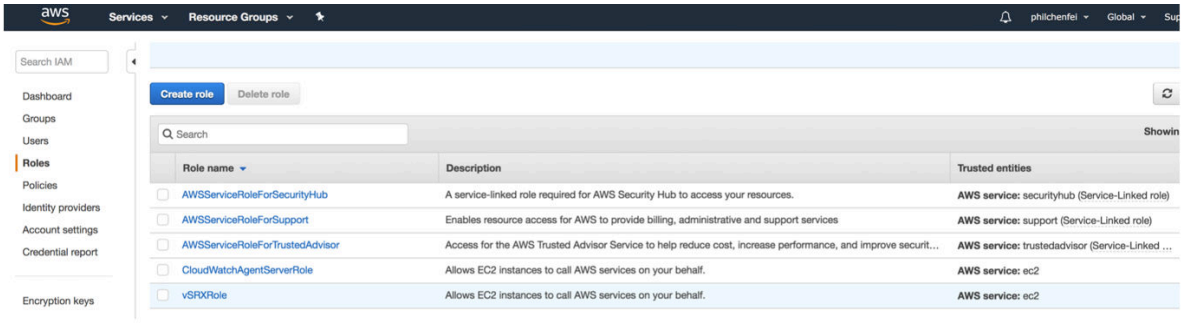
This section provides you details on how to enable access on vSRX instances to interact with AWS CloudWatch and Security Hub.

1. Create an IAM role using AWS IAM console.

   Login to AWS IAM console, create IAM role and attach the role to vSRX instances to grant those permissions. You must create an IAM role before you can launch an instance with that role or attach it to an instance. For more information, see IAM Roles for Amazon EC2.

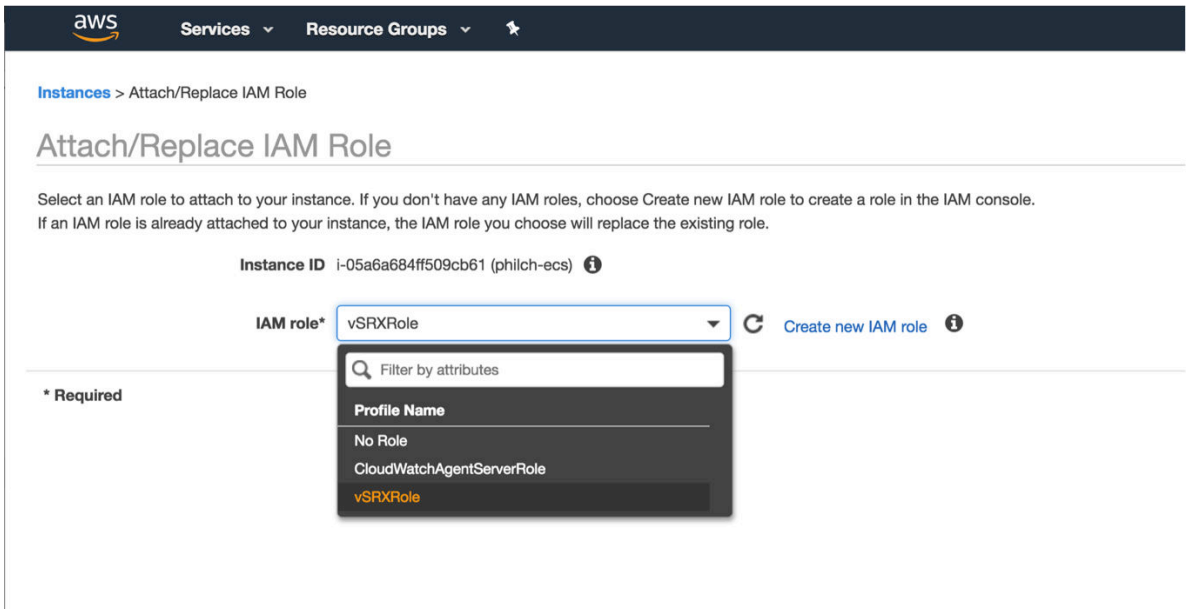
2. Configure an IAM role role on the AWS console and attach the role to vSRX instance. After you create and IAM role, the role can be viewed on IAM console and edited as necessary.





3. To launch an instance with an IAM role or to attach or replace an IAM role for an existing instance, permissions have to be granted to pass the role to the instance. AWS has to grant permission to pass an IAM role to an instance. For more information, see Granting an IAM User Permission to Pass an IAM Role to an Instance.
4. Attach an IAM role to vSRX instances by selecting a IAM role and the vSRX instance ID under the **Attach/Replace IAM Role** tab on the AWS console as shown in Figure 89 on page 373. With the created role, you can enable CloudWatch and Security Hub access for vSRX instance by attaching the role.

Figure 89: Attach or Replace IAM Role to the vSRX Instances



Enable Monitoring of vSRX Instances with AWS CloudWatch Metric

This procedure provides us steps to enable monitoring of vSRX with AWS CloudWatch Metric.

Metric is data about the performance of the system. By enabling CloudWatch Metric monitoring, you can monitor some resources of vSRX instances.

1. Enable CloudWatch and Security Hub using the AWS console.
2. Configure CloudWatch metric in the Cloudwatch agent.

   To enable CloudWatch metric monitoring, you need to configure metric namespace and collection interval on the instance by executing the **# set security cloud aws cloudwatch metric namespace <namespace> collect-interval <integer>** command.

   A namespace is a container for CloudWatch metrics. Metric in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics. Different vSRX instances can use same CloudWatch metric namespace. Metric from different vSRX instances can be differentiated by dimensional data (instances id/name) in metric value.

   Collection interval is the frequency at which the firewall publishes the metrics to CloudWatch. The value can be set between 1 minute and 60 minutes. The default value is 3 minutes.

   Once the Cloudwatch metric monitoring is enabled, the cloudagent running on vSRX collects all the required metric and publishes the metric data on the Cloudwatch.

   Once monitoring is enabled you can view CloudWatch Metric. CloudWatch metric can be graphed after cloudagent starts to collect and post metric data to the cloud. By selecting the metric namespaces created from vSRX on AWS CloudWatch console, administrator can check and display all metric data. Check AWS CloudWatch guide for how to filter and display on those collected metric.
3. View the Cloudwatch metric data. CloudWatch metrics can be graphed after cloudagent starts to collect and post metric data to cloud.
4. Configure CloudWatch dashboard to display metric items with chart widget.

   Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view, even those resources that are spread across different regions. You can manually create a dashboard for the vSRX under monitoring.

Collect, Store, and View vSRX Logs to AWS CloudWatch

CloudWatch Logs are used to monitor, store, and access log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources. For a vSRX instance, cloudagent collects both system and security logs and then post these logs to CloudWatchLog. The log collection in cloudagent will cache logs in a time window and post them to CloudWatchLog in a batch.

This procedure provides you details on how to enable and configure CloudWatch Logs on vSRX

1. To enable log collection for CloudWatchLog, you need to configure a log group, collect interval and from which file to collect log messages on the device.

```
# set security cloud aws cloudwatch log group vsrx-group
# set security cloud aws cloudwatch log file mylog collect-interval 2
# set security cloud aws cloudwatch log file syslog collect-interval 1
```

   A log stream is a sequence of log events that share the same source. Each separate source of logs into CloudWatch Logs makes up a separate log stream. For log collection, one vSRX will post logs as a dedicated stream which means vSRX will automatically create a log stream in the destination log group.

   A log group is a group of log streams that share the same retention, monitoring, and access control settings. By defining the log groups on the vSRX instance, yiu can specify which streams are placed into which group.

   Collection interval is the frequency at which the firewall publishes logs to CloudWatchLog. The value can be set between 1 minute and 60 minutes. The default value is 3 minutes.

   Three vSRX log files can be collected in CloudWatch simultaneously per vSRX instance. Each log file will create a corresponding a log stream in Cloudwatch. The log stream will be named under log group with convention <vsrx_instance_id> <log_file_name>.

   After you enable CloudWatch logging in the cloudagent on vSRX instances, you need to configure syslog message file.

2. Configure the syslog message file.

   Any filters can be applied based on vSRX syslog filtering. It provides the capability to define which log messages will be sent to CloudWatchLogs. For example, the below configuration means system will log any error messages to the syslog file under the **/var/log** and cloudagent will collect the messages from **/var/log/syslog** and post the messages to CloudWatchLogs.

```
# set security cloud aws cloudwatch log file syslog collect-interval 1
# set system syslog file syslog any error
```

3. View and search vSRX logs on CloudWatchLog console. Log groups and stream will be created automatically after configured on vSRX instances.

   Select the log group and stream to check and search those logs sent to CloudWatch from the vSRX instance.

Enable and Configure Security Hub on vSRX

To import security events to AWS Security Hub, you need to configure CloudWatch log collection and then import the security events based on the log messages.

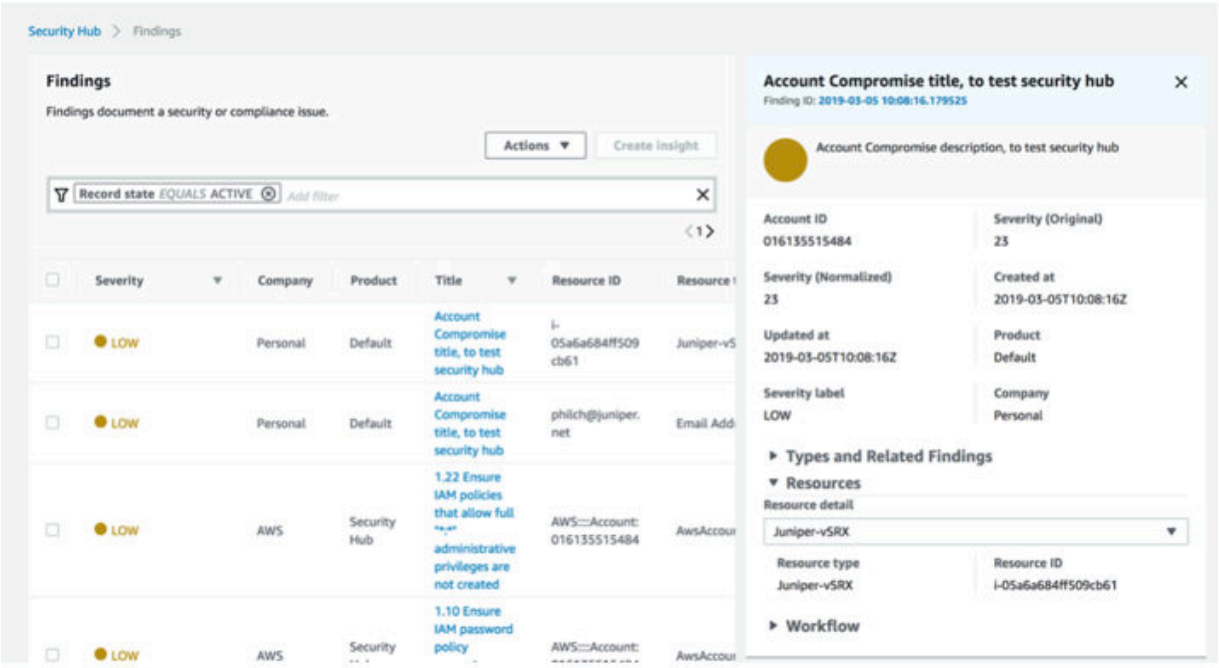
For example:

```
# set security cloud aws cloudwatch log group vsrx-group

# set security cloud aws cloudwatch log file mylog security-hub-import

# set security cloud aws cloudwatch log file mylog collect-interval 1

# set system syslog file mylog any any

# set system syslog file mylog structured-data
```

In the above configuration you are configuring CloudWatch log collection on file mylog under **/var/log** directory and any security events in the log file will be imported from the vSRX to Security Hub in the AWS security finding format.

> **NOTE**: The `security-hub-import` option is only supported on log files with structured-data format. Which means if a message is logged with plain text format, security events in log messages cannot be converted to AWS security finding and imported to Security Hub.

You can view the security findings posted from vSRX on the Security Hub console.



Configure vSRX Using the CLI

IN THIS SECTION

Understand vSRX on AWS Preconfiguration and Factory Defaults

vSRX on AWS deploys with the following preconfiguration defaults:

- SSH access with the RSA key pair configured during the installation
- No password access allowed for SSH access
- The management (fxp0) interface is preconfigured with the AWS Elastic IP and default route

Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, the following example summarizes the preconfiguration statements added to a factory-default configuration for vSRX on AWS instances:

```
set groups aws-default system root-authentication ssh-rsa "ssh-rsa XXXRSA-KEYXXXXX”
set groups aws-default system services ssh no-passwords
set groups aws-default system services netconf ssh
set groups aws-default system services web-management https system-generated-certificate
set groups aws-default interfaces fxp0 unit 0 family inet address aws-ip-address
set groups aws-default routing-options static route 0.0.0.0/0 next-hop aws-ip-address
set apply-groups aws-default
```

For Junos OS Release 15.1X49-D70 and earlier, the following example summarizes the preconfiguration statements added to a factory-default configuration for vSRX on AWS instances:

```
set system root-authentication ssh-rsa "ssh-rsa XXXRSA-KEYXXXXX”
set system services ssh no-passwords
set interfaces fxp0 unit 0 family inet addressaws-ip-address
set routing-options static route 0.0.0.0/0 next-hop aws-ip-address
```

CAUTION: Do not use the `load factory-default` command on a vSRX AWS instance. The factory default configuration removes the AWS preconfiguration. If you must revert to factory default, ensure that you manually reconfigure AWS preconfiguration statements before you commit the configuration; otherwise, you will lose access to the vSRX instance.

Add a Basic vSRX Configuration

You can either create a new configuration on vSRX or copy an existing configuration from another SRX or vSRX and load it onto your vSRX on AWS. Use the following steps to copy and load an existing configuration:

1. Saving a Configuration File
2. Loading a Configuration File

To configure a vSRX instance using the CLI:

1. Log in to the vSRX instance using SSH and start the CLI.

   NOTE: Starting in Junos OS Release 17.4R1, the default user name has changed from `root@` to `ec2-user@`.

   ```
   ec2-user@% cli
   ec2-user@>
   ```

2. Enter configuration mode.

   ```
   ec2-user@> configure
   [edit]
   ec2-user@#
   ```

3. Set the authentication method to log into the vSRX. You can specify a password by entering a cleartext password or an encrypted password. If you require a more robust level of authentication security, we recommend that you select an SSH public key string (DSA, ECDSA, or RSA).

   ```
   ec2-user@# set system root-authentication ssh-rsa <public-key>
   ```

or

```
ec2-user@# set system root-authentication plain-text-password
New password: password
Retype new password: password
```

4. Optionally, enable passwords for SSH if you want to create password access for additional users.

```
ec2-user@# delete services ssh no-passwords
```

5. Configure the hostname.

```
ec2-user@# set system host-name host-name
```

6. For each vSRX revenue interface, assign the IP address defined on AWS. For example:

```
ec2-user@# set interfaces ge-0/0/0 unit 0 family inet address 10.0.10.197/24
```

For multiple private addresses, enter a `set` command for each address. Do not assign the Elastic IP address.

7. Specify a security zone for the public interface.

```
ec2-user@# set security zones security-zone untrust interfaces ge-0/0/0.0
```

8. Specify a security zone for the private interface.

```
ec2-user@# set security security-zone trust interfaces ge-0/0/1.0
```

9. Configure routing to add a separate virtual router and routing option for the public and private interfaces.

NOTE: We recommend putting the revenue (data) interfaces in routing instances as a best practice to avoid asymmetric traffic/routing, because fxp0 is part of the default (inet.0) table by default. With fxp0 as part of the default routing table, there might be two default routes needed: one for the fxp0 interface for external management access, and the other for the

revenue interfaces for traffic access. Putting the revenue interfaces in a separate routing instance avoids this situation of two default routes in a single routing instance.

```
set routing-instances aws instance-type virtual-router
set routing-instances aws interface ge-0/0/0.0
set routing-instances aws interface ge-0/0/1.0
set routing-instances aws interface st0.1
set routing-instances aws routing-options static route 0.0.0.0/0 next-hop 10.0.0.1
set routing-instances aws routing-options static route 10.20.20.0/24 next-hop st0.1
```

10. Verify the configuration.

```
ec2-user@# commit check
configuration check succeeds
```

11. Commit the configuration to activate it on the device.

```
ec2-user@# commit
commit complete
```

12. Optionally, use the `show` command to display the configuration to verify that it is correct.

For an example of how to configure vSRX to NAT all hosts behind the vSRX instance in the Amazon Virtual Private Cloud (Amazon VPC) to the IP address of the vSRX egress interface on the untrust zone, see *Example: Configuring NAT for vSRX*. This configuration allows hosts behind vSRX in a cloud network to access the Internet.

For an example of how to configure IPsec VPN between two instances of vSRX on AWS on different Amazon VPCs, see *Example: Configure VPN on vSRX Between Amazon VPCs*.

Add DNS Servers

vSRX does not include any DNS servers in the default configuration. You might need DNS configured to deploy Layer 7 services, such as IPS, to pull down signature updates, for example. You can use your own external DNS server or use an AWS DNS server. If you enable DNS on your Amazon VPC, queries to the Amazon DNS server (169.254.169.253) or the reserved IP address at the base of the VPC network range plus two should succeed. See AWS - Using DNS with Your Amazon VPC for complete details.

Add vSRX Feature Licenses

Certain Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds to each licensed

feature. To conform to software feature licensing requirements, you must purchase one license per feature per instance. The presence of the appropriate software unlocking key on your virtual instance allows you to configure and use the licensed feature.

See Managing Licenses for vSRX for details.

RELATED DOCUMENTATION

- CLI User Guide
- AWS - Using DNS with Your VPC

Configure vSRX Using the J-Web Interface

IN THIS SECTION

Access the J-Web Interface and Configure vSRX

To configure vSRX using the *J-Web* Interface:

1. Enter the AWS Elastic IP address of the eth0 interface in the browser Address box.
2. Specify the username and password.
3. Click **Log In**, and select the **Configuration Wizards** tab from the left navigation panel. The J-Web Setup Wizard page opens.
4. Click **Setup**.

   You can use the Setup wizard to configure a device or edit an existing configuration.

   - Select **Edit Existing Configuration** if you have already configured the wizard using the factory mode.
   - Select **Create New Configuration** to configure a device using the wizard.

The following configuration options are available in the guided setup:

- Basic

  Select **basic** to configure the device name and user account information as shown in Table 72 on page 382.

  - Device name and user account information

    Table 72: Device Name and User Account Information

Field	Description
Device name	Type the name of the device. For example: **vSRX**.
Root password	Create a default root user password.
Verify password	Verify the default root user password.
Operator	Add an optional administrative account in addition to the root account. User role options include: • **Superuser**: This user has full system administration rights and can add, modify, and delete settings and users. • **Operator**: This user can perform system operations such as a system reset but cannot change the configuration or add or modify users. • **Read only**: This user can only access the system and view the configuration. • **Disabled**: This user cannot access the system.

- Select either **Time Server** or **Manual**. Table 73 on page 382 lists the system time options.

  Table 73: System Time Options

Field	Description
Time Server	
Host Name	Type the hostname of the time server. For example: **ntp.example.com**.

Table 73: System Time Options *(Continued)*

Field	Description
IP	Type the IP address of the time server in the IP address entry field. For example: **192.168.1.254**.
NOTE: You can enter either the hostname or the IP address.	
Manual	
Date	Click the current date in the calendar.
Time	Set the hour, minute, and seconds. Choose **AM** or **PM**.
Time Zone (mandatory)	
Time Zone	Select the time zone from the list. For example: GMT Greenwich Mean Time GMT.

- Expert

  a. Select **Expert** to configure the basic options as well as the following advanced options:

  - Four or more internal zones
  - Internal zone services
  - Application of security policies between internal zones

  b. Click **Need Help** for detailed configuration information.

  You see a success message after the basic configuration is complete.

Apply the Configuration Settings for vSRX

To apply the configuration settings for vSRX:

1. Review and ensure that the configuration settings are correct, and click **Next**. The Commit Configuration page appears.
2. Click **Apply Settings** to apply the configuration changes to vSRX.
3. Check the connectivity to vSRX, because you might lose connectivity if you have changed the management zone IP. Click the URL for reconnection instructions on how to reconnect to the device.

4. Click **Done** to complete the setup.

   After successful completion of the setup, you are redirected to the J-Web interface.

> **CAUTION**: After you complete the initial setup, you can relaunch the J-Web Setup wizard by clicking **Configuration>Setup**. You can either edit an existing configuration or create a new configuration. If you create a new configuration, the current configuration in vSRX will be deleted.

Add vSRX Feature Licenses

Certain Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds to each licensed feature. To conform to software feature licensing requirements, you must purchase one license per feature per instance. The presence of the appropriate software unlocking key on your virtual instance allows you to configure and use the licensed feature.

See Managing Licenses for vSRX for details.

Upgrade Junos OS Software on a vSRX Instance

IN THIS SECTION

This section outlines how to upgrade Junos OS software on your vSRX instance to a newer release. Depending upon your preference, you can replace the vSRX software in one of two ways:

Upgrade the Junos OS for vSRX Software Release

You can directly upgrade the Junos OS for vSRX software using the CLI. Upgrading or downgrading Junos OS can take several hours, depending on the size and configuration of the network. You download the desired Junos OS Release for vSRX .tgz file from the Juniper Networks website.

You also can upgrade using J-Web (see J-Web) or the Junos Space Network Management Platform (see Junos Space).

For the procedure on upgrading a specific Junos OS for vSRX software release, see the *Migration, Upgrade, and Downgrade Instructions* topic in the release-specific *vSRX Release Notes* available on the vSRX TechLibrary.

Replace the vSRX Instance on AWS

To replace a vSRX instance on AWS with a different software release:

1. Log in to the vSRX instance using SSH and start the CLI.

   NOTE: Starting in Junos OS Release 17.4R1, the default user name has changed from `root@` to `ec2-user@`.

   ```
   ec2-user@% cli
   ec2-user@>
   ```

2. Enter configuration mode.

   ```
   ec2-user@> configure
   [edit]
   ec2-user@#
   ```

3. Copy the existing Junos OS configuration from the vSRX. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it.

   NOTE: By default, the configuration is saved to a file in your home directory.

   - See Saving a Configuration File for additional background information on saving a Junos OS configuration file.
   - See file copy for information on how to copy files from one location to another location on the local device or to a location on a remote device that is reachable by the local device.

   ```
   ec2-user@#save <filename>
   [edit]
   ec2-user@#
   ```

4. Remove the vSRX instance on AWS as described in *Remove a vSRX Instance on AWS*.

5. Once the vSRX instance on AWS has been successfully removed, define the specifics of a vSRX instance prior to launching it. See *Configure an Amazon Virtual Private Cloud for vSRX*.
6. Launch the vSRX image using the desired software version available from AWS Marketplace as described in *Launch a vSRX Instance on an Amazon Virtual Private Cloud*
7. Load the previously copied Junos OS configuration file onto your new (upgraded) vSRX instance as described in Loading a Configuration File.

Remove a vSRX Instance on AWS

To remove a vSRX instance on AWS:

1. Log in to the AWS Management Console and select **Services > Compute > EC2 > Instances**.
2. Select the vSRX instance and select **Actions > Instance State > Terminate** to remove the instance.
3. In the dialog box, expand the section and select **Release associated Elastic IP**.
4. Click **Yes, Terminate**.

NOTE: See Deleting Your VPC to remove any unused VPCs from AWS.

PART 7

vSRX Deployment for Microsoft Azure

CHAPTER 20

Overview

IN THIS CHAPTER

Understand vSRX with Microsoft Azure Cloud

IN THIS SECTION

This section presents an overview of vSRX as deployed in the Microsoft Azure cloud.

vSRX with Microsoft Azure

Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX to the Microsoft Azure Cloud. Microsoft Azure is Microsoft's application platform for the public cloud. It is an open, flexible, enterprise-grade cloud computing platform for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It provides Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) services. You place your virtual machines (VMs) onto Azure virtual networks, where the distributed and virtual networks in Azure help ensure that your private network traffic is logically isolated from traffic on other Azure virtual networks.

The Azure WALinuxAgent performs the provisioning job for the vSRX instances. When a new vSRX instance is deployed, the continued increasing size of the waagent log file might cause the vSRX to stop. If the vSRX is still operating, then delete the **/var/log/waagent.log** directly or run the `clear log waagent.log all` command to clear the log file.

Or you can run the `set groups azure-provision system syslog file waagent.log archive size 1m` and `set groups azure-provision system syslog file waagent.log archive files 10` commands to prevent the growing of the waagent logs. These configurations will cause the rotation of log of waagent with the size bigger than 1MB and set a maximum of 10 backups.

You can add a vSRX virtual security appliance to provide networking security features as an application instance within an Azure virtual network. The vSRX protects the workloads that run within the virtual network on the Microsoft Azure Cloud.

You can deploy the vSRX VM in Azure using the following deployment methods:

- Azure Marketplace—Deploy the vSRX VM from the Azure Marketplace. The Azure Marketplace provides you with different methods to deploy a vSRX VM in your virtual network. You can choose a customized solution template offered by Juniper Networks to automate the vSRX VM deployment based on specific use cases (for example, a security gateway). A solution template automates the dependencies associated with specific deployment use cases, such as VM settings, virtual network settings (such as multiple subsets for the management interface (fxp0) and two revenue (data) interfaces), and so on. Or, you can select the vSRX VM image and define the deployment settings and dependencies based on your specific networking requirements. Starting in Junos OS Release 15.1X49-D91 for vSRX, you can deploy the vSRX to Microsoft Azure Cloud from the Azure Marketplace.

  Azure Marketplace also enables you to discover and subscribe to software that supports regulated workloads through Azure Marketplace for Azure Government Cloud (US).

- Azure CLI—Deploy the vSRX VM from the Azure CLI. You can customize the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud. To help automate and simplify the deployment of the vSRX VM in the Microsoft Azure virtual network, Juniper Networks provides a series of scripts, Azure Resource Manager (ARM) templates and parameter files, and configuration files in a GitHub repository.

  NOTE: Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX to Microsoft Azure Cloud from the Azure CLI.

In Microsoft Azure, you can host servers and services on the cloud as a pay-as-you-go (PAYG) or bring-your-own-license (BYOL) service.

NOTE: vSRX PAYG images do not require any Juniper Networks licenses.

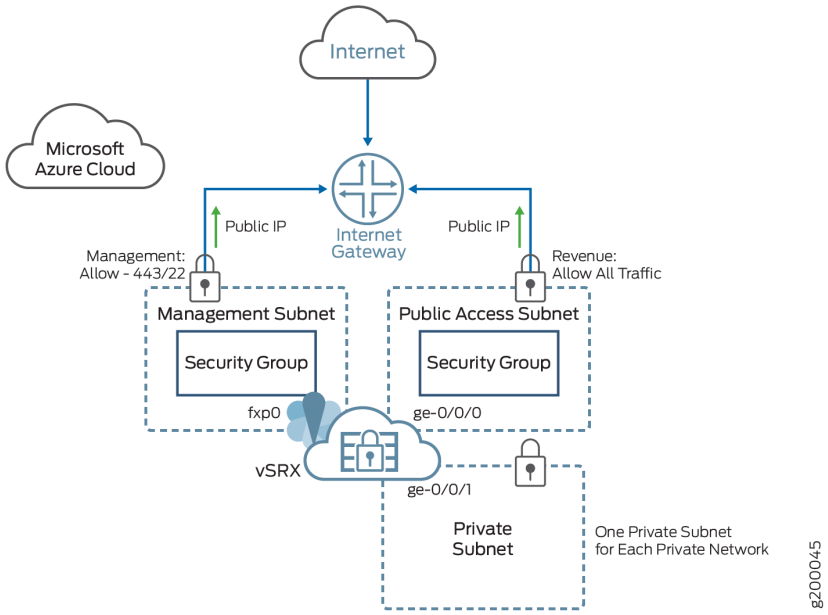
Starting in Junos OS Release 15.1X49-D120, vSRX on Microsoft Azure Cloud supports the vSRX Premium-Next Generation Firewall with Anti-Virus Protection bundle for PAYG, available as 1-hour or 1-year subscriptions. This bundle includes:

- Standard (STD) features of core security, including core firewall, IPsec VPN, NAT, CoS, and routing services.
- Advanced Layer 4 through 7 security services such as AppSecure features of AppID, AppFW, AppQoS, and AppTrack, IPS and rich routing capabilities, including the UTM antivirus feature.

Figure 90 on page 390 illustrates the deployment of a vSRX in Microsoft Azure.

In the Microsoft Azure, public subnets have access to the Internet gateway, but private subnets do not. vSRX requires two public subnets and one or more private subnets for each individual instance group. The public subnets consist of one for the management interface (fxp0) and one for a revenue (data) interface. The private subnets, connected to the other vSRX interfaces, ensure that all traffic between applications on the private subnets and the Internet must pass through the vSRX instance.

Figure 90: vSRX Deployed to Microsoft Azure



For a glossary of Microsoft Azure terms see Microsoft Azure glossary.

Release History Table

Release	Description
15.1X49-D91	Starting in Junos OS Release 15.1X49-D91 for vSRX, you can deploy the vSRX to Microsoft Azure Cloud from the Azure Marketplace.

15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX to the Microsoft Azure Cloud.
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX to Microsoft Azure Cloud from the Azure CLI.
15.1X49-D120	Starting in Junos OS Release 15.1X49-D120, vSRX on Microsoft Azure Cloud supports the vSRX Premium-Next Generation Firewall with Anti-Virus Protection bundle for PAYG, available as 1-hour or 1-year subscriptions.

RELATED DOCUMENTATION

- Microsoft Azure
- Azure Virtual Networks
- Microsoft Azure portal overview

Requirements for vSRX on Microsoft Azure

IN THIS SECTION

This section presents an overview of requirements for deploying a vSRX instance on Microsoft Azure Cloud.

System Requirements for vSRX on Microsoft Azure Cloud

Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX to the Microsoft Azure Cloud. Microsoft Azure supports a wide variety of sizes and options for deployed Azure virtual machines (VMs).

For the vSRX deployment in Microsoft Azure, we recommend DSv2-series VMs. The DSv2-series VMs provided from Microsoft Azure use Premium Storage(SSD) and are ideal for applications that demand faster CPUs and better local disk performance, or have higher memory demands. Of the available DSv2-series VMs, we recommend that you select Standard_DS3_v2, Standard_DS4_v2, or Standard_DS5_v2 for the vSRX VM deployment in Microsoft Azure. For more details, see DSv2-series.

Table 74 on page 392 lists the properties of the Standard_DS3_v2 VM available in Microsoft Azure.

Table 74: Properties of the Standard_DS3_v2 VM in Microsoft Azure

Component	Specification
Size	Standard_DS3_v2
CPU cores	4
Memory	14 GiB
Maximum number of data disks	16
Maximum cached and local disk storage throughput: IOPS/MBps (cache size in GB)	16,000/128 (172)
Maximum uncached disk throughput: IOPS/MBps	12,800/192
Max NICs/Expected network bandwidth (Mbps)	4/3000

Table 75 on page 393 lists the properties of the Standard_DS4_v2 VM available in Microsoft Azure.

Table 75: Properties of the Standard_DS4_v2 VM in Microsoft Azure

Component	Specification
Size	Standard DS4_v2
CPU cores	8
Memory	28 GiB
Maximum number of data disks	32
Temp storage (SSD) GiB	56
Max cached and temp storage throughput: IOPS/MBps (cache size in GiB)	32000/256 (344)
Max uncached disk throughput: IOPS/MBps	25600/384
Max NICs/Expected network bandwidth (Mbps)	8/6000

NOTE: The vSRX does not provide support for a high availability configuration in Microsoft Azure. In addition, the vSRX does not support Layer 2 transparent mode in Microsoft Azure.

Table 76 on page 393 lists the properties of the Standard_DS5_v2 VM available in Microsoft Azure.

Table 76: Properties of the Standard_DS5_v2 VM in Microsoft Azure

Component	Specification
Size	Standard DS5_v2
CPU cores	16

Table 76: Properties of the Standard_DS5_v2 VM in Microsoft Azure *(Continued)*

Component	Specification
Memory	56 GiB
Maximum number of data disks	64
Temp storage (SSD) GiB	112
Max cached and temp storage throughput: IOPS/MBps (cache size in GiB)	64000/512 (688)
Max uncached disk throughput: IOPS/MBps	51200/768
Max NICs/Expected network bandwidth (Mbps)	8/12000

Network Requirements for vSRX on Microsoft Azure Cloud

When you deploy a vSRX VM in a Microsoft Azure virtual network, note the following specifics of the deployment configuration:

- A dual public IP network configuration is a requirement for vSRX VM network connectivity; the vSRX VM requires two public subnets and one or more private subnets for each instance group.
- The public subnets required by the vSRX VM consist of one subnet for the out-of-band management interface (fxp0) for management access and another for the two revenue (data) interfaces. By default, one interface is assigned to the untrust security zone and the other to the trust security zone on the vSRX VM.
- In the Microsoft Azure deployment of the vSRX VM, the vSRX supports the management interface (fxp0) and the two revenue (data) interfaces (port ge-0/0/0 and ge-0/0/1), which includes public IP address mapping and data traffic forwarding to and from the vSRX VM.

Microsoft Azure Instances and vSRX Instance Types

Microsoft Azure instance types supported for vSRX are listed in Table 77 on page 395.

Table 77: Supported Microsoft Azure Instance Types for vSRX

Instance Type	vSRX Type	vCPUs	Memory in Instance Type (GB)	RSS Type
Standard_DS3_v2	VSRX-4CPU-14G memory	4	14	HWRSS
Standard_DS4_v2	VSRX-8CPU-28G memory	8	28	HWRSS
Standard_DS5_v2	VSRX-16CPU-56G memory	16	56	HWRSS

Interface Mapping for vSRX on Microsoft Azure

Table 78 on page 395 lists the vSRX and Microsoft Azure interface names. The first network interface is used for the out-of-band management (fxp0) for vSRX.

Table 78: vSRX and Microsoft Azure Interface Names

Interface Number	vSRX Interface	Microsoft Azure Interface
1	fxp0	eth0
2	ge-0/0/0	eth1
3	ge-0/0/1	eth2
4	ge-0/0/2	eth3
5	ge-0/0/3	eth4
6	ge-0/0/4	eth5

Table 78: vSRX and Microsoft Azure Interface Names *(Continued)*

Interface Number	vSRX Interface	Microsoft Azure Interface
7	ge-0/0/5	eth6
8	ge-0/0/6	eth7

NOTE: Refer Dv2 and DSv2-series for information on maximum number of NICs supported per Azure instance type.

We recommend putting revenue interfaces in routing instances as a best practice to avoid asymmetric traffic/routing, because fxp0 is part of the default (inet.0) table by default. With fxp0 as part of the default routing table, there might be two default routes needed: one for the fxp0 interface for external management access, and the other for the revenue interfaces for traffic access. Putting the revenue interfaces in a separate routing instance avoids this situation of two default routes in a single routing instance. Ensure that interfaces belonging to the same security zone are in the same routing instance.

vSRX Default Settings on Microsoft Azure

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.

Table 79 on page 396 lists the factory-default settings for security policies on the vSRX

Table 79: Factory-Default Settings for Security Policies

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

CAUTION: Do not use the `load factory-default` command on the vSRX instance in Microsoft Azure. The factory-default configuration removes the "azure provision" preconfiguration. This group contains critical system-level settings and route information for the vSRX. A misconfiguration in the group "azure-provision" may result in the possible loss of connectivity to vSRX from Microsoft Azure. If you must revert to factory default, ensure that you first manually reconfigure the Microsoft Azure preconfiguration statements before you commit the configuration; otherwise, you will lose access to the vSRX instance.

We strongly recommend that when you commit a configuration, perform an explicit `commit confirmed` to avoid the possibility of losing connectivity to vSRX. Once you have verified that the change works correctly, you can keep the new configuration active by entering the `commit` command within 10 minutes. Without the timely second confirm, configuration changes will be rolled back. See *Configure vSRX Using the CLI* for preconfiguration details.

Best Practices for Improving vSRX Performance

Review the following deployment practices to improve vSRX performance:

- Disable the source/destination check for all vSRX interfaces.
- Limit public key access permissions to 400 for key pairs.
- Ensure that there are no contradictions between Microsoft Azure security groups and your vSRX configuration.
- Use vSRX NAT to protect your instances from direct Internet traffic.

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX to the Microsoft Azure Cloud.

RELATED DOCUMENTATION

KB Article - Interface must be in the same routing instance as the other interfaces in the zone

Windows virtual machines in Azure

CHAPTER 21

Deploy vSRX from the Azure Portal

IN THIS CHAPTER

Before You Deploy vSRX from the Azure Portal

You can deploy a vSRX virtual security appliance and its advanced security features in your virtual network directly from the Azure portal. This method provides a browser-based user interface for creating and configuring virtual machines and all related resources.

The Azure Marketplace provides you with different methods to deploy a vSRX virtual machine (VM) in a virtual network. You can choose a customized solution template offered by Juniper Networks in the Azure Marketplace to automate the vSRX deployment based on a specific use case (for example, a security gateway).

Solution templates allow the bundling of multiple Azure services and a software image into a template that enables you to quickly deploy a preconfigured solution. You access vSRX solution templates from the Azure Marketplace to simplify the end-to-end configuration steps involved in deploying a vSRX VM in your Azure virtual network. A solution template automates the dependencies associated with specific deployment use cases, such as VM settings, virtual network settings (such as multiple subsets for the management interface (fxp0) and two revenue (data) interfaces), and so on.

A vSRX solution template is based on a custom Microsoft Azure Resource Manager (ARM) template. The ARM template consists of JavaScript Object Notation (JSON) expressions that construct specific values for your vSRX deployment. To integrate with the Azure portal, each solution template uses **mainTemplate.json** and **createUiDefinition.json** files to define the components of the customized solution template for vSRX VM deployment.

You also have the option to select the vSRX image from Azure Marketplace and customize the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud. This deployment approach might be required if you have a vSRX VM deployment scenario that is outside of the use cases offered in the vSRX VM solution templates available from Juniper Networks.

Before you deploy the vSRX virtual security appliance from the Azure Marketplace:

- Review the requirements for deploying a vSRX VM in Microsoft Azure Cloud in *Requirements for vSRX on Microsoft Azure*.
- Obtain an account for and a subscription to Microsoft Azure (see Microsoft Azure).
- Use your Microsoft account username and password to log into the Microsoft Azure portal.
- Purchase a vSRX license or request an evaluation license. Licenses can be procured from the Juniper Networks License Management System (LMS).
- Ensure that your Azure subscription includes the following for your vSRX VM:
  - Resource group, as described in *Create a Resource Group*.
  - Storage account, as described in *Create a Storage Account*.
  - Virtual network, as described in *Create a Virtual Network*.

RELATED DOCUMENTATION

Microsoft Azure portal

Microsoft Azure portal overview

Create a Resource Group

A resource group contains the resources required to successfully deploy a vSRX VM in Azure. It is a container that holds related resources for an Azure solution. In Azure, you logically group related resources such as storage accounts, virtual networks, and virtual machines (VMs) to deploy, manage, and maintain them as a single entity.

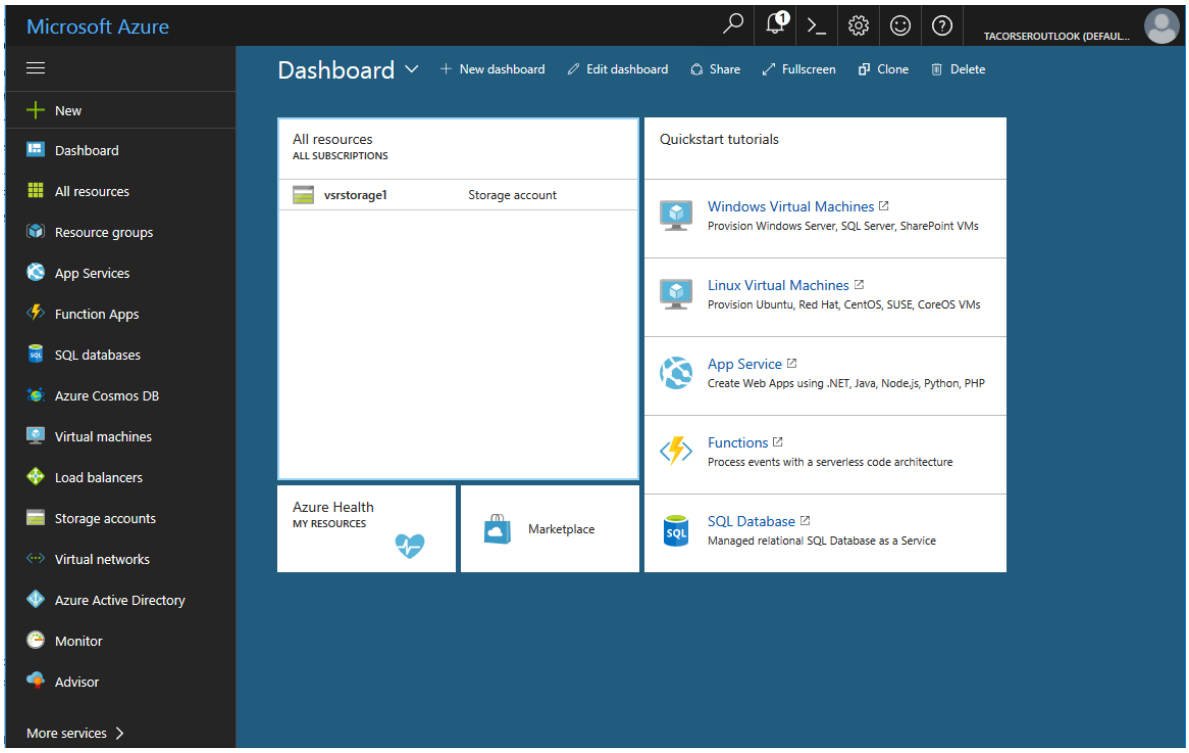
If you do not have an existing resource group in your subscription, then follow the steps outlined in this procedure.

To create a resource group in Azure:

1. Log in to the Microsoft Azure portal using your Microsoft account username and password. The Dashboard appears in the Azure portal (see Figure 91 on page 400). You see a unified dashboard for

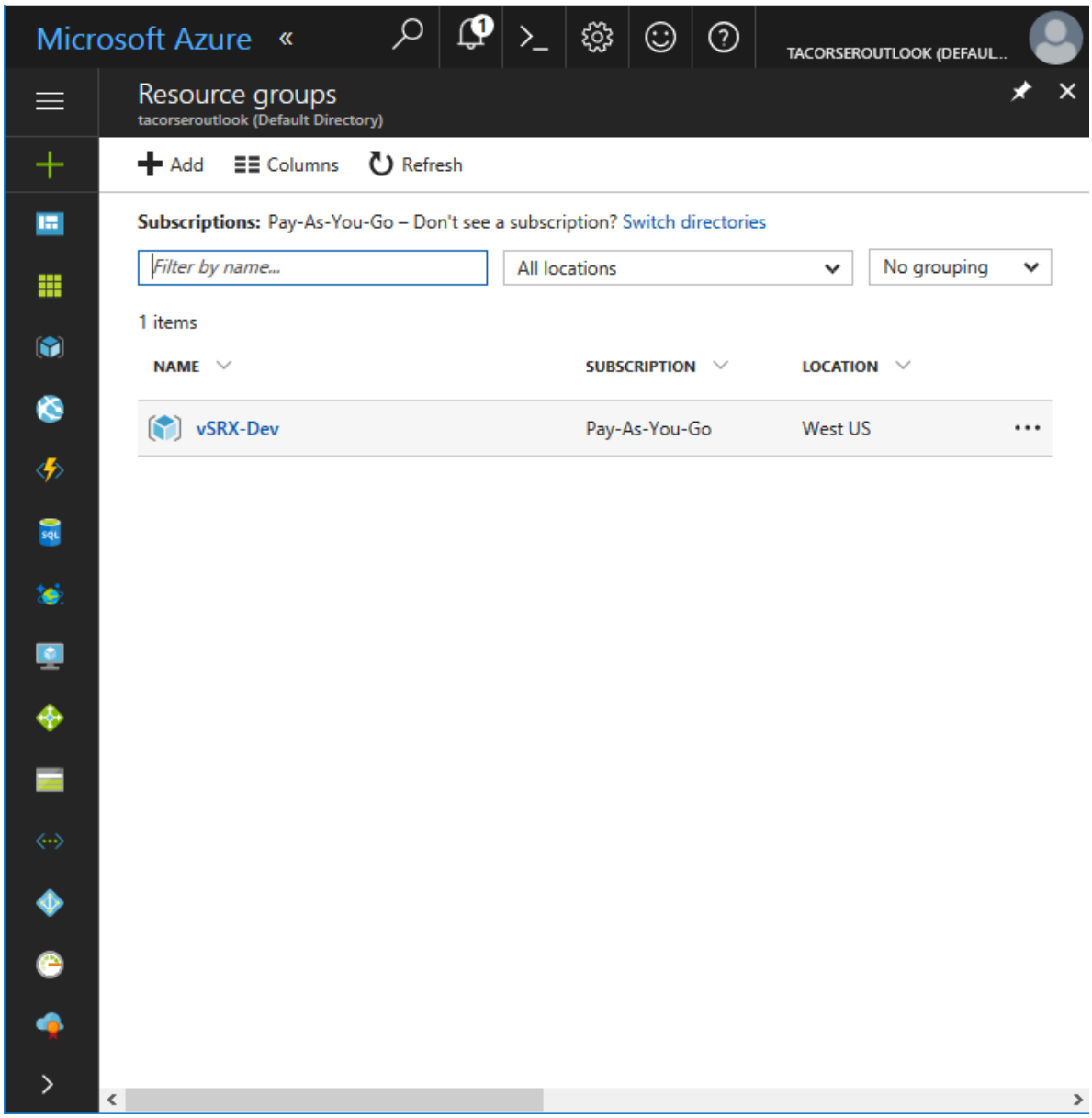
all your assets in Azure. Verify that the dashboard includes all subscriptions to which you currently have access, and all resource groups and associated resources.

Figure 91: Microsoft Azure Portal Dashboard



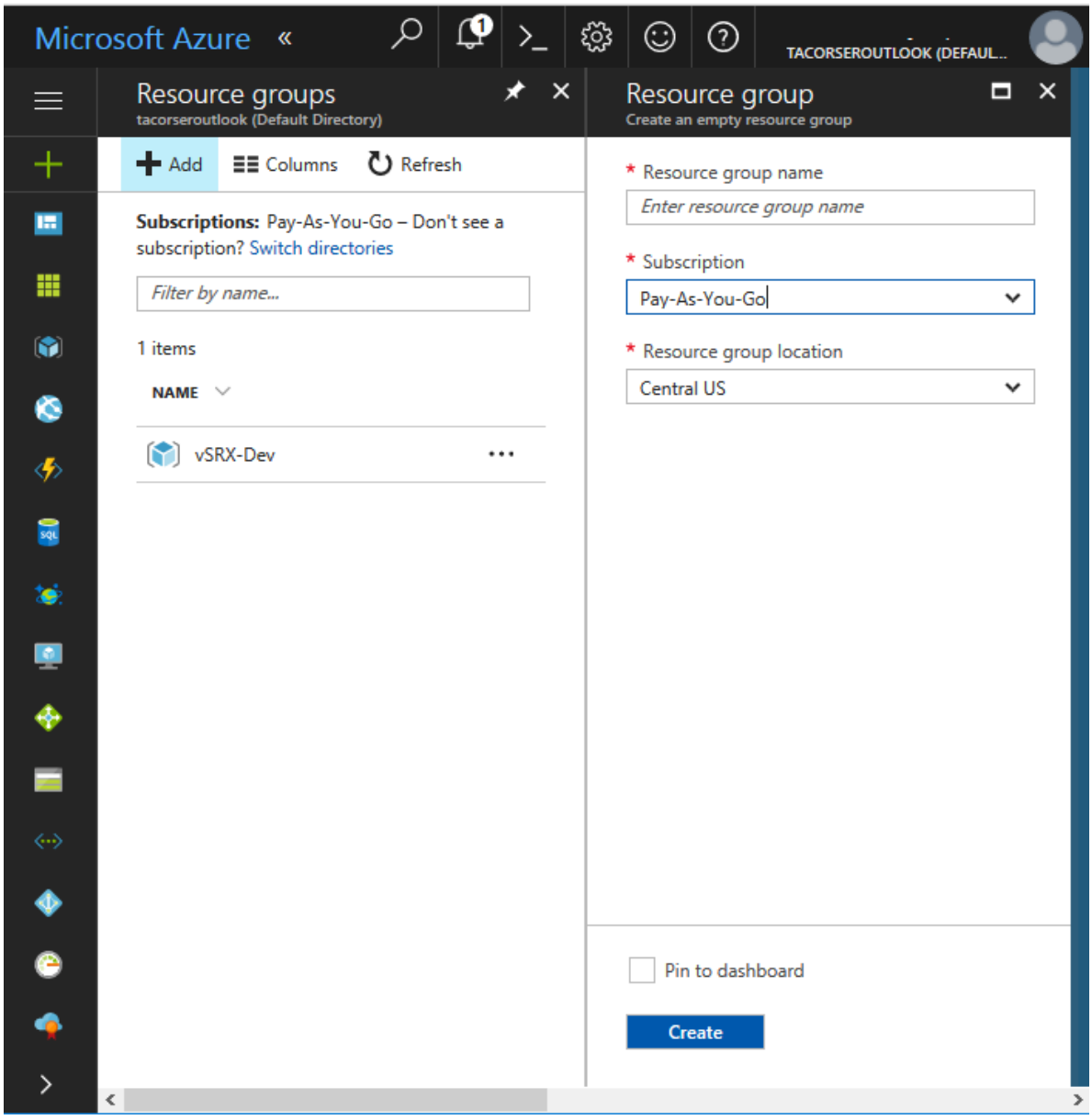
2. Click **Resource groups** from the menu of services to access the Resource Groups blade (see Figure 92 on page 401). You will see all the resource groups in your subscription listed in the blade.

Figure 92: Resource Groups



3. click **Add (+)** to create a new resource group. The Create Resource Group blade appears (see Figure 93 on page 402).

Figure 93: Creating a Resource Group



4. Provide the following information for the new resource group.

Parameter	Description
Resource Group Name	Enter a unique name for your new resource group. A resource group name can include alphanumeric characters, periods (.), underscores (_), hyphens (-), and parenthesis (), but the name cannot end with a period.
Subscription	Select your Microsoft Azure subscription.
Resource Group Location	Select the location of the Microsoft Azure data center from which you intend to deploy the vSRX VM. Specify a location where the majority of your resources will reside. Typically, select the location that is closest to your physical location.

5. Click **Create**. The resource group might take a few seconds to create. Once it is created, you see the resource group on the Azure portal dashboard.

RELATED DOCUMENTATION

- Azure Resource Manager overview
- Deploy resources with Resource Manager templates and Azure portal
- Manage Azure resources through portal

Create a Storage Account

An Azure storage account provides a unique namespace to store and access your Azure storage data objects. All objects in a storage account are billed together as a group. By default, the data in your account is available only to the account owner.

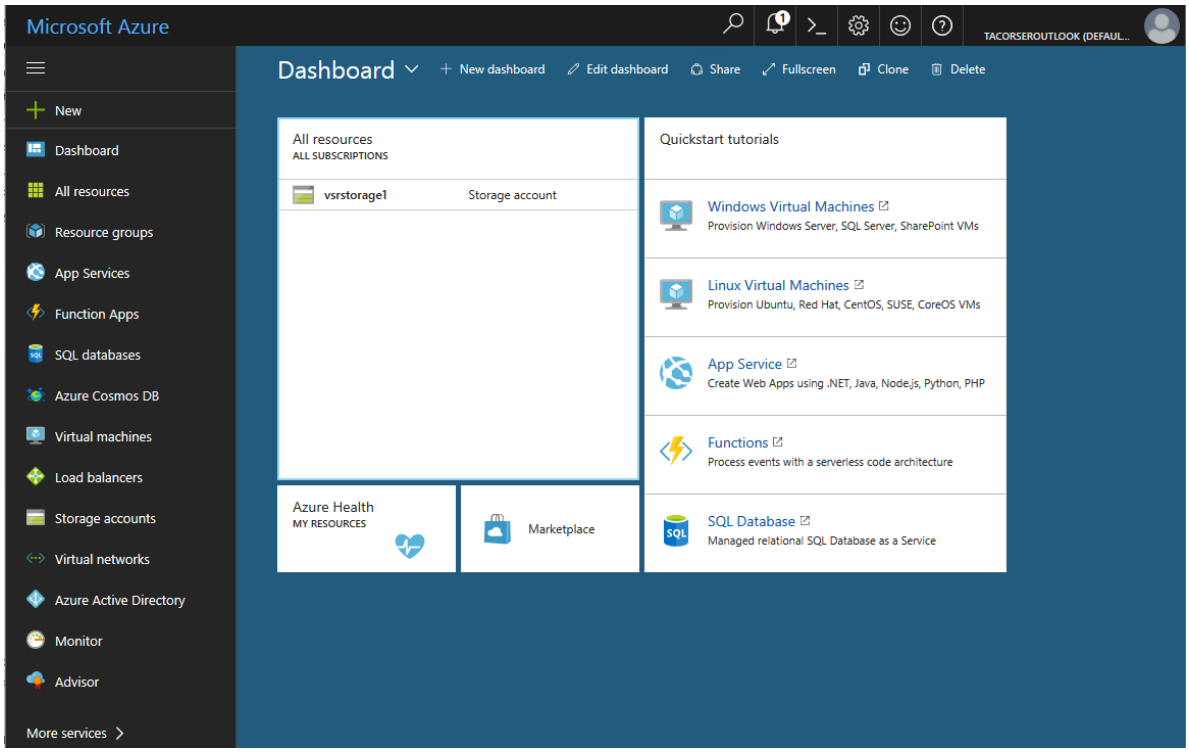
If you do not have an existing storage account in your subscription, follow the steps outlined in this procedure.

To create a storage account in Azure:

1. Log in to the Microsoft Azure portal using your Microsoft account username and password. The Dashboard appears in the Azure portal (see Figure 94 on page 404). You see a unified dashboard for

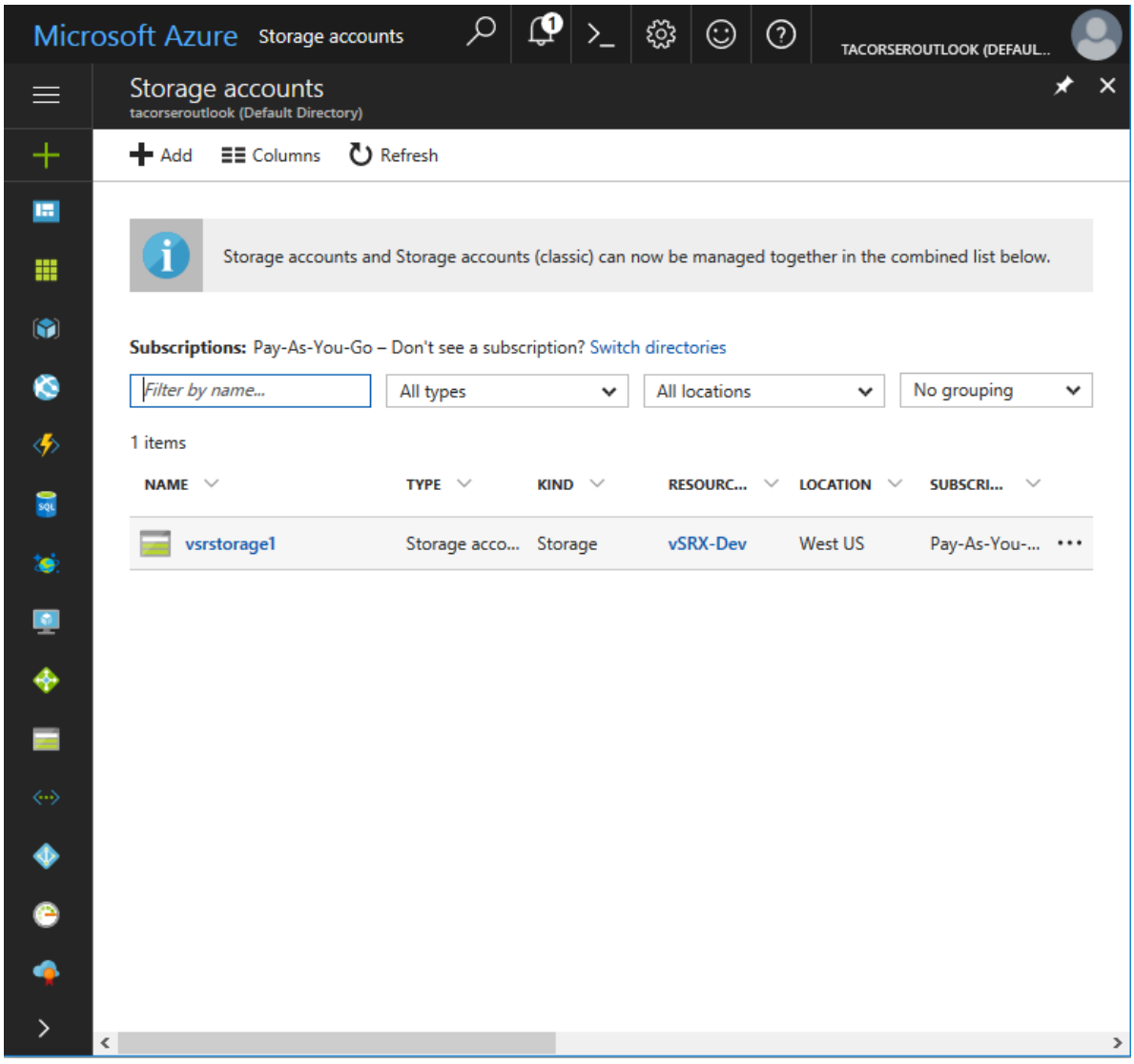
all your assets in Azure. Verify that the dashboard includes all subscriptions to which you currently have access, and all resource groups and associated resources.

Figure 94: Microsoft Azure Portal Dashboard



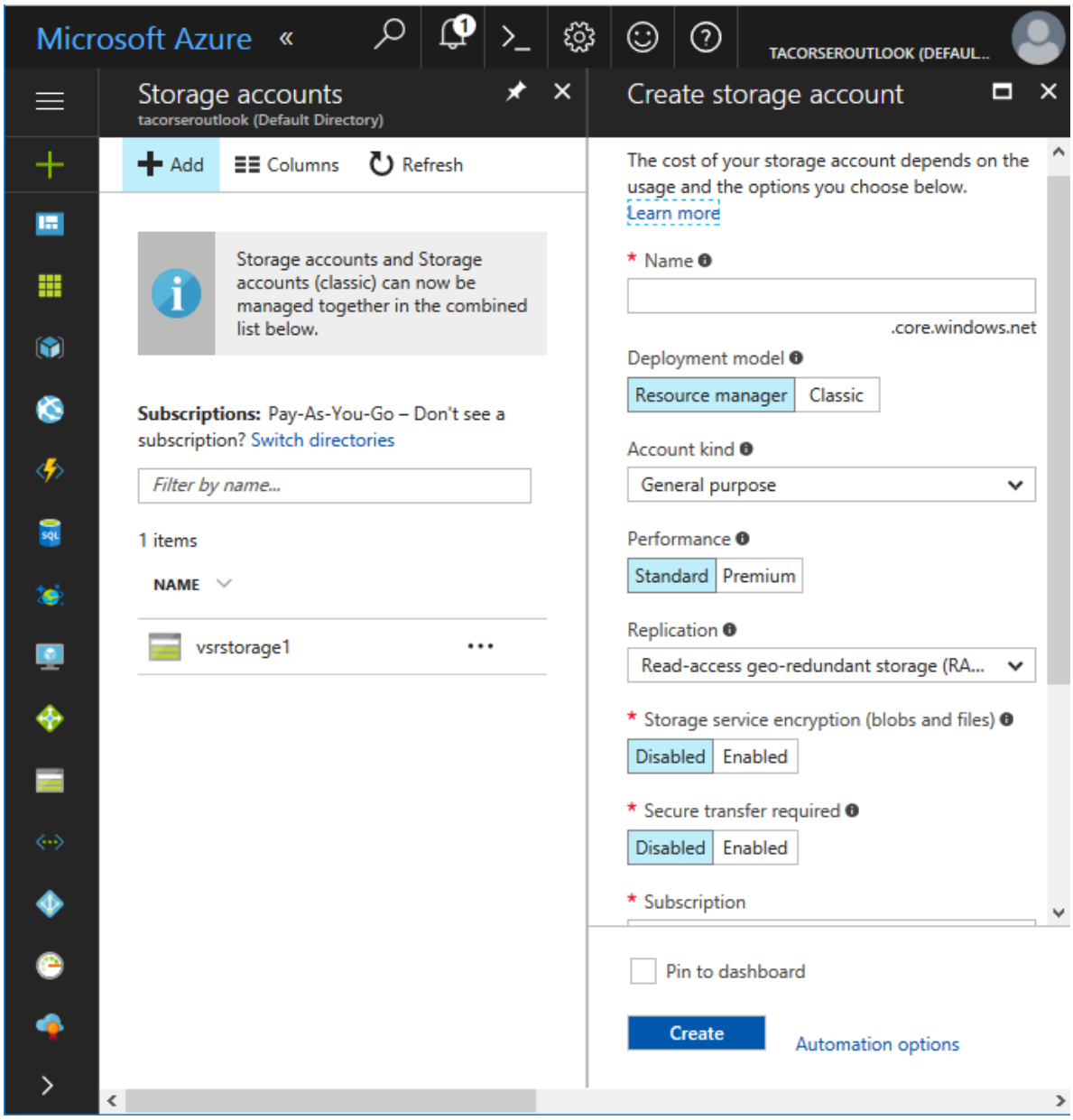
2. Click **Storage Accounts** from the menu of services to access the Storage Accounts blade (see Figure 95 on page 405).

Figure 95: Azure Portal Storage Accounts



3. Click **Add (+)** to create a new storage account. The Create Storage Account blade appears (see Figure 96 on page 406).

Figure 96: Creating a Storage Account



4. Provide the following information for the new storage account.

Parameter	Description
Name	Enter a unique name for your new storage account. A storage account name can contain only lowercase letters and numbers, and must be between 3 and 24 characters.
Deployment Model	Select **Resource Manager** as the deployment model.
Account Kind	Select the type of storage account: **General purpose** or **Blob storage**. The default is **General purpose**. • If General Purpose was selected, then specify the performance tier: **Standard** or **Premium**. The default is Standard. • If Blob storage was selected, then specify the access tier: **Hot** or **Cool**. The default is Hot.
Performance	Select the type of performance: **Standard** or **Premium**. The default is **Standard**.
Replication	Select the replication option for the storage account: **Locally redundant storage (LRS)**, **Geo-redundant storage (GRS)**, **Read-access geo-redundant storage (RA-GRS)**, or **Zone-redundant storage (ZRS)**. The default is RA-GRS.
Storage Service Encryption	Enable or disable this option to protect your data at rest. Azure Storage encrypts data as written in an Azure datacenter, and decrypts that data once it is accessed. The default is Disabled.
Secure Transfer Required	Enable or disable this option to enhance the security of your storage account by allowing requests to the storage account by HTTPS only. The default is Disabled.
Subscription	Select your Microsoft Azure subscription.
Resource Group	Select an existing resource group or create a new one (see *Create a Resource Group*).

(Continued)

Parameter	Description
Location	Select the Azure data center geographic region in which you are deploying the vSRX VM. Typically, select the location that is closest to your physical location.

5. Click **Create**. The storage account might take a few seconds to create. Once it is created, you see the storage account on the Azure portal dashboard.

RELATED DOCUMENTATION

- Introduction to Microsoft Azure Storage
- About Azure storage accounts

Create a Virtual Network

The Azure Virtual Network service enables you to securely connect Azure resources to each other with virtual networks. A virtual network is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can also connect virtual networks to your on-premises network.

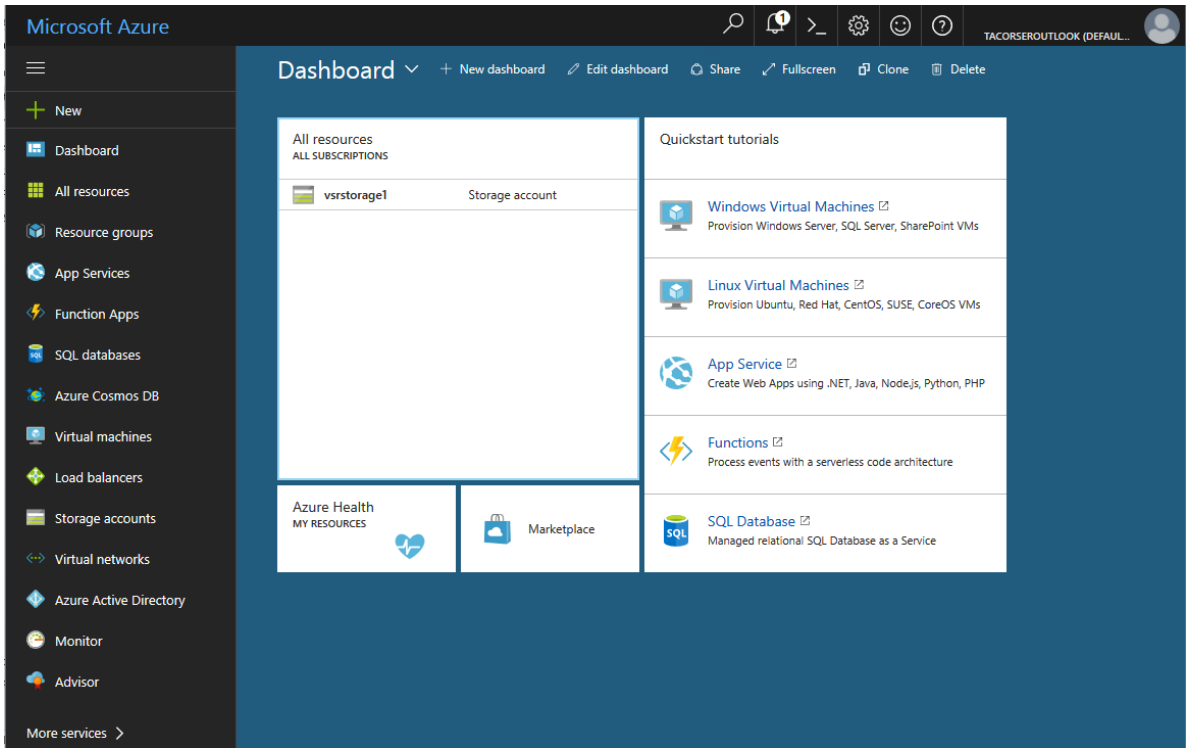
If you do not have an existing Azure virtual network, follow the steps outlined in this procedure.

To create an Azure virtual network:

1. Log in to the Microsoft Azure portal using your Microsoft account user name and password. The Dashboard appears in the Azure portal (see Figure 97 on page 409). You will see a unified dashboard

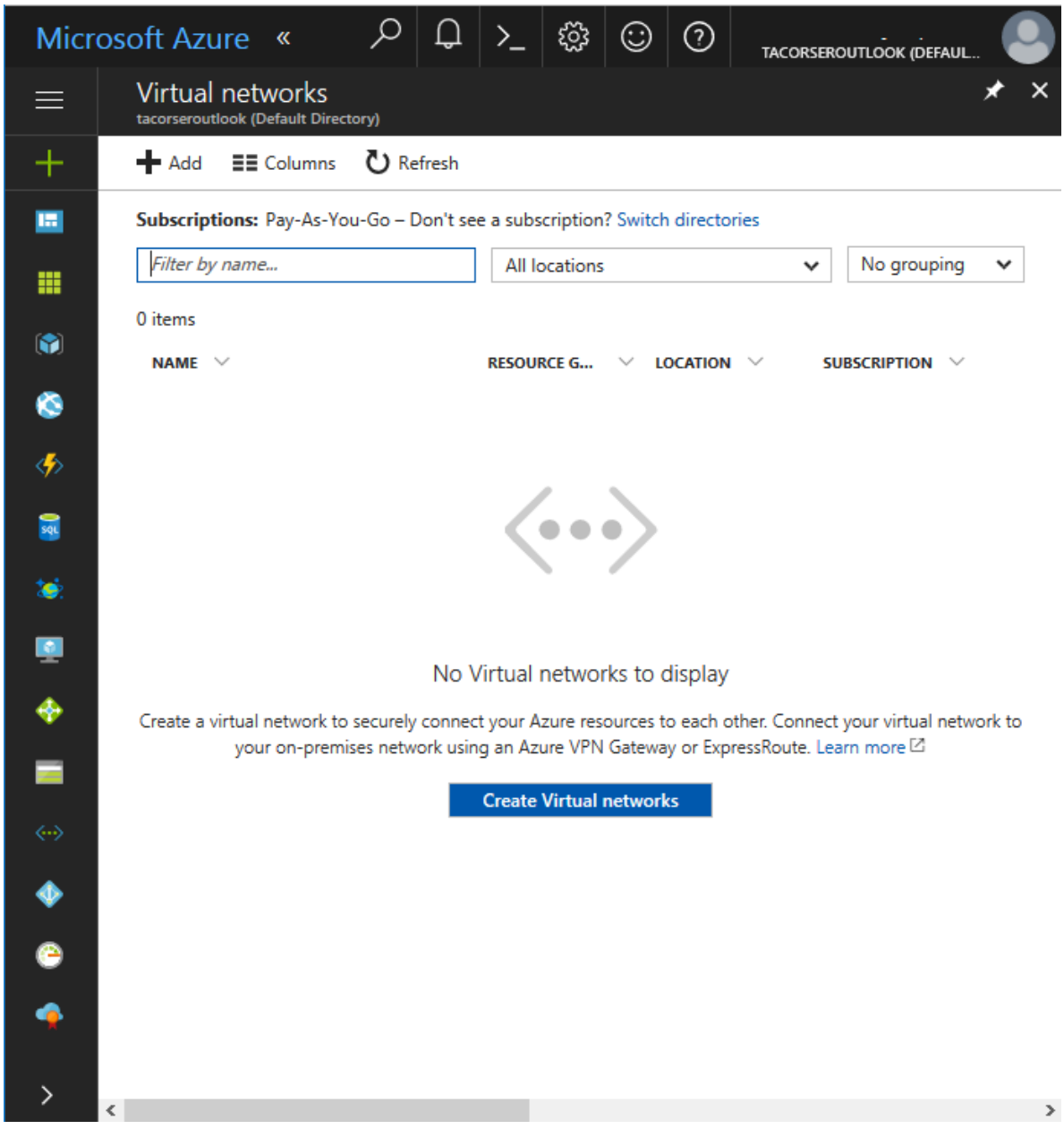
for all your assets in Azure. Verify that the dashboard includes all subscriptions to which you currently have access, and all resource groups and associated resources.

Figure 97: Microsoft Azure Portal Dashboard



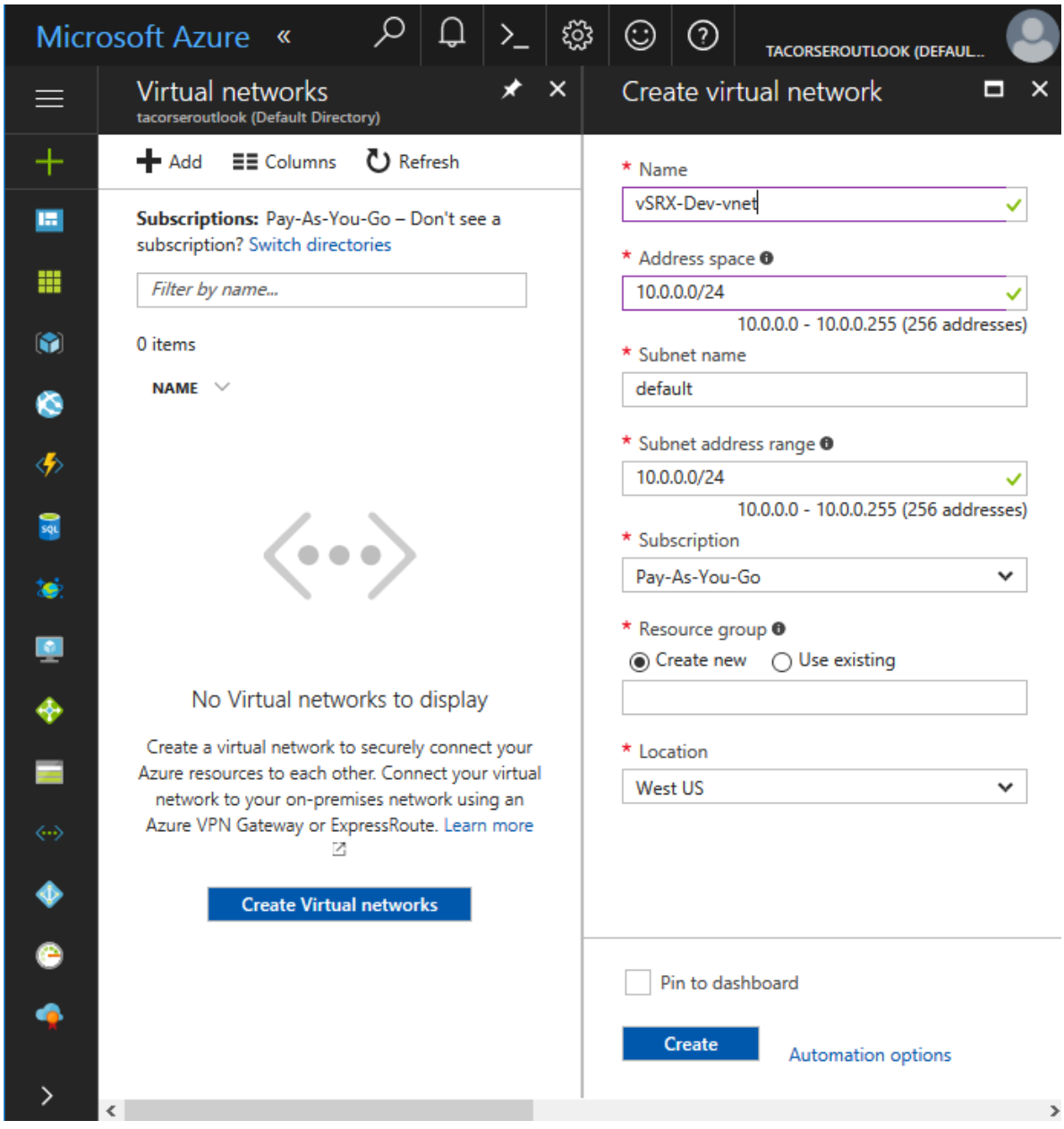
2. Click **Virtual Networks** from the menu of services to access the Virtual Networks blade (see Figure 98 on page 410).

Figure 98: Azure Portal Virtual Networks



3. Click **Add (+)** to create a new virtual network. The Create Virtual Network blade appears (see Figure 99 on page 411).

Figure 99: Creating a Virtual Network



4. Provide the following information for the new virtual network.

Parameter	Description
Name	Enter a unique name for your new virtual network. The virtual network name must begin with a letter or number, end with a letter, number, or underscore, and the name may contain only letters, numbers, underscore, periods, or hyphens.
Address Space	Enter the virtual network's address range in CIDR notation. By default, the address range is 10.0.0.0/24. **NOTE**: Ensure that the address space does not overlap with an existing network.
Subnet name	Enter a unique name for the subnet of the Azure virtual network. The subnet name must begin with a letter or number, end with a letter, number, or underscore, and the name may contain only letters, numbers, underscore, periods, or hyphens.
Subnet Address Range	Enter a network subnet address range in CIDR notation. It must be contained by the address space of the virtual network, as defined in the Address Space field. Subnet address ranges cannot overlap one another. By default, the address range is 10.0.0.0/24. The subnet is a range of IP addresses in your virtual network to isolate VMs. Public subnets have access to the Internet gateway, but private subnets do not. **NOTE**: The address range of a subnet that is already in use cannot be edited.
Subscription	Select your Microsoft Azure subscription.
Resource Group	Select an existing resource group or create a new one (see *Create a Resource Group*).
Location	Select the Azure data center geographic region in which you are deploying the vSRX VM. Typically, select the location that is closest to your physical location.

5. Click **Create**. The virtual network might take a few seconds to create. Once it is created, you will see the virtual network on the Azure portal dashboard.

RELATED DOCUMENTATION

- Virtual networks and Windows virtual machines in Azure
- Create a virtual network
- Create, change, or delete network interfaces
- Create a VM (Classic) with multiple NICs

Deploy the vSRX Image from Azure Marketplace

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D91 for vSRX, you can deploy the vSRX virtual security appliance in your Azure virtual network by selecting the vSRX image from Azure Marketplace and customizing the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud.

This deployment approach might be needed if you have a vSRX VM deployment scenario that is outside of the use cases offered in the vSRX VM solution templates available from Juniper Networks.

NOTE: Be sure you have an account for and a subscription to Microsoft Azure before deploying the vSRX to Azure (see Microsoft Azure).

If you do not have an Azure subscription, then you can create a free account before you begin. See the Microsoft Azure website for more details.

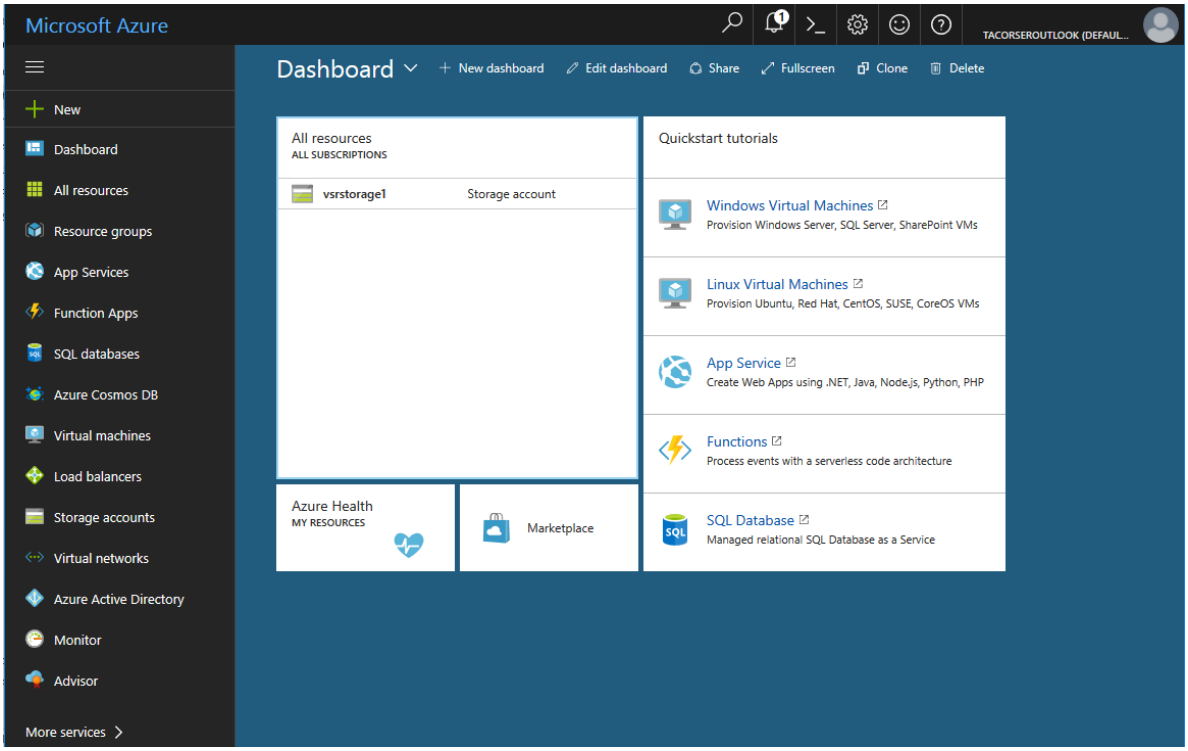
Use the following procedures to deploy and configure a vSRX VM into an Azure virtual network from the Azure portal.

Deploy the vSRX Image

To deploy and configure a vSRX VM into an Azure virtual network using the vSRX image from Azure Marketplace:

1. Log in to the Microsoft Azure portal using your Microsoft account user name and password. The Dashboard appears in the Azure portal (see Figure 100 on page 414). You will see a unified dashboard for all your assets in Azure. Verify that the dashboard includes all subscriptions to which you currently have access, and all resource groups and associated resources.

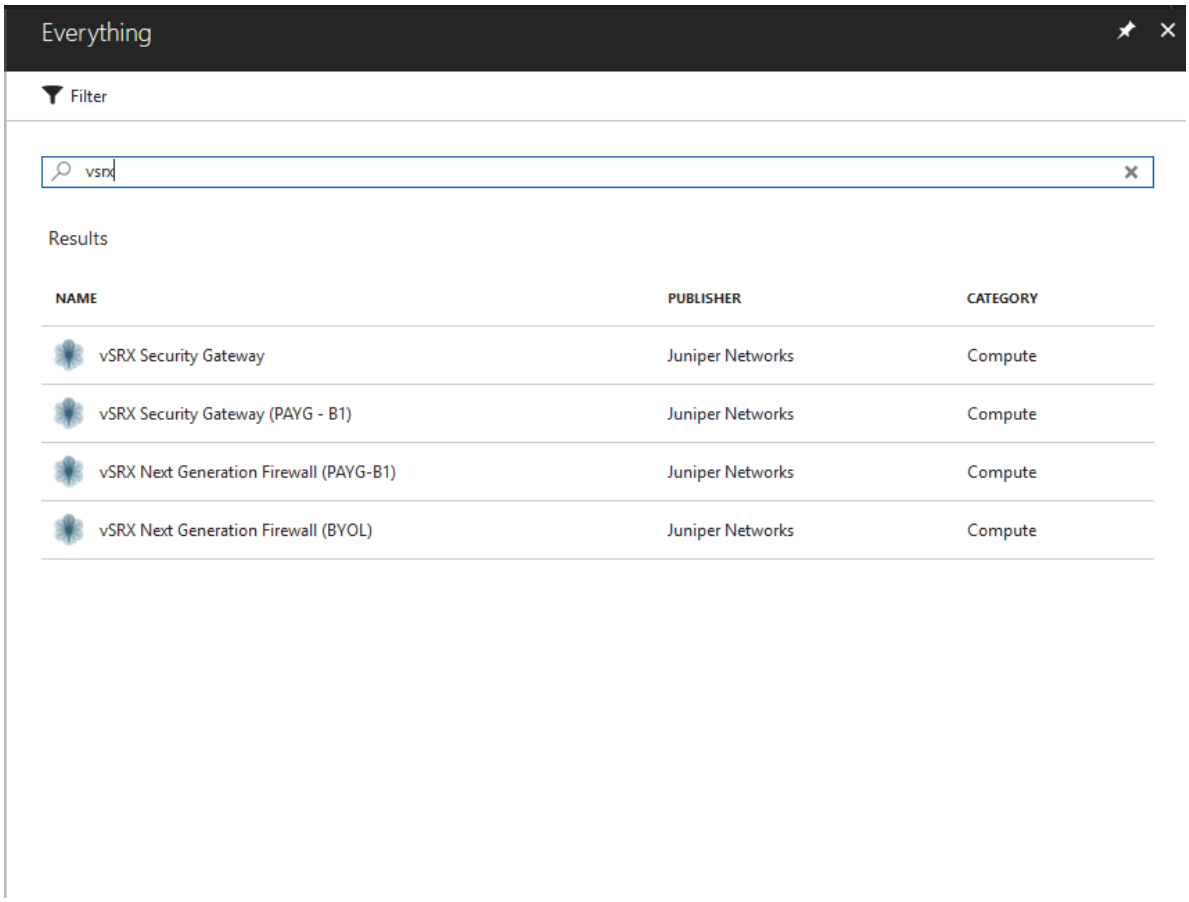
Figure 100: Microsoft Azure Portal Dashboard



2. Click **Marketplace** from the dashboard to access the Azure Marketplace, and then click **Everything** (or click **New > Everything**). Enter **vsrx** to search for the available Juniper Networks vSRX VM images in

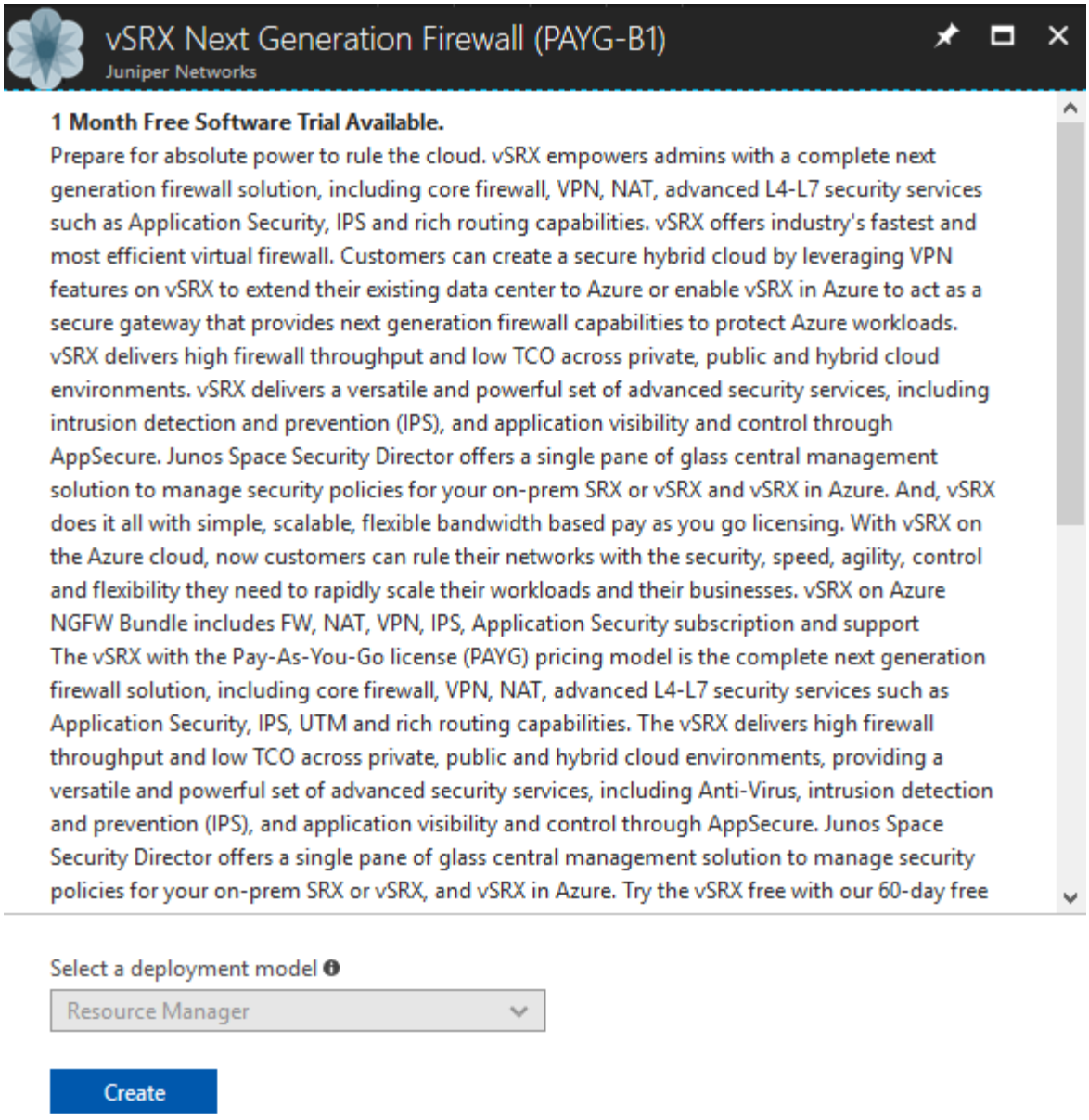
the Azure Marketplace (see Figure 101 on page 415). The vSRX image is available as a pay-as-you-go (PAYG) or bring-your-own-license (BYOL) service.

Figure 101: Locating the vSRX VM Image in the Azure Marketplace



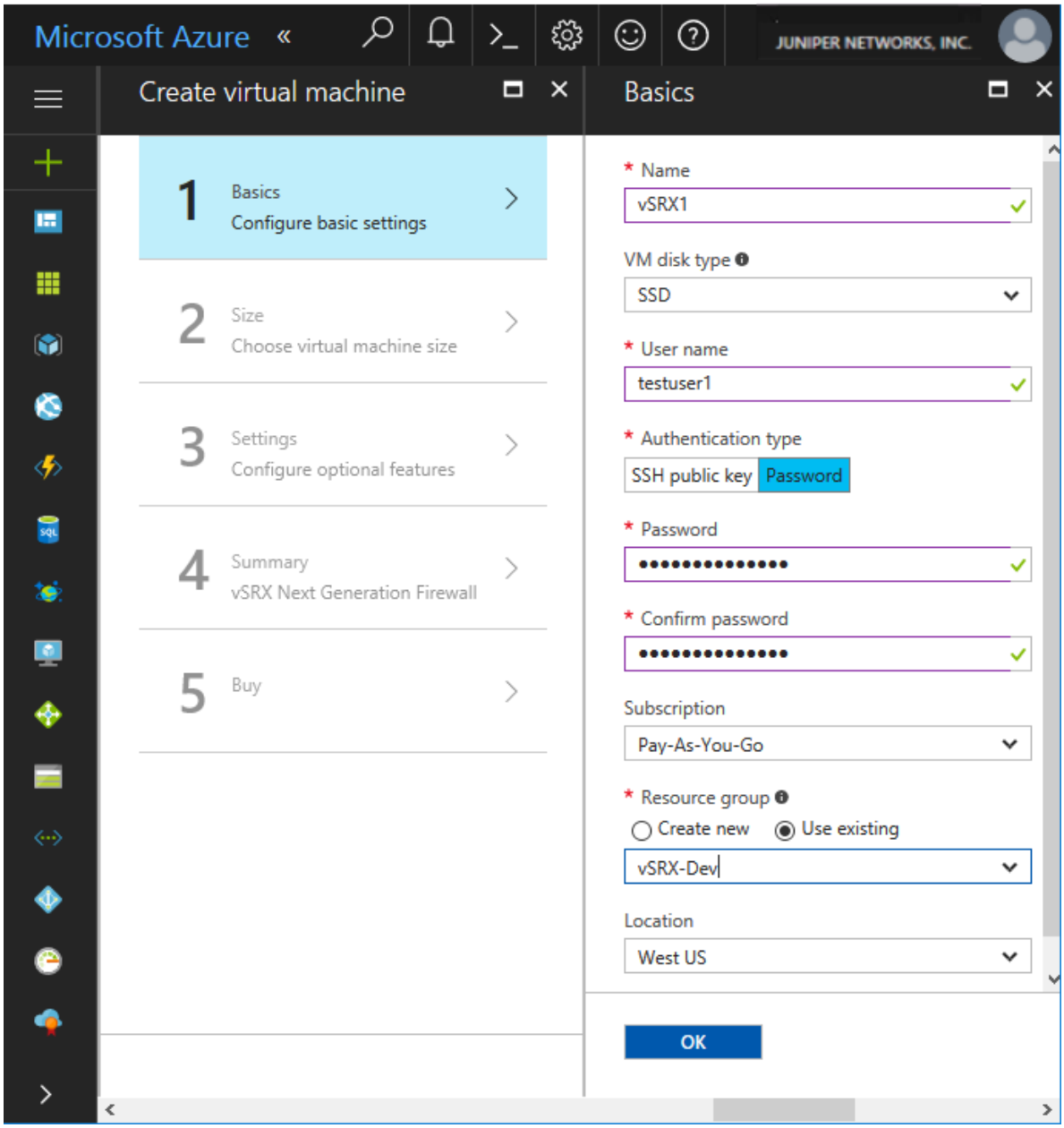
3. Select the vSRX VM image from the list and then click **Create** to initiate the vSRX VM deployment process (see Figure 102 on page 416).

Figure 102: Initiating vSRX VM Deployment



4. From the Create Virtual Machine blade, **1 Basics**, configure the following parameters (see Figure 103 on page 417).

Figure 103: Create Virtual Machine - Basics



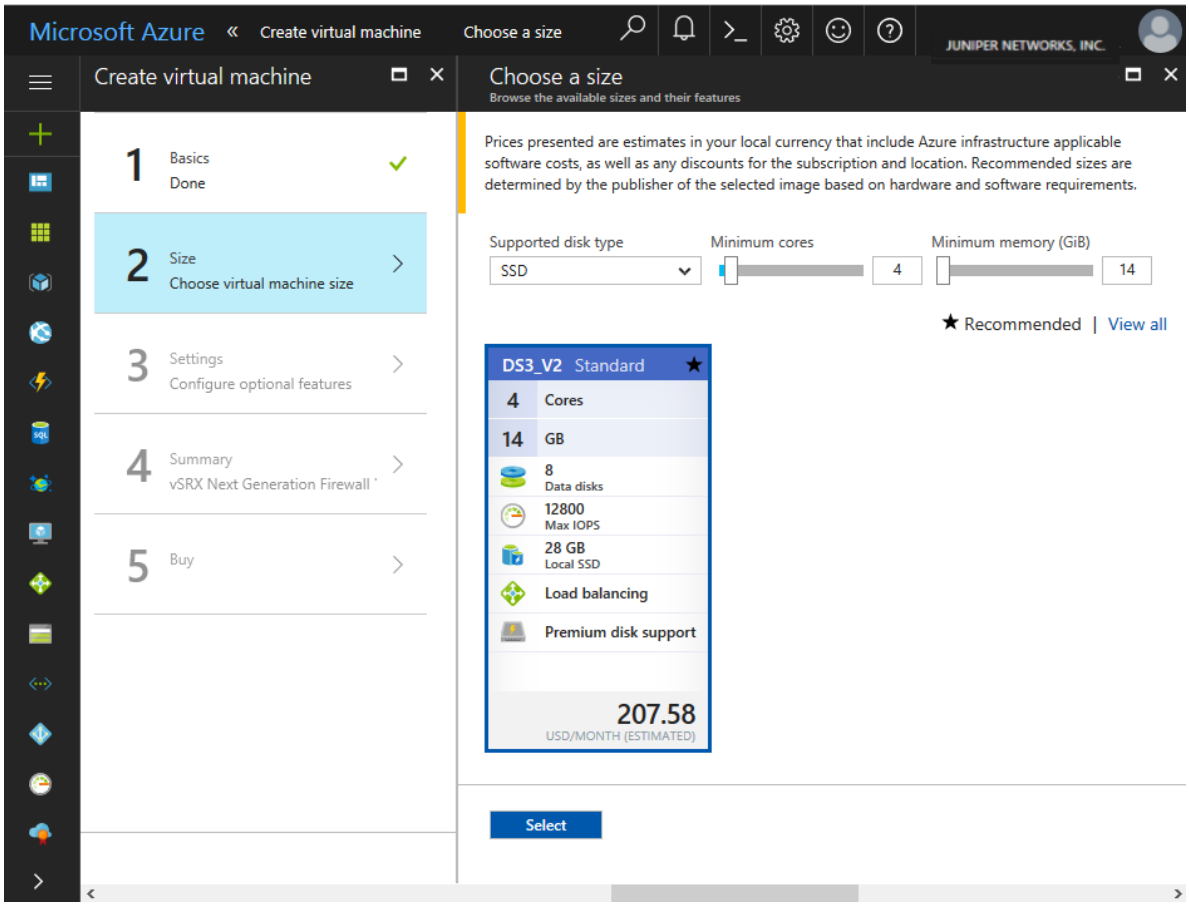
Parameter	Description
Name	Specify a name for your vSRX VM. Your vSRX VM name cannot contain non-ASCII or special characters.
VM Disk Type	Specify the disk type to use for the vSRX VM: **SSD** or **HDD**. The default is **SSD**.
User name	Enter a username to access the vSRX VM. The username cannot contain uppercase characters, special characters, or start with a “$” or “-” character.
Authentication type	Select the required method of authentication to access the vSRX VM: **Password** or **SSH public key**. Select Password as type of authentication and then enter (and confirm) your password. **NOTE**: In Junos OS Release 15.1X49-D91 for vSRX, SSH public key is not a supported authentication method. You will need to specify a password to log in to the vSRX VM. Starting in Junos OS Release 15.1X49-D110 for vSRX, SSH public key is a supported authentication method.
Password	Enter an appropriate root password used to access the vSRX VM.
Subscription	Select your Microsoft Azure subscription.
Resource Group	Select an existing resource group or create a new one (see *Create a Resource Group*).
Location	Select the Azure geographic region in which you are deploying the vSRX VM.

Click **OK**.

5. From the Create Virtual Machine blade, **2 Size**, select **DS3_v2 Standard** as the vSRX VM size (see Figure 104 on page 419). Click **Select**.

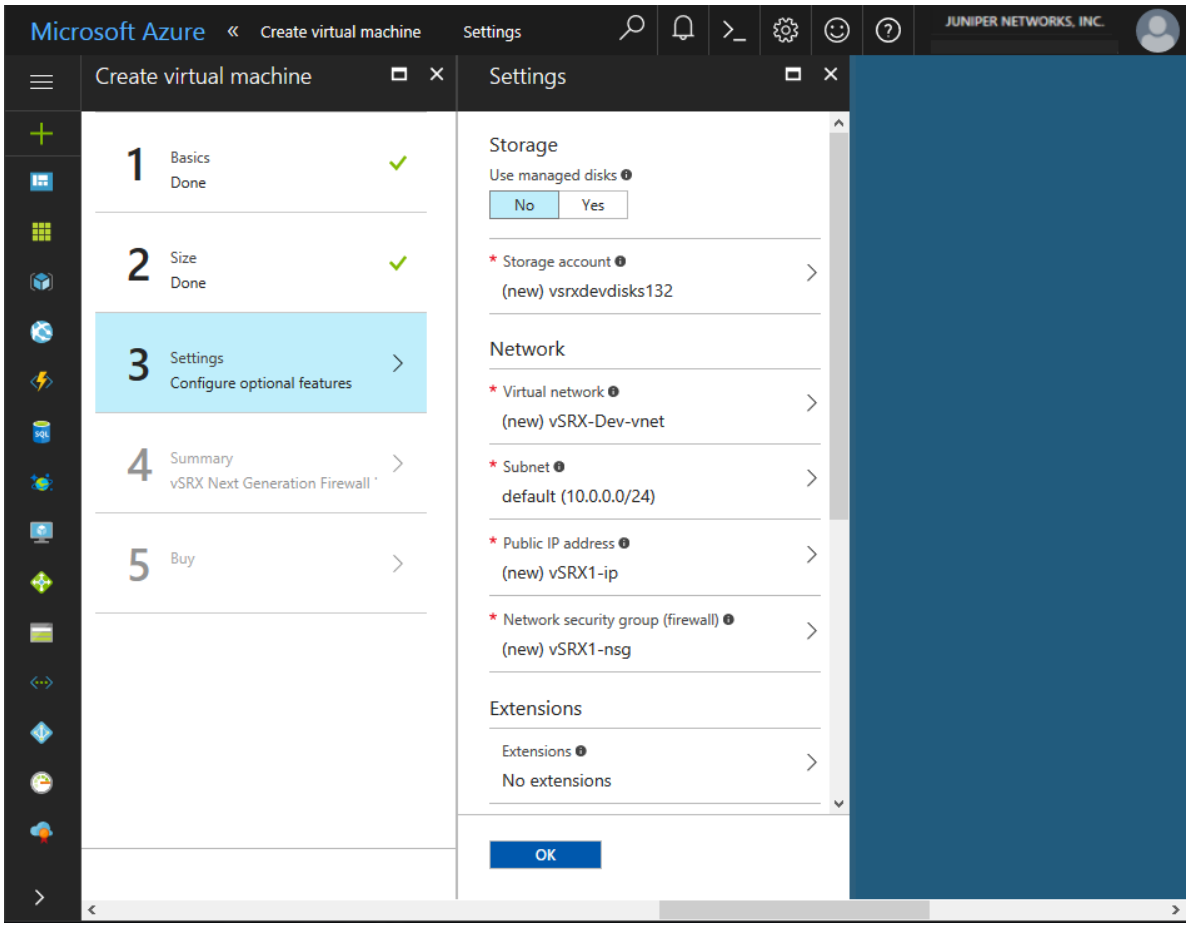
DS3_v2 Standard is used for a vSRX VM deployment. See *Requirements for vSRX on Microsoft Azure* for the recommended system requirements for a vSRX instance in Microsoft Azure.

Figure 104: Create Virtual Machine - Choose a Size



6. From the Create Virtual Machine blade, **3 Settings**, configure the following parameters to define the storage, networking, and monitoring settings for the vSRX VM (see Figure 105 on page 420). Click **OK** when completed.

Figure 105: Create Virtual Machine - Settings



Parameter	Description
Storage	
Used Managed Disks	Specify whether you want Azure to automatically manage the availability of disks to provide data redundancy and fault tolerance without you creating and managing a storage account. Click **No**.

(Continued)

Parameter	Description
Storage Account	If you need to change the storage account for the vSRX VM, click the right arrow to access the Choose Storage Account blade. Select an existing storage account for the vSRX VM, or click **Create new (+)** to create a new one. See *Create a Storage Account* for details about creating a new storage account.
Network	
Virtual Network	If you need to change the virtual network for the vSRX VM, click the right arrow to access the Choose Virtual Network blade. Select an existing virtual network for the vSRX VM, or click **Create new (+)** to create a new one. See *Create a Virtual Network* for details about creating a new virtual network.
Subnet	Enter a subnet, which is a range of IP addresses in your virtual network to isolate VMs. Public subnets have access to the Internet gateway, but private subnets do not. A vSRX VM requires two public subnets and one or more private subnets for each individual instance group. The public subnets consist of one for the management interface (fxp0) and another for the two revenue (data) interfaces. The private subnets, connected to other vSRX interfaces, ensure that all traffic between applications on the private subnets and the Internet must pass through the vSRX instance. To modify the subset for the virtual network, click the right arrow to access the Create Subnet blade. Configure the following parameters: • Subnet name—A unique name for the subnet in the Azure virtual network. • Subnet address range—The subnet's address range in CIDR notation. It must be contained by the address space of the virtual network. Subnet address ranges cannot overlap one another. By default, the address range is 10.0.0.0/24. **NOTE**: The address range of a subnet that is already in use cannot be edited.

(Continued)

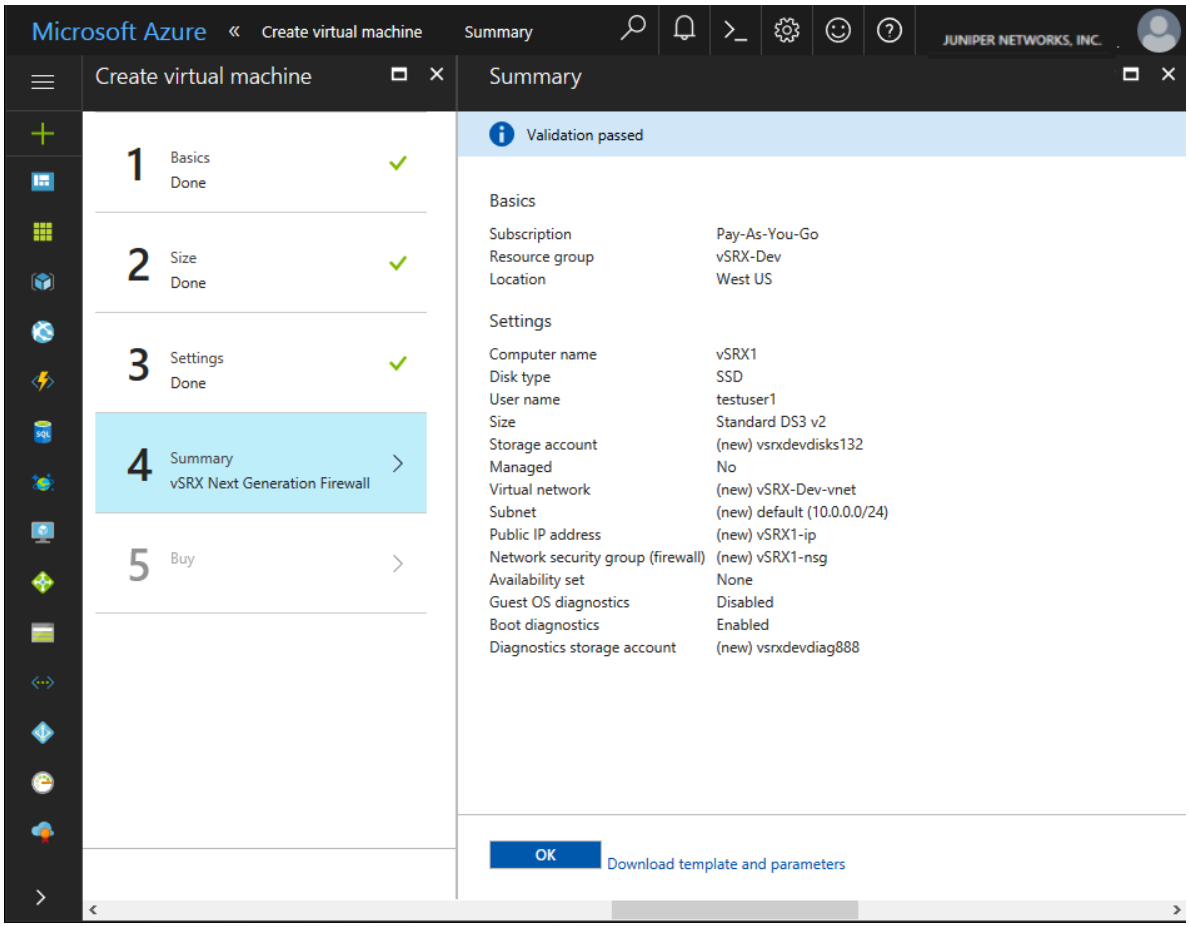
Parameter	Description
Public IP address	Specify the public IP address that allows communication to the vSRX VM from outside the Azure virtual network. To modify the public IP address for the vSRX VM, click the right arrow to access the Choose Public IP Address blade. Select a public IP address in your Azure subscription and location, or click **Create new (+)** to create a new one. Configure the following parameters: • Name—A unique name for the public IP address. • Assignment—There are two methods in which an IP address is allocated to a public IP resource: dynamic or static. By default, public IP addresses are dynamic, where an IP address is not allocated at the time of its creation. Instead, the public IP address is allocated when you start (or create) the resource. The IP address associated to them may change when the vSRX VM is deleted. To guarantee that the vSRX VM always uses the same public IP address, we recommend you assign a static public IP address.
Network security group	Specify a network security group, which is a set of firewall rules that control traffic to and from the vSRX VM. Each network security group can contain multiple inbound and outbound security rules that enable you to filter traffic by source and destination IP address, port, and protocol. You can apply a network security group to each NIC in the VM. To modify the network security group for the vSRX VM to filter traffic, click the right arrow to access the Choose Network Security blade. Select a network security group in your Azure subscription and location, or click **Create new (+)** to create a new one. Configure the following parameters: • Name—A unique name for the network security group. • Inbound rules—You can add one or more inbound security rules to allow or deny traffic to the vSRX VM. • Outbound rules—You can add one or more outbound security rules to allow or deny traffic originating from the vSRX VM.

(Continued)

Parameter	Description
Extensions	
Extensions	No extensions are used for the vSRX VM.
High Availability	
Availability Set	Confiigure two or more VMs in an availability set to provide redundancy to an application. **NOTE**: Availability Set should be set to **None** for the vSRX VM. Availablilty Set is not used for the vSRX VM in Azure because chassis clustering is not supported by the vSRX at this time.
Monitoring	
Boot Diagnostics	Enables or disables the capturing of serial console output and screenshots of the VM running on the host to help diagnose start-up issues. The default is Enabled.
Guest OS Diagnostics	Enables or disables the ability to obtain metrics every minute for the VM. Choices are: **Disabled** or **Enabled**. The default is Disabled.
Diagnostics Storage Account	Click the right arrow to view the details of the diagnostics storage account. Automatically fills in with the name of the diagnostics storage account from which you can analyze a set of metrics with your own tools.

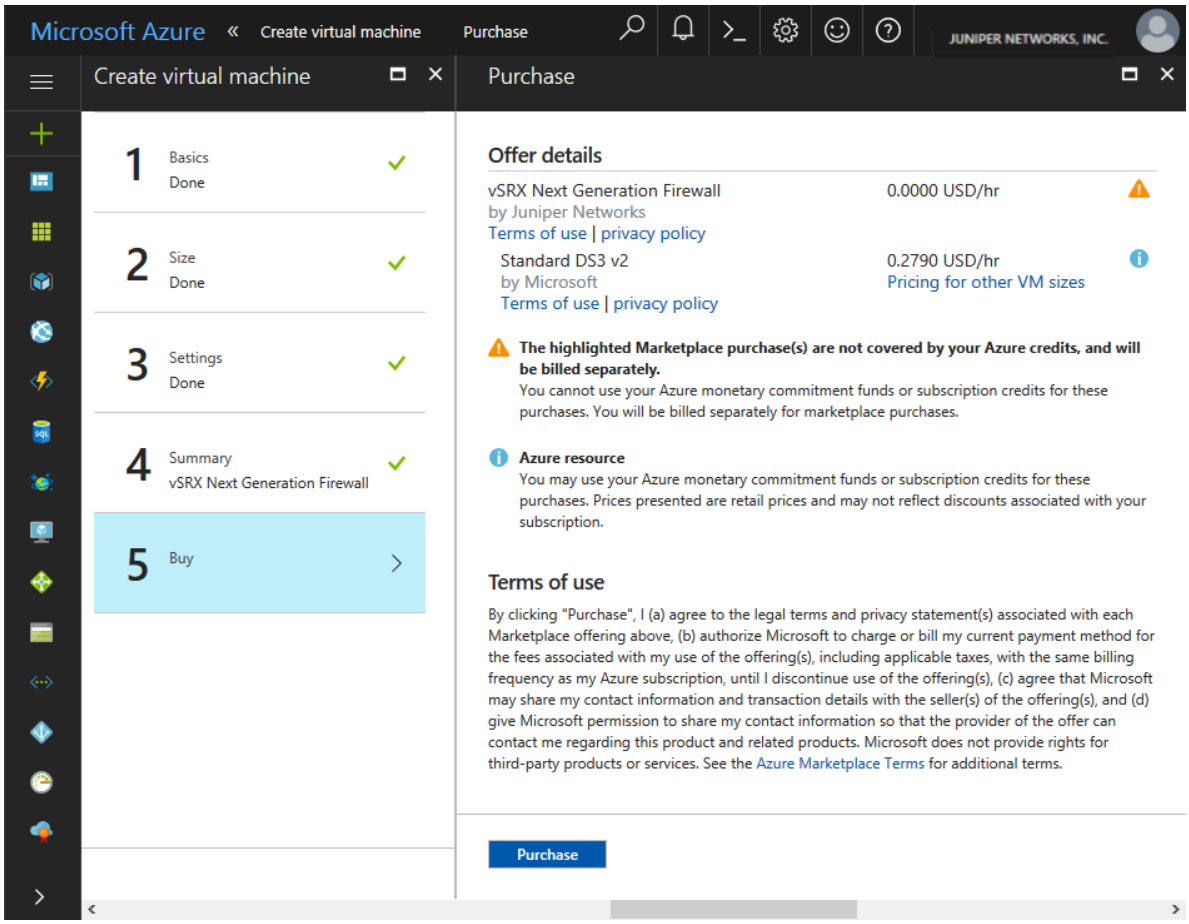
7. From the Create Virtual Machine blade, **4 Summary** , review the configuration settings (see Figure 106 on page 424). If you are satisfied with the configuration settings, click **OK**.

Figure 106: Create Virtual Machine - Summary



8. From the Create Virtual Machine blade, **5 Buy** review the offer details and the terms of use (see Figure 107 on page 425). If you are satisfied with the offer details and terms of use, click **Purchase**.

Figure 107: Create Virtual Machine - Purchase



You return to the Azure portal dashboard, and the dashboard displays the deployment status of the vSRX VM.

Verify Deployment of vSRX to Microsoft Azure

After the vSRX VM is created, the Azure portal dashboard lists the new vSRX VM under Resource Groups. The corresponding cloud service and storage account also are created and listed. Both the vSRX VM and the cloud service are started automatically and their status is listed as `Running`

To verify the deployment of the vSRX instance to Microsoft Azure:

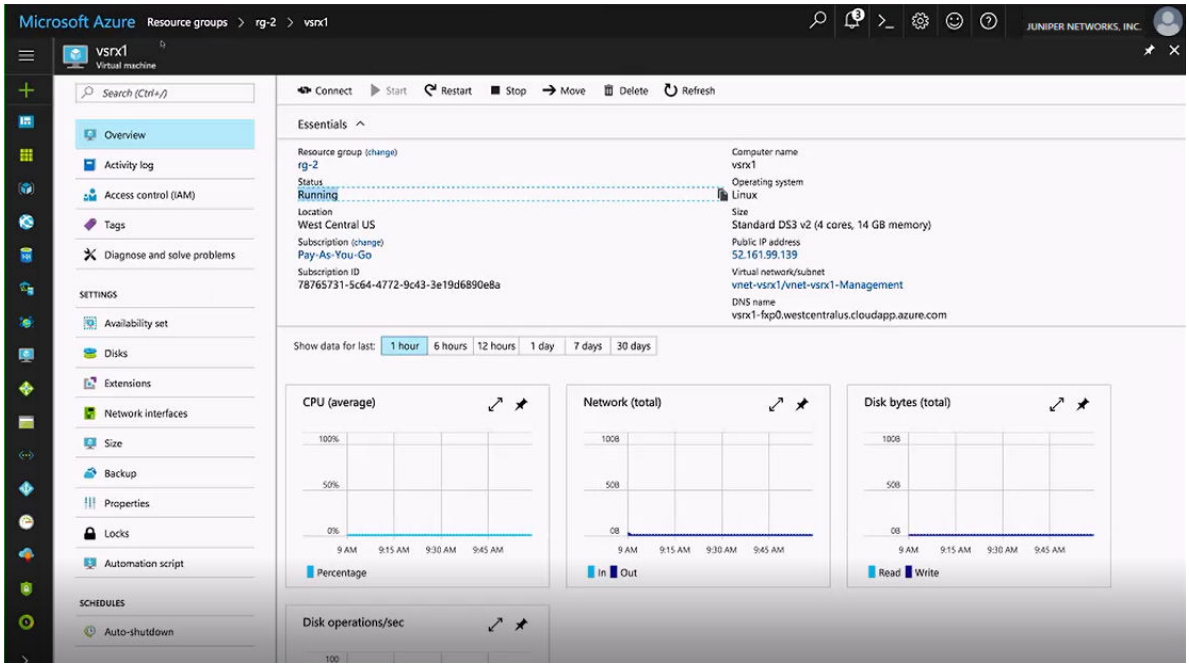
1. To view the vSRX resource group and its resources after deployment is completed, from the right-hand menu, click **Resource groups** to access the Resource Groups page.

2. To view details of the vSRX VM associated with the resource group, click the name of the vSRX VM. Observe that the status is `Running`.

> **NOTE**: You can stop, start, restart, and delete a vSRX VM from the Virtual Machine page in the Microsoft Azure portal.

Figure 108 on page 426 shows an example of a Resource groups vSRX VM in the Microsoft Azure portal.

Figure 108: Microsoft Azure Resource Groups VM Example



Log In to a vSRX VM

After vSRX deployment is completed, the vSRX VM is automatically powered on and launched. At this point you can use an SSH client to log in to the vSRX VM.

> **NOTE**: In Microsoft Azure, individuals and enterprises can host servers and services on the cloud as a pay-as-you-go (PAYG) or bring-your-own-license (BYOL) service. For the vSRX on Microsoft Azure deployment, only the BYOL model is supported.

To log in to the vSRX VM:

1. From the Azure portal, click **Resource groups** from the menu of services on the dashboard, and then select the vSRX VM. Locate the public IP address of the vSRX VM from the Settings blade.
2. Use an SSH client to log in to a vSRX VM.
3. At the prompt, enter the following login credentials:

   NOTE: The vSRX instance is automatically configured for username and password authentication. To log in, use the login credentials that were defined during the vSRX VM configuration (see "Deploy the vSRX Image" on page 413). After initially logging in to the vSRX, you can configure SSH public and private key authentication.

   ```
   # ssh <username@vsrx_vm_ipaddress>
   ```

   ```
   The authenticity of host 'x.x.x.x (x.x.x.x)' ...
   ECDSA key fingerprint is SHA256:XXXXXXXXXXXXXXXXXXXXXXX.
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added 'x.x.x.x' (ECDSA) to the list of known hosts.
   Password: xxxxxxxx
   username@vsrx_vm_ipaddress>
   ```

4. Configure the basic settings for the vSRX VM (see *Configure vSRX Using the CLI*).

Release History Table

Release	Description
15.1X49-D91	Starting in Junos OS Release 15.1X49-D91 for vSRX, you can deploy the vSRX virtual security appliance in your Azure virtual network by selecting the vSRX image from Azure Marketplace and customizing the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud.

RELATED DOCUMENTATION

- How to Deploy in Microsoft Azure using Azure Portal and Template
- Microsoft Azure portal overview
- Virtual networks and Windows virtual machines in Azure
- Create, change, or delete network interfaces
- Create a VM (Classic) with multiple NICs

CHAPTER 22

Deploy vSRX from the Azure CLI

IN THIS CHAPTER

Before You Deploy vSRX Using the Azure CLI

Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX from the Azure CLI and customize the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud.

To help automate and simplify the deployment of the vSRX in the Microsoft Azure virtual network, Juniper Networks provides a series of scripts, Azure Resource Manager (ARM) templates and parameter files, and configuration files in the GitHub repository https://github.com/Juniper/vSRX-Azure. The ARM template includes resource parameters that enable you to customize your vSRX VM deployment, such as login credentials, network interfaces, and storage container name. The template consists of JavaScript Object Notation (JSON) expressions for your vSRX deployment.

The vSRX deployment files in the GitHub repository include:

- The **deploy-azure-vsrx.sh** shell script to automate the deployment and configuration of the vSRX virtual machine (VM).
- The **vsrx.json** template file to define the components of the Azure resource group and virtual hardware settings (VM size, interface number and network) of the vSRX VM.
- The **vsrx.parameters.json** parameter file to identify the network interface parameters used to deploy the vSRX VMin Azure.

Before you deploy the vSRX virtual security appliance from the Azure CLI:

- Review the requirements for deploying a vSRX VM in Microsoft Azure Cloud in *Requirements for vSRX on Microsoft Azure*.
- Obtain an account for and a subscription to Microsoft Azure (see Microsoft Azure).

- From the Azure portal, you must first manually deploy the vSRX image (only once) by using either the **vSRX Next Generation Firewall (BYOL)** or the **vSRX Next Generation Firewall (PAYG)** SKU to accept the EULA terms. This is a requirement before you can deploy the vSRX image from the Azure CLI. By default, the Azure portal deployment tool uses **vSRX Next Generation Firewall (BYOL)** SKU as the source image. Use your Microsoft account username and password to log into the Microsoft Azure portal.

  NOTE: You will encounter a **MarketplacePurchaseEligibilityFailed** error if do not first accept the EULA terms for the vSRX image in the Azure portal before attempting to deploy the vSRX image from the Azure CLI.

- Install Azure command line interface (Azure CLI) 1.0 and enable Azure Resource Management (ARM) mode (see Install the Azure CLI).

  NOTE: The vSRX for Azure deployment shell script **deploy-azure-vsrx.sh** is written in shell and Azure CLI version 1.0 commands and does not support Azure CLI version 2.0.

- Purchase a vSRX license or request an evaluation license. Licenses can be procured from the Juniper Networks License Management System (LMS).

NOTE: Deployment of vSRX to Microsoft Azure does not support the use of the Azure CLI from Microsoft Windows. This is because the deploy-azure-vsrx.sh shell script that is used as part of the deployment procedure can be run only from the Linux or Mac OS CLI.

When you deploy a vSRX VM in an Azure virtual network, note the following specifics of the deployment configuration:

- Use your Microsoft account username and password to log into the Microsoft Azure portal.
- Ensure that your Azure subscription includes the following for your vSRX VM:
  - Resource group, as described in *Create a Resource Group*.
  - Storage account, as described in *Create a Storage Account*.
  - Virtual network, as described in *Create a Virtual Network*.

vSRX deployment from the Azure CLI is described in detail in *Deploy vSRX from the Azure CLI*.

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX from the Azure CLI and customize the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud.

RELATED DOCUMENTATION

- Azure Resource Manager overview
- Deploy resources with Resource Manager templates and Azure CLI

Deploy vSRX from the Azure CLI

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX from the Azure CLI and customize the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud.

Use the following procedure to deploy and configure vSRX as a virtual security appliance in a Microsoft Azure virtual network from the Azure CLI. In this procedure, you use the Azure CLI running in Azure Resource Manager (ARM) mode.

NOTE: Be sure you have an account for and a subscription to Microsoft Azure before deploying the vSRX to Azure (see Microsoft Azure).

If you do not have an Azure subscription, then you can create a free account before you begin. See the Microsoft Azure website for more details.

NOTE: From the Azure portal, you must first manually deploy the vSRX image (only once) by using either the **vSRX Next Generation Firewall (BYOL)** or the **vSRX Next Generation Firewall (PAYG)** SKU to accept the EULA terms. This is a requirement before you can deploy the vSRX image from the Azure CLI. By default, the Azure portal deployment tool uses **vSRX Next Generation Firewall (BYOL)** SKU as the source image. Use your Microsoft account username and password to log into the Microsoft Azure portal.

You will encounter a **MarketplacePurchaseEligibilityFailed** error if do not first accept the EULA terms for the vSRX image in the Azure portal before attempting to deploy the vSRX image from the Azure CLI.

Install the Microsoft Azure CLI

To install and log in to the Microsoft Azure CLI:

1. Install the Microsoft Azure CLI 1.0 as outlined in Install the Azure CLI. You have several options to install the Azure CLI package for either the Linux or Mac OS; be sure to select the correct installation package.

   NOTE: The vSRX for Azure deployment shell script **deploy-azure-vsrx.sh** is written in shell and Azure CLI version 1.0 commands and does not support Azure CLI version 2.0.

   NOTE: Deployment of vSRX to Microsoft Azure does not support the use of the Azure CLI from Microsoft Windows. This is because the **deploy-azure-vsrx.sh** shell script that is used as part of the deployment procedure can be run only from the Linux or Mac OS CLI.

2. Log into the Azure CLI.

   ```
   > azure login
   ```

3. At the prompt. copy the code that appears in the command output.

```
Executing command login
To sign in, use a web browser to open the page http://aka.ms/devicelogin. Enter the
codeXXXXXXXXX to authenticate
```

4. Open a Web browser to http://aka.ms/devicelogin, enter the code, and then click **Continue**. Enter your Microsoft Azure username and password credentials. When the process completes, the command shell completes the login process.

```
Added subscription Microsoft Azure Enterprise
To sign in, use a web browser to open the page http://aka.ms/deviceloginlogin command OK
```

> **NOTE**: If you have multiple Azure subscriptions, connecting to Azure grants access to all subscriptions associated with your credentials. One subscription is selected as the default, and used by the Azure CLI when performing operations. You can view the subscriptions, including the current default subscription, using the `azure account list` command.

5. Ensure that the Azure CLI is in Azure Resource Manager (ARM) mode.

```
> azure config mode arm
```

> **NOTE**: When the Azure CLI is initially installed, the CLI is in ARM mode.

Download the vSRX Deployment Tools

Juniper Networks provides a set of scripts, templates, parameter files, and configuration files in Juniper's GitHub repository. These tools are intended to help simplify the deployment of the vSRX to Azure when using the Azure CLI.

> **NOTE**: For background information on the scripts, templates, parameter files, and configuration files, see *Before You Deploy vSRX Using the Azure CLI*.

To download the vSRX deployment tools:

1. Access GitHub by using the following link: https://github.com/Juniper/vSRX-Azure.

2. Click **Clone or download** to download to you computer the `vSRX-Azure-master.zip` file from Github containing all files and directories from `vSRX-Azure`. The `vSRX-Azure-master` directory includes the following directories and files:

```
vSRX-Azure-master
├── README.md
├── LICENSE
├── sample-templates
│   ├── arm-templates-tool
│       ├── README.md
│       ├── deploy-azure-vsrx.sh
│       ├── templates
│       │   ├── app-vm
│       │   │   ├── vm.json
│       │   │   └── vm.parameters.json
│       │   └── vsrx-gateway
│       │       ├── vsrx.json
│       │       └── vsrx.parameters.json
│       └── utils
│           ├── decode_param_file.py
│           ├── gen_param_file.py
│           └── gen_template_file.py
│   ├── simple-vsrx-demo
│       ├── README.md
│       ├── vsrx.json
│       ├── vsrx.parameters.json
└── marketplace-solution-templates
    └── vpn-gateway
        ├── createUiDefinition.json
        ├── mainTemplate.json
        ├── vSRX-password.json
        └── vSRX-sshPublicKey.json
```

3. Extract the compressed `vSRX-Azure-master.zip` file to a location on your computer.

Change Parameter Values in the vsrx.parameter.json File

In the **vsrx.parameters.json** file, you need to modify parameter values specific to your vSRX deployment in Microsoft Azure. These parameters are used as part of the automatic deployment performed by the **deploy-azure-vsrx.sh** script.

Keep in mind that by default vSRX uses fxp0 as the egress interface to the Internet. For features requiring Internet connections that use a revenue port (such as VPN, UTM, and so on), routing instances are required to isolate the traffic between the management network and the revenue network.

To change parameter values in the **vsrx.parameters.json** file:

1. Open the **vsrx.parameters.json** file with a text editor.
2. Modify the values in the **vsrx.parameters.json** file based on the specifics of your vSRX deployment. As an example, the following table outlines the parameters in the **vsrx.parameters.json** file found in `sample-templates\arm-templates-tool\templates\vsrx-gateway` that might require modification.

> **CAUTION**: It is critical that you change the vsrx-username and vsrx-password login credentials listed in the **vsrx.parameters.json** file before you launch the vSRX instance and login for the first time. Note that you cannot reset login credentials for the vSRX using the Microsoft Azure portal or the Azure CLI.

Parameter	Default Value	Comment
storageAccountName	juniperstore01	Must be unique for each deployment.
storageContainerName	vhds	Name of the Microsoft Azure storage container (VHDs).
vsrx-name	vsrx-gw	Specifies the vSRX hostname.
vsrx-addr-ge-0-0-0	192.168.10.20	IP address of vSRX interface ge-0/0/0.0.
vsrx-addr-ge-0-0-1	192.168.20.20	IP address of vSRX interface ge-0/0/1.0.
vsrx-username	demo	Change to an appropriate username for the login credentials used to access the vSRX.

(Continued)

Parameter	Default Value	Comment
vsrx-password	Demo123456	Change to an appropriate password for the login credentials used to access the vSRX.
vsrx-sshkey	ssh-rsa placeholder	Specifies the root authentication password for the vSRX VM by entering an SSH public key string (RSA or DSA). By default, the **deploy-azure-vsrx.sh** deployment script selects the password authentication method, unless `-p`, followed by the SSH RSA public key file (id_rsa.pub by default), is specified. **NOTE**: Starting in Junos OS Release 15.1X49-D100 for vSRX, both password and SSH public key authentication are supported, and password authentication is chosen by default.
vsrx-disk	placeholder	The source image to create the vSRX instance. By default, the **deploy-azure-vsrx.sh** script uses the **vSRX Next Generation Firewall (BYOL)** SKU in the Azure Marketplace as the source image to deploy vSRX instance, unless `-i` is used to explicitly specify the vSRX instance image location.
vnet-prefix	192.168.0.0/16	IP address prefix of the virtual network.
vnet-mgt-subnet-basename	mgt-subnet	Name of management network connected to fxp0.

(Continued)

Parameter	Default Value	Comment
vnet-mgt-subnet-prefix	192.168.0.0/24	IP address prefix of management network connected to fxp0.
vnet-trust-subnet-basename	trust-subnet	Name of network connected to trust security zone: ge-0/0/1.0 on the vSRX.
vnet-trust-subnet-prefix	192.168.20.0/24	IP address prefix of network connected to trust security zone: ge-0/0/1.0 on the vSRX.
vnet-untrust-subnet-basename	untrust-subnet	Name of network connected to untrust security zone: ge-0/0/0.0 on the vSRX.
vnet-untrust-subnet-prefix	192.168.10.0/24	IP address prefix of network connected to untrust security zone: ge-0/0/0.0 on the vSRX.

3. Save your changes to the **vsrx.parameters.json** file.

Deploy the vSRX Using the Shell Script

The **deploy-azure-vsrx.sh** shell script deploys the vSRX virtual machine in a resource group that is based on your Azure Cloud geographic location. The script uses the storage account and network values defined in the **vsrx.parameters.json** file.

To deploy vSRX to the Azure virtual network:

1. At the bash prompt in the Azure CLI, run the **deploy-azure-vsrx.sh** script. By default, the script deploys the vSRX VM using the **vSRX Next Generation Firewall (BYOL)** SKU as the source image from the Azure Marketplace. The following information is read from the vsrx.json file as part of the deployment:
   - VM Size: Standard_D3_v2
   - Publisher: Juniper Networks
   - SKU: vsrx-byol-azure-image

- Offering: vsrx-next-generation-firewall

The following is an example of the command syntax. In this example, the script uses the vSRX image to deploy the vSRX VM in resource group “example_rg” at the Azure location “westus.” The storage account and network values are defined in the **vsrx.parameters.json** file.

```
> ./deploy-azure-vsrx.sh -g example_rg -l westus -f vSRX-Azure/sample-templates/arm-templates-tool/templates/
vsrx-gateway/vsrx.json -e vSRX-Azure/sample-templates/arm-templates-tool/templates/vsrx-gateway/
vsrx.parameters.json
```

NOTE: When you specify the vSRX source image URL with the option `-i`, the script copies the vSRX source image to create the virtual hardware disk file and to set the `vsrx-disk` parameter in **vsrx.parameters.json** to this value.

The default parameter values in the command syntax include:

- *example_rg* is the resource group name (`-g`).
- *westus* is the Azure location (`-l`).
- *vsrx.json* in the folder **vSRX-Azure/sample-templates/arm-templates-tool/templates/vsrx-gateway** is the default Azure template file (`-f`).
- *vsrx.parameters.json* in the folder **vSRX-Azure/sample-templates/arm-templates-tool/templates/vsrx-gateway** is the default parameter file (`-e`).

2. Monitor the stages of deployment of vSRX to Microsoft Azure as they occur on screen. Deployment encompasses operations such as creating a resource group, storage account, template group (including configuration parameters).

NOTE: Creation of the storage account can take approximately 3 to 5 minutes on average. However, in some cases, it might take as long as 15 to 20 minutes.

```
➜  arm-templates-tool ./deploy-azure-vsrx.sh
Use default resource group name 'vsrx'
info:    Executing command config mode
info:    New mode is arm
info:    config mode command OK
info:    Executing command group create
+ Getting resource group vsrx
+ Creating resource group vsrx
```

```
info:    Created resource group vsrx
data:    Id:                  /subscriptions/1c3367ba-71fc-48df-898a-d9eab4f1d673/
resourceGroups/vsrx
data:    Name:                vsrx
data:    Location:            westus
data:    Provisioning State:  Succeeded
data:    Tags: null
data:
info:    group create command OK
info:    Executing command storage account create
…
data:    DeploymentName     : deployvsrx
data:    ResourceGroupName  : vsrx
data:    ProvisioningState  : Succeeded
data:    Timestamp          : Thu Jul 20 2017 12:31:45 GMT+0800 (CST)
data:    Mode               : Incremental
data:    CorrelationId      : a99b89f8-5919-4dbc-b8a5-6d76b30fcb67
data:    DeploymentParameters :
data:    Name                          Type          Value
data:    ----------------------------  ------------  -------------------
data:    storageAccountName            String        jnprsa01
data:    storageContainerName          String        vhds
data:    vsrx-name                     String        vsrx-test01
data:    vsrx-addr-ge-0-0-0            String        192.168.10.20
data:    vsrx-addr-ge-0-0-1            String        192.168.20.20
data:    vsrx-username                 String        demo
data:    vsrx-password                 SecureString  undefined
data:    vsrx-sshkey                   String        ssh-rsa placeholder
data:    vsrx-disk                     String        placeholder
data:    vnet-prefix                   String        192.168.0.0/16
data:    vnet-mgt-subnet-basename      String        mgt-subnet
data:    vnet-mgt-subnet-prefix        String        192.168.0.0/24
data:    vnet-trust-subnet-basename    String        trust-subnet
data:    vnet-trust-subnet-prefix      String        192.168.20.0/24
data:    vnet-untrust-subnet-basename  String        untrust-subnet
data:    vnet-untrust-subnet-prefix    String        192.168.10.0/24
info:    group deployment create command OK
```

When the deployment process completes, you will see the message "`info: group deployment create command Ok`.

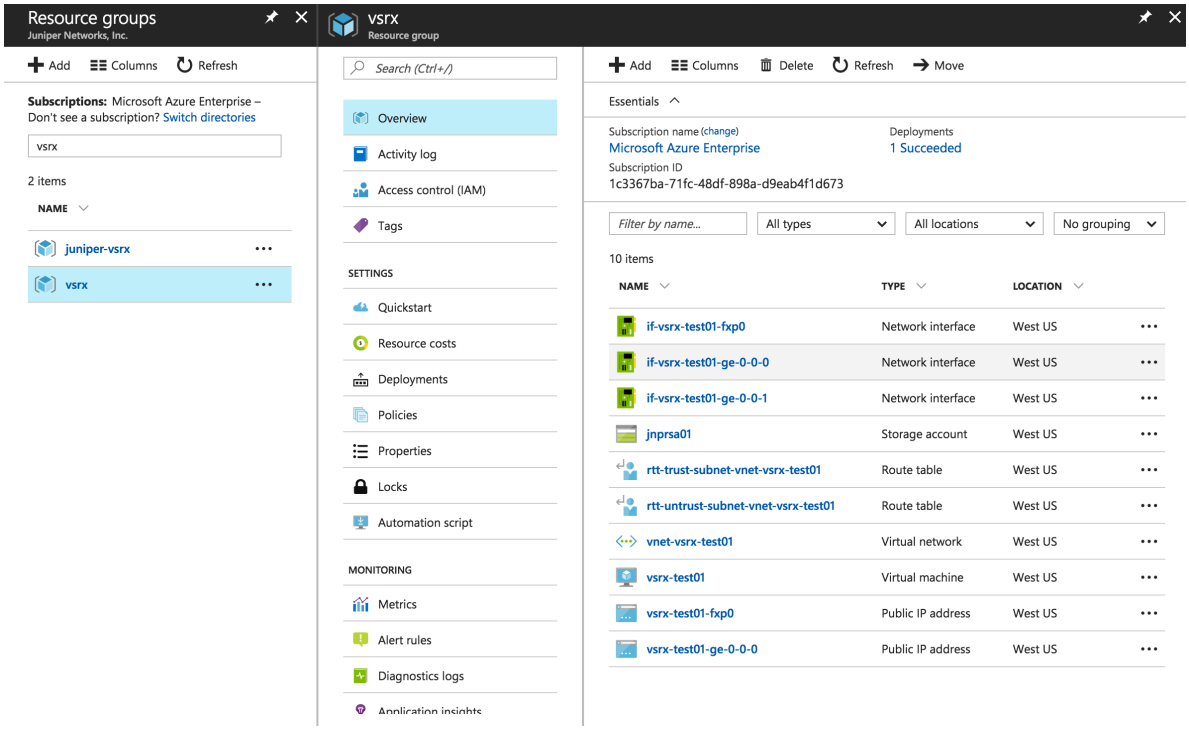
Verify Deployment of vSRX to Microsoft Azure

To verify the deployment of the vSRX instance to Microsoft Azure:

1. Open a Web browser to https://portal.azure.com/ and login to the Microsoft Azure portal using your login credentials. The Dashboard view appears in the Azure portal . You will see a unified dashboard for all your assets in Azure. Verify that the Dashboard includes all subscriptions to which you currently have access, and all resource groups and associated resources.
2. To view the vSRX resource group and its resources after deployment is completed, from the right-hand menu, click **Resource groups** to access the Resource Groups page.

   Figure 109 on page 439 shows an example of the Resources group page in the Microsoft Azure portal.

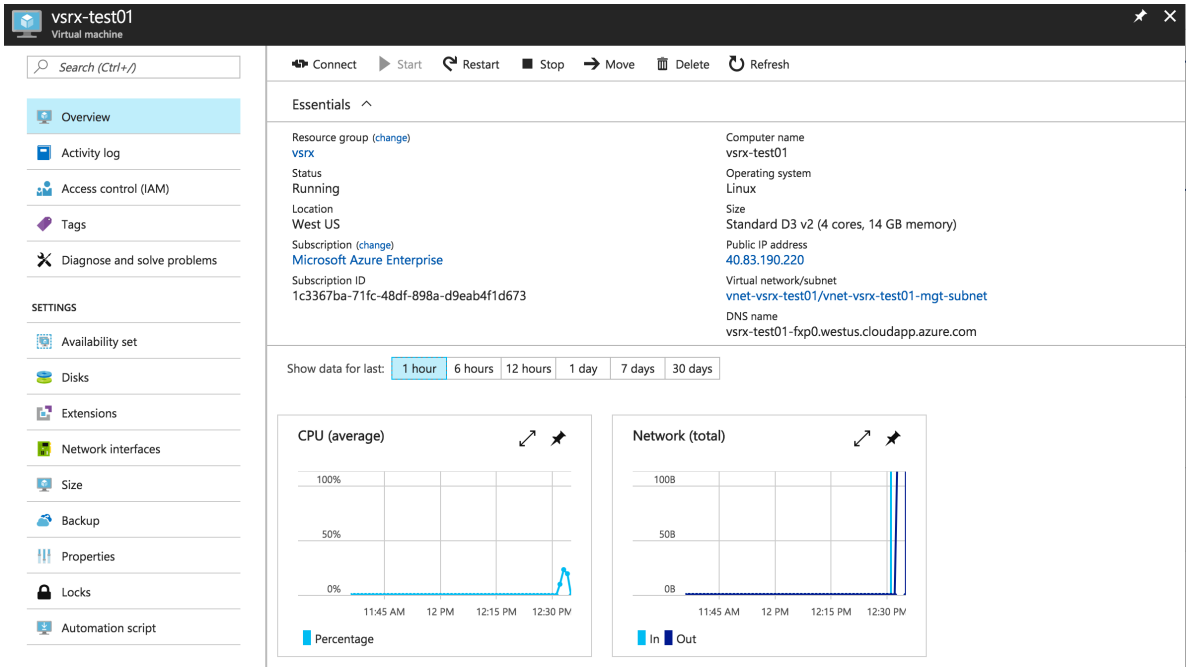
Figure 109: Microsoft Azure Resource Groups Page Example



3. To view details of the vSRX VM associated with the resource group, click the name of the vSRX.

Figure 110 on page 440 shows an example of the Resource groups VM in the Microsoft Azure portal.

Figure 110: Microsoft Azure Resource Groups VM Example

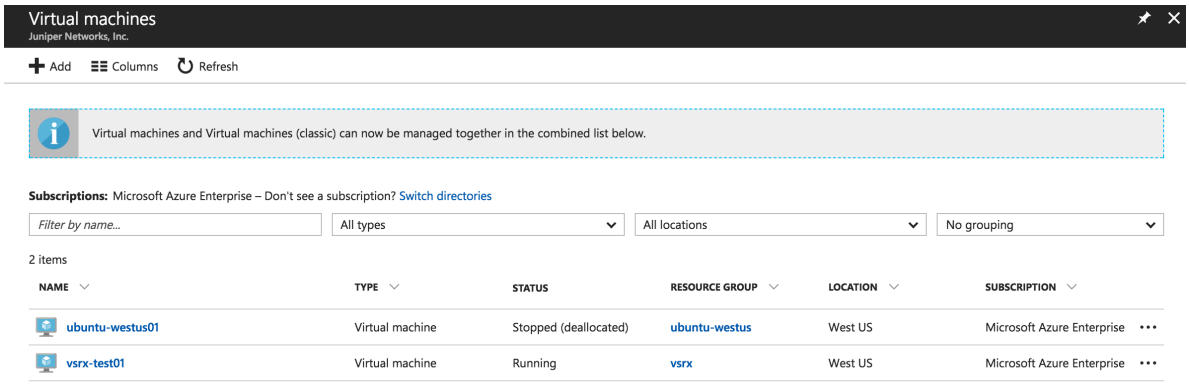


4. To see a summary view of the VMs in your subscription, including the newly deployed vSRX, click the Virtual Machines icon in the left pane. On the Virtual machines page, check the vSRX VM status after deployment is completed. Observe that the status is `Running`.

> **NOTE**: You can stop, start, restart, and delete a VM from the Virtual machines page in the Microsoft Azure portal.

Figure 111 on page 441 shows an example of the Microsoft Azure Virtual machines page.

Figure 111: Microsoft Azure Virtual Machines Page Example



Log In to a vSRX Instance

After vSRX deployment is completed, the vSRX instance is automatically powered on and launched. At this point you can use an SSH client to log in to the vSRX instance.

> **NOTE**: In Microsoft Azure, individuals and enterprises can host servers and services on the cloud as a pay-as-you-go (PAYG) or bring-your-own-license (BYOL) service. For the vSRX on Microsoft Azure deployment, only the BYOL model is supported.

To log in to the vSRX VM:

1. From the Azure portal, click **Resource groups** from the menu of services on the dashboard, and then select the vSRX VM. Locate the public IP address of the vSRX VM from the Settings blade.
2. Use an SSH client to log in to a vSRX instance.
3. At the prompt, enter the following login credentials:

> **NOTE**: Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, only password authentication is supported. Starting in Junos OS Release 15.1X49-D100 for vSRX, both password and SSH public key authentication are supported, and password authentication is chosen by default.
>
> The vSRX instance is automatically configured for username and password authentication. To log in, use the login credentials that were defined in the **vsrx.parameters.json** file (see

"Change Parameter Values in the vsrx.parameter.json File" on page 433). After initially logging to the vSRX, you can configure SSH public and private key authentication.

```
# ssh <username@vsrx_vm_ipaddress>
```

```
The authenticity of host 'x.x.x.x (x.x.x.x)' ...
ECDSA key fingerprint is SHA256:XXXXXXXXXXXXXXXXXXXXXXX.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'x.x.x.x' (ECDSA) to the list of known hosts.
Password: xxxxxxxx
username@vsrx_vm_ipaddress>
```

4. Configure the basic settings for the vSRX VM (see *Configure vSRX Using the CLI*).

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, you can deploy the vSRX from the Azure CLI and customize the vSRX VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud.
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, only password authentication is supported.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 for vSRX, both password and SSH public key authentication are supported, and password authentication is chosen by default.
15.1X49-D100	Starting in Junos OS Release 15.1X49-D100 for vSRX, both password and SSH public key authentication are supported, and password authentication is chosen by default.

RELATED DOCUMENTATION

Connect from Microsoft Azure CLI

CHAPTER 23

Configure and Manage vSRX for Microsoft Azure

IN THIS CHAPTER

Configure vSRX Using the CLI

To configure the vSRX instance using the CLI:

1. Verify that the instance is powered on.
2. Log in using the username and password credentials for your vSRX VM deployment.
3. Start the CLI.

```
root#cli
root@>
```

4. Enter configuration mode.

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

5. Set the root authentication password by entering a *cleartext* password, an encrypted password, or an SSH public key string (*DSA* or *RSA*).

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

```
New password: password
Retype new password: password
```

6. Configure the traffic interfaces.

```
[edit]
root@# set interfaces ge-0/0/0 unit 0 family inet address assigned_ip/netmask
root@# set interfaces ge-0/0/1 unit 0 family inet address assigned_ip/netmask
```

> **NOTE**: Configuration of the management interface fxp0 for the vSRX is not necessary, because it is configured during vSRX VM deployment. Do not change the configuration for interface fxp0 and the default routing table or you will lose connectivity.

7. Configure routing interfaces to isolate management network and traffic network.

```
[edit]
root@# set routing-instances vsrx-vr1 instance-type virtual-router
root@# set routing-instances vsrx-vr1 interface ge-0/0/0.0
root@# set routing-instances vsrx-vr1 interface ge-0/0/1.0
```

8. Verify the configuration changes.

```
[edit]
root@# commit check
configuration check succeeds
```

9. Commit the current configuration to make it permanent and to avoid the possibility of losing connectivity to the vSRX instance.

```
[edit]
root@# commit confirmed
commit confirmed will be automatically rolled back in 10 minutes unless confirmed
commit complete
# commit confirmed will be rolled back in 10 minutes
```

10. Commit the configuration to activate it on the instance.

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

11. Optionally, use the `show` command to display the configuration to verify that it is correct.

NOTE: Certain Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds to each licensed feature. To conform to software feature licensing requirements, you must purchase one license per feature per instance. The presence of the appropriate software unlocking key on your virtual instance allows you to configure and use the licensed feature. See Managing Licenses for vSRX for details.

RELATED DOCUMENTATION

- Junos OS for SRX Series
- CLI User Guide

Configure vSRX Using the J-Web Interface

IN THIS SECTION

Access the J-Web Interface and Configuring vSRX

Use the Junos OS CLI to configure, at a minimum, the following parameters before you can access a vSRX VM using J-Web:

CAUTION: Do not change the configuration for interface fxp0 and default routing table or you will lose connectivity to the vSRX instance.

To configure vSRX using the *J-Web* Interface:

1. Launch a Web browser from the management instance.
2. Enter the vSRX fxp0 interface IP address in the Address box.
3. Specify the username and password.
4. Click **Log In**, and select the **Configuration Wizards** tab from the left navigation panel. The J-Web Setup wizard page opens.
5. Click **Setup**.

   You can use the Setup wizard to configure the vSRX VM or edit an existing configuration.

   - Select **Edit Existing Configuration** if you have already configured the wizard using the factory mode.
   - Select **Create New Configuration** to configure the vSRX VM using the wizard.

     The following configuration options are available in the guided setup:

     - Basic

       Select **basic** to configure the vSRX VM name and user account information as shown in Table 80 on page 446.

       - Instance name and user account options

         Table 80: Instance Name and User Account Information

Field	Description
Instance name	Type the name of the instance. For example: **vSRX**.
Root password	Create a default root user password.
Verify password	Verify the default root user password.

Table 80: Instance Name and User Account Information *(Continued)*

Field	Description
Operator	Add an optional administrative account in addition to the root account. User role options include: • **Super User**: This user has full system administration rights and can add, modify, and delete settings and users. • **Operator**: This user can perform system operations such as a system reset but cannot change the configuration or add or modify users. • **Read only**: This user can only access the system and view the configuration. • **Disabled**: This user cannot access the system.

- Select either **Time Server** or **Manual**. Table 81 on page 447 lists the system time options.

Table 81: System Time Options

Field	Description
Time Server	
Host Name	Type the hostname of the time server. For example: **ntp.example.com**.
IP	Type the IP address of the time server in the IP address entry field. For example: **192.0.2.254**.
NOTE: You can enter either the hostname or the IP address.	
Manual	
Date	Click the current date in the calendar.
Time	Set the hour, minute, and seconds. Choose **AM** or **PM**.
Time Zone (mandatory)	

Table 81: System Time Options *(Continued)*

Field	Description
Time Zone	Select the time zone from the list. For example: GMT Greenwich Mean Time GMT.

- Expert

  a. Select **Expert** to configure the basic options as well as the following advanced options:

  - Four or more internal zones
  - Internal zone services
  - Application of security policies between internal zones

  b. Click the **Need Help** icon for detailed configuration information.

  You see a success message after the basic configuration is complete.

Apply the Configuration

To apply the configuration settings for vSRX:

1. Review and ensure that the configuration settings are correct, and click **Next**. The Commit Configuration page appears.
2. Click **Apply Settings** to apply the configuration changes to vSRX.
3. Check the connectivity to the vSRX instance because you might lose connectivity if you have changed the management zone IP. Click the URL for reconnection instructions on how to reconnect to the instance.
4. Click **Done** to complete the setup.

   After successful completion of the setup, you are redirected to the J-Web interface.

> **CAUTION**: After you complete the initial setup, you can relaunch the J-Web Setup wizard by clicking **Configuration>Setup**. You can either edit an existing configuration or create a new configuration. If you create a new configuration, the current configuration in vSRX will be deleted.

Add vSRX Feature Licenses

Certain Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds to each licensed

feature. To conform to software feature licensing requirements, you must purchase one license per feature per instance. The presence of the appropriate software unlocking key on your virtual instance allows you to configure and use the licensed feature.

To understand more about vSRX Licenses, see, Licenses for vSRX. Please refer to the Licensing Guide for general information about License Management. Please refer to the product Data Sheets for further details, or contact your Juniper Account Team or Juniper Partner.

Remove a vSRX Instance from Microsoft Azure

To remove a vSRX instance from Microsoft Azure:

1. Log in to the Azure Portal.
2. In the left pane of the Azure Portal, click the Virtual Machines icon.
3. To remove the vSRX instance, in the right pane, select the vSRX instance you want to remove, then click Delete.

   NOTE: You can delete a VM when the VM is running. If desired, you can stop the vSRX instance before deleting.

4. To delete the disks attached to the deleted vSRX virtual machine, click Delete and then select Delete the Associated VHD.
5. To delete the related cloud service for the deleted vSRX virtual machine, access the Cloud Service tab and click Delete to remove the related cloud services.

Upgrade Junos OS Software on a vSRX Instance

IN THIS SECTION

This section outlines how to upgrade Junos OS software on your vSRX instance to a newer release. Depending upon your preference, you can replace the vSRX software in one of two ways:

Upgrade the Junos OS for vSRX Software Release

You can directly upgrade the Junos OS for vSRX software using the CLI. Upgrading or downgrading Junos OS can take several hours, depending on the size and configuration of the network. You download the desired Junos OS Release for vSRX .tgz file from the Juniper Networks website.

You also can upgrade using J-Web (see J-Web) or the Junos Space Network Management Platform (see Junos Space).

For the procedure on upgrading a specific Junos OS for vSRX software release, see the *Migration, Upgrade, and Downgrade Instructions* topic in the release-specific *vSRX Release Notes* available on the vSRX TechLibrary.

Replace the vSRX Instance on Azure

To replace a vSRX instance on Azure with a different software release:

1. Log in to the vSRX instance using SSH and start the CLI.

```
root@% cli
root@>
```

2. Enter configuration mode.

```
root@> configure
root@#
```

3. Copy the existing Junos OS configuration from the vSRX. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it.

NOTE: By default, the configuration is saved to a file in your home directory.

- See Saving a Configuration File for additional background information on saving a Junos OS configuration file.
- See file copy for information on how to copy files from one location to another location on the local device or to a location on a remote device that is reachable by the local device.

```
root@#save <filename>
[edit]
root@#
```

4. Remove the vSRX instance on Azure as described in *Remove a vSRX Instance from Microsoft Azure*.
5. Once the vSRX instance on Azure has been successfully removed, define the specifics of a vSRX instance prior to launching it.
6. Launch the vSRX image using the desired software version available from Azure Marketplace.
7. Load the previously copied Junos OS configuration file onto your new (upgraded) vSRX instance as described in Loading a Configuration File.

CHAPTER 24

Configure Azure Features on vSRX and Use Cases

IN THIS CHAPTER

Deployment of Microsoft Azure Hardware Security Module on vSRX 3.0

IN THIS SECTION

Microsoft Azure Key Vault Hardware Security Module Integration Overview

Microsoft Azure Key Vault hardware security module (HSM) is a cloud service that works as a secure secrets store. You can securely store keys, passwords, certificates, and other secrets. This service from cloud vendors helps us to securely generate, store and manage Crypto keys. vSRX applications use these

Crypto keys to protect data at rest, such as private keys, passwords and other sensitive data. Azure Key Vault HSM can also be used as a Key Management solution. Azure Key Vault makes it easy to create and control the encryption keys used to encrypt your data. When you provide the master encryption password then that password is used to encrypt the sensitive data and save encrypted data (AES256) on disk. The master encryption password is also protected using RSA key-pair generated and stored in HSM.

vSRX (mgd process) generates hash of configuration. This hash (and other sensitive data) is protected using master encryption password as key for AES-GCM 256 encryption.

The master password is used to protect secrets such as the RADIUS password, IKE preshared keys, and other shared secrets in the Junos OS management process (mgd) configuration. The master password is protected using the master encryption password. The master password itself is not saved as part of the configuration. The password quality is evaluated for strength, and the device gives feedback if weak passwords are used.

Sensitive data such as PKI private keys and configuration that are stored in plain text on vSRX 3.0 instances can now be protected using HSM service.

When you enable Microsoft Azure Key Vault HSM on vSRX, vSRX creates, an RSA key pair of 2048 size and uses it to encrypt, a PKI private key file located in **/var/db/certs/common/key-pairs**, configuration hash and a master password, which is saved in: **/config/unrd-master-password.txt**.

NOTE: Existing keypairs prior to enabling HSM will not be encrypted and are deleted.

By enabling the HSM, the software layer leverages the use of the underlying HSM service that protects sensitive information such as private keys, system master passwords, and so on, by storing the information using 256-bit AES encryption (instead of storing in cleartext format). The device also generates a new SHA256 hash of the configuration each time the administrator commits the configuration. This hash is verified each time the system boots up. If the configuration has been tampered with, the verification fails and the device will not continue to boot. Both the encrypted data and the hash of the configuration are protected by the HSM module using the master encryption password.

Hash validation is performed during any commit operation by performing a validation check of the configuration file against the saved hash from previous commits. In a chassis cluster system, hash is independently generated on the backup system as part of the commit process.

Hash is saved only for the current configuration and not for any rollback configurations. Hash is not generated during reboot or shutdown of the device.

vSRX uses HSM to encrypt the following secrets:

- SHA256 hash of the configuration

- Device master password
- All key pairs on the device

Keys created by each vSRX 3.0 instance will be tagged and/or named using the UUID of each VM. You can log in to the cloud portal, access the keys, and verify their properties or the operations requested.

Configure Microsoft Azure Key Vault HSM on vSRX 3.0

Key vault on Azure stack provides cloud HSM service for all Azure applications. All applications need to be registered in Azure active directory to use services such as Key Vault.

vSRX3.0 is integrated with Microsoft Azure Cloud HSM when running on Azure. You can login to cloud portal, access the keys, and verify their properties or operations requested for.

For each public cloud vendor, there are unique steps to be performed to integrate vSRX with cloud HSM. This section provides the steps needed to integrate vSRX 3.0 with Microsoft Azure Key Vault HSM.

You will need the following listed items to integrate vSRX with Microsoft Azure Key Vault HSM:

- vSRX 3.0 instance
- Microsoft Azure Key vault
- Setup key vault authentication for vSRX
- Microsoft Azure-specific configurations for integrating HSM

Microsoft Azure Key Vault is a cloud-hosted management service that allows users to encrypt keys and small secrets by using keys that are protected by hardware security modules (HSMs).

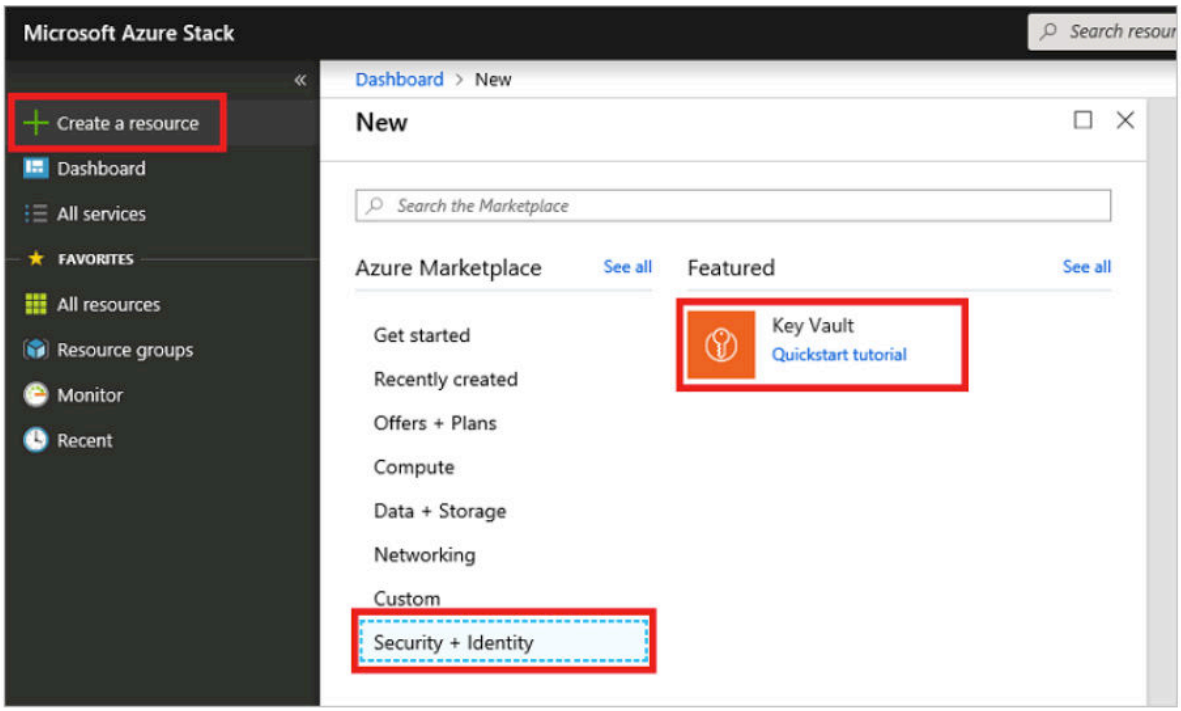
This procedure provides the general steps to integrate Microsoft Azure Key Vault HSM with vSRX 3.0.

1. Launch vSRX 3.0 instance in Microsoft Azure environment.

   For launching vSRX 3.0 instances see, vSRX Deployment Guide for Microsoft Azure Cloud.

2. Create Key vault. From the dashboard, select **+ Create a resource**, **Security + Identity**, and then **Key Vault** as shown in Figure 112 on page 455.

Figure 112: Create Key Vault



You need to create “premium” key vault to access cryptographic key features needed by vSRX 3.0. After you create a key vault, for more information on how to create and manage keys and secrets within the vault, see Manage Key Vault in Azure Stack using the portal.

3. Enable managed identity for vSRX 3.0.

   System assigned managed identity helps vSRX authenticate to other services (example Key vault) without saving credentials in the code by registering your application to Azure Active directory. Enabling this identity will generate unique object ID, which can be used to refer it across other vSRX instances.

   To enable managed identity for vSRX on Microsoft Azure, you need to configure managed identities for Microsoft Azure resources on a VM using the Azure portal as shown in Figure 113 on page 456 and Figure 114 on page 457.

For more information, see Configure managed identities for Azure resources on a VM using the Azure portal

Figure 113: Enable System Assigned Managed Identity During Creation of a VM

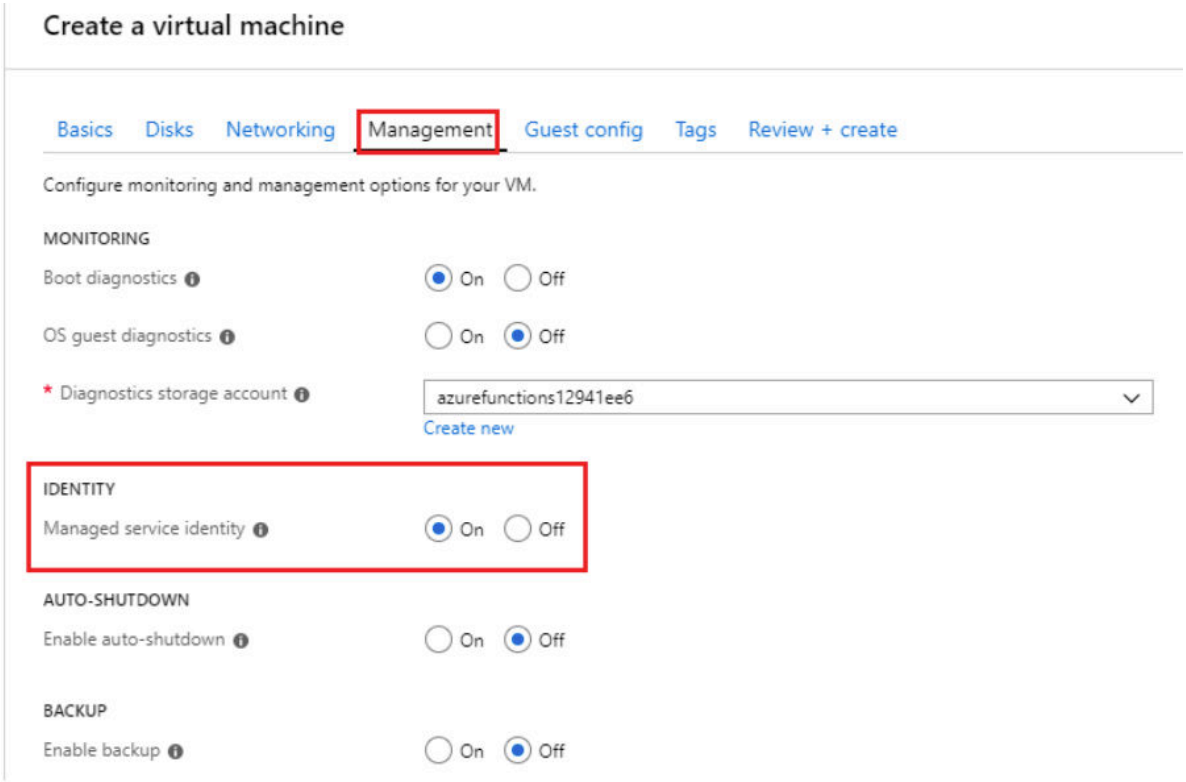
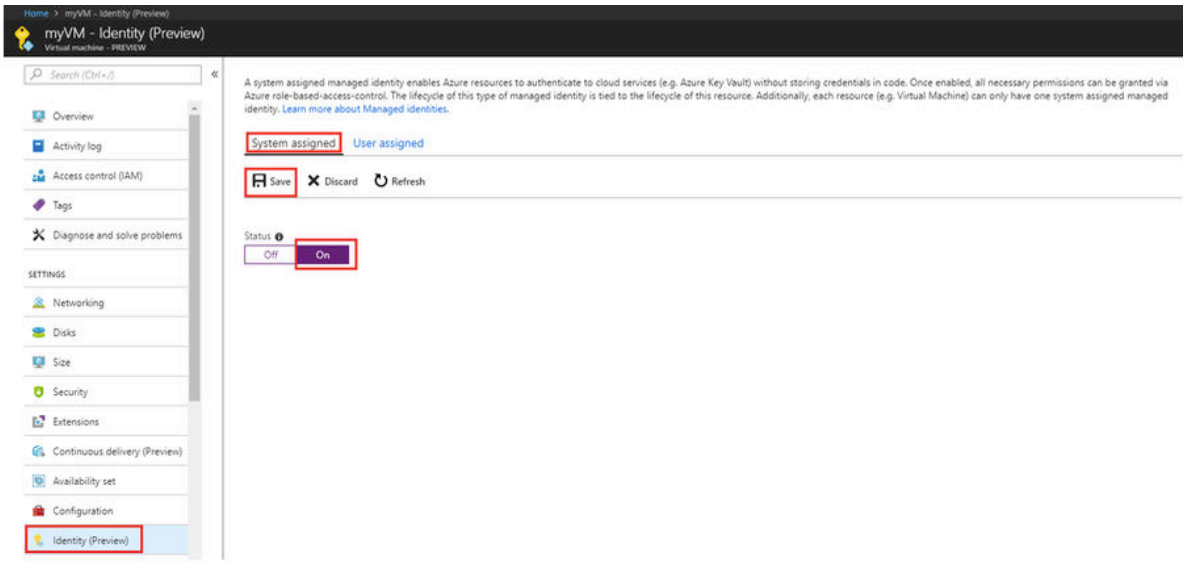


Figure 114: Enable System Assigned Managed Identity on an Existing VM



4. Add access policy in Microsoft Azure Key Vault.

   For applications such as vSRX 3.0 VM to access Microsoft Azure Key Vault, access policies have to be enabled. For more information on how to add new policy, see Secure access to a key vault refer this link to add new policy.

   Steps to add access policy in Microsoft Azure Key Vault are:

   a. Go to **Key Vault Resource** page on Microsoft Azure portal.

   b. Click **Access Policies** tab on the left side of the page.

   c. Click on **Add New** tab and then click **Select Principal**, where you search for your vSRX user name assigned when it was created.

   d. Select all the **key permissions** and click **Save**.

   > **NOTE**: Do not select any **Authorized application**.

5. Check fxp0 (management) interface status

   vSRX3.0 uses fxp0 for communication with the Microsoft Azure Key Vault. Use the `show interface terse fxp0` command and ensure to check if fxp0 is configured and is able to ping external servers.

NOTE: vSRX 3.0 connects to cloud HSM using management interface. If management interface is not configured or does not get connected, then cloud HSM features cannot be used.

6. Enable and start communicating with key vault.

   - To enable key vault, run the `request security hsm set key-vault <name-of-key-vault>` command.

     NOTE: URL used to access Microsoft Azure Key Vault is generally in the format as: https://<name-of-key-vault>.vault.azure.net/keys.

   - To establish communication with key vault, create RSA key pair in HSM, generate and encrypt configuration hash, and encrypt master password and PKI key pair files run the `request security hsm master-encryption-password set plain-text-password`.

   - You will be prompted to enter the master encryption password twice, to make sure that these passwords match. The master encryption password is validated for required password strength. After the master encryption password is set, the system encrypts the sensitive data with the master encryption password, that is encrypted by the MEK that is owned and protected by HSM.

   - To configure the master password run the `set system master-password plain-text-password` command. Otherwise, certain sensitive data will not be protected by the HSM. If HSM is not enabled, master password will be saved in plain text format in the **/config/unrd-master-password.txt** file

     NOTE: To ensure master password is not saved as plain text on vSRX 3.0, an error will be displayed on console indicating that, it is insecure to set master password without enabling HSM and command operation will be terminated.

Change the Master Encryption Password

If you want to change the master encryption password then you can run the `request security hsm master-encryption-password set plain-text-password` command from operational mode:

NOTE: It is recommended that no configuration changes are made while you are changing the master encryption password.

The system checks if the master encryption password is already configured. If master encryption password is configured, then you are prompted to enter the current master encryption password.

The entered master encryption password is validated against the current master encryption password to make sure these master encryption passwords match. If the validation succeeds, you will be prompted to enter the new master encryption password as plain text. You will be asked to enter the key twice to validate the password.

The system then proceeds to re-encrypt the sensitive data with the new master encryption password. You must wait for this process of re-encryption to complete before attempting to change the master encryption password again.

If the encrypted master encryption password file is lost or corrupted, the system will not be able to decrypt the sensitive data. The system can only be recovered by re-importing the sensitive data in clear text, and re-encrypting them.

Verify the Status of the HSM

IN THIS SECTION

Purpose

To check connectivity with HSM.

Action

You can use the `show security hsm status` command to verify the status of the HSM. The following information is displayed:

- If HSM is enabled and reachable or disabled
- Is Master Binding Key (RSA Key pair) created in HSM
- Is Master Encryption Key configured - master encryption password status (set or not set)
- Cloud vendor Information

request security hsm master-encryption-password

IN THIS SECTION

Syntax

```
request security hsm master-encryption-password set plain-text-password
```

Release Information

Command introduced in Junos OS Release 19.4R1.

Description

Use this command to set or replace the password (in plain text).

Options

plain-text-password	Set or replace the password (in plain text).

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security hsm master-encryption-password set plain-text-password

```
user@host>              request security hsm master-encryption-password set plain-text-password
```

```
Enter new master encryption password:
Repeat new master encryption password:
Binding password with HSM
Master encryption password is bound to HSM
Encoding master password ..
Successfully encoded master password
Deleting all previous local certificates, keypairs and certificate requests
```

show security hsm status

IN THIS SECTION

Syntax

```
show security HSM status
```

Release Information

Command introduced in Junos OS Release 19.4R1.

Description

Display the current status of the Hardware Security Module (HSM). You can use this `show security hsm status` command to check the status of HSM, master binding key, master encryption password, and cloud vendor details.

Options

This command has no options.

Required Privilege Level

security

Output Fields

Table 82 on page 462 lists the output fields for the `show security hsm status` command.

Table 82: show security hsm status Output Fields

Field Name	Field Description
`Enabled`	Specifies whether HSM is enabled or disabled.
`Master Binding Key`	Displays the HSM's Master Binding Key status whether it is created or not created in HSM. HSM generates cryptographic keys and encrypts them so that those can only be decrypted by the HSM. This process is know as binding. Each HSM has a master binding key, which is also know as storage root key.
`Master Encryption Key`	Displays Master Encryption configuration status whether it is set or not set. The encrypted data and the hash of the configuration is protected by vSRX using Microsoft Key Vault (HSM) service.
`Cloud vendor Details`	Displays the details specific to the cloud vendor.

Sample Output

show security hsm status (HSM status command output when vSRX initially boots up but this feature is not enabled)

```
user@host>              show security hsm status
```

```
HSM Status:
  Accessible: no
  Master Binding Key: not-created
  Master Encryption Key: not-configured
  Azure Key Vault: unknown
```

Sample Output

show security hsm status (HSM status command output after successful integration with key vault)

```
user@host>              show security hsm status
```

```
HSM Status:
  Accessible: yes
  Master Binding Key: created
  Master Encryption Key: configured
  Azure Key Vault: vsrx3-hsm-kv
```

SEE ALSO

request security hsm master-encryption-password

Deployment of Microsoft Azure Hardware Security Module on vSRX 3.0 | 452

Understanding VPN Functionality with Microsoft Azure Key Vault HSM Service

IN THIS SECTION

With the integration of Microsoft Azure Key Vault HSM Service on vSRX3.0, you can now use the HSM service to create, store, and perform the required VPN keypair operations. Keypair creation is now enabled in HSM service.A PKI based VPN tunnel can now be established using the keypairs generated using the HSM. Once the master encryption key is configured, you can configure the VPN functionality using HSM service. You can generate only RSA keypairs of length 2048 and 4096 bits. Operations such as private key signing during CSR creation in PKID, private key signing during verification of the certificate received from the CA server in PKID, and private key signing during IKE negotiations at IKED is off-loaded from vSRX and is now performed by the HSM service.

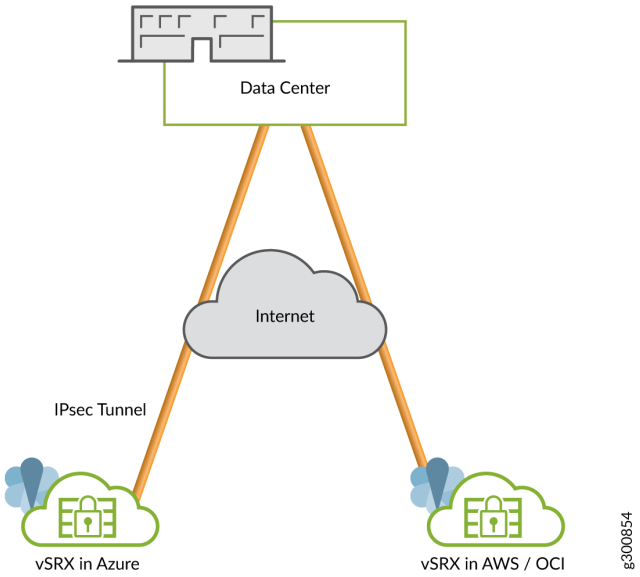
NOTE: Keypair generation using HSM service is only for pkid and iked processes. Also, existing keypairs in the filesystem before HSM service is enabled are not encrypted and those keypairs are deleted.

Deployment Scenario

This section provides a deployment scenario where vSRX 3.0 instance is launched as a gateway in a virtual network connecting to a data center using a pure IPsec connection.

Figure 115 on page 465 shows the deployment scenario.

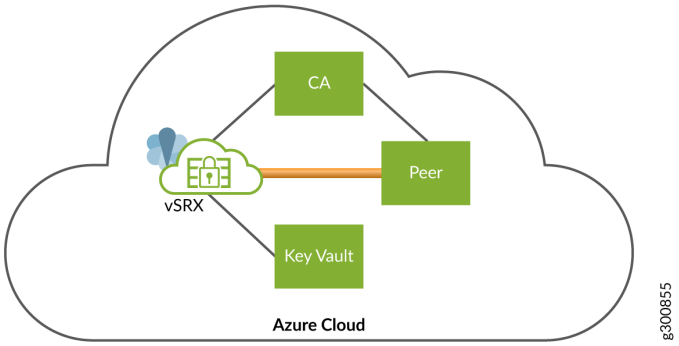
Figure 115: Deployment Scenario of vSRX using an IPsec Connection



You can generate key pairs using Microsoft Azure cloud HSM service for pkid process and use these keypairs for getting a local certificate from the CA server. Use the keypair present in the cloud HSM service for private key signing during IKE negotiations.

The VPN functionality performed within the Microsoft Azure cloud using HSM service is as shown in Figure 116 on page 465.

Figure 116: Components for VPN with HSM in Microsoft Azure Cloud



The components involved here are:

- vSRX 3.0 launched in the Microsoft Azure cloud.
- Peer—Second vSRX 3.0 instance launched in the Azure cloud. A tunnel is established between the frist vSRX 3.0 and and the Peer.
- Key Vault—The HSM service launched in the Azure cloud. You can interact between the vSRX 3.0 and the HSM. and the peer can create and store keypairs locally.
- Certificate Authority Server—Any CA server that can be accessed by the vSRX instances. The CA server is launched on the Azure Cloud.

This procedure provides steps on how to allow access from vSRX to the HSM by authenticating the vSRX with the cloud HSM service.

1. Initialize a session with the HSM service—Each process that needs to interact with the HSM has to initialize a separate session of its own. For the VPN functionality you must establish 2 sessions with the HSM service for each device involved. One session is established with pkid process and another session with iked process. These sessions with the HSM service are established only once during the init process of the daemon. If a daemon is restarted, a new session is established with the HSM service. When a session is successfully established with the HSM service, a valid session context is returned. Sessions will be established with the HSM service only if Master Encryption Key (MEK) is enabled. Each session will be a secure TLS connection between the vSRX and the cloud HSM.
2. Handling Keypairs at the HSM—To create and store keypairs at the HSM use the `request security pki generate-key-pair certificate-id certificate-id-name <size> <type>` command.

   NOTE: The term certificate-id is just an identifier associated with the keypair that has been generated. There is no connection to a certificate creation yet. If no type and size are mentioned, then the default values of type as RSA and and size of 2048 is considered.

3. Redirection to the HSM—With HSM enabled, the same CLI command will be redirected to the HSM. A new keypair with the given parameters is created at the HSM. Keys created by each vSRX will be tagged using the UUID of each VM. You can login to cloud portal, access the keys and verify their properties/operations that you want. The UUID of each key is of the following format:**<key-name>_<unique vm-instance id>**. You need to provide the key name at the time of key creation. The VM instance is the factor that will make the key id unique in the HSM service. Thus, it is required that the **vm-instance id** must be unique for each VM which is up and running. This is ensured by Microsoft Azure. The HSM redirection will be a timed call, wherein if no response is received within *x* seconds, then an error message `call to HSM failed` is displayed.
4. Retrieval of Public Key Information—After the creation of the keypair at the HSM, we retrieve the public key components of the keypair. The HSM returns the modulus and the exponent. These components are converted into EVP_PKEY structure using OpenSSL API's. The public key structure is then stored as a new entry in the hash of keys. In this way, the public key components can be

retrieved from the hash when required. Currently, the HSM does not detect duplicate keypairs, instead when error key id is received again, the HSM will overwrite the pre-existing keypair. To avoid this overwrite of keypairs, the public key is saved in the hash at the time of key creation itself. This way, a duplicate keypair creation is stopped at the device level itself, without making a call to the HSM.

You will receive an error `error: Failed to generate key pair at HSM. Found a key with the same name at HSM. Use a different certificate id next time. Refer to PKID logs for more details` when you try to use the same name to create a new keypair, even if you have deleted the previous keypair.

5. Deletion of Keypairs—HSM does not support an API to delete keypairs created at the HSM. The delete keypair command issued at the CLI will result in the public key component being deleted from the disk and the key hash. The keypair will not be deleted from the HSM. To delete the keypair from the HSM, you need to access the HSM and manually delete the keypair. If Azure key vault has soft delete feature enabled, you will also need to eliminate the keypair from the keypair before you can re-use the keypair name.

   NOTE: Exporting keys from the file is not supported. When you use the `request security pki local-certificate export` and `request security pki key-pair export` commands to export keys, you will receive an error message `Export of keypairs/certificate is not supported when HSM is enabled.`

6. Private Key Signing—The private key is now present at the HSM. So, all operations requiring the private key have been offloaded to the HSM. The operations involve:

   Private key signing operation are used during:

   - Creating the Certificate Signing Request (CSR)
   - Verification of the local certificate received from the CA
   - RSA signing during IKE negotiations
   - SHA-1 Inter-operability. The Azure key vault supports private key signing for only SHA-256 digests.

CLI Behavior With and Without HSM

CLI	Non-HSM	HSM
`request security pki generate-key-pair`	Creates a keypair locally	Creates a keypair at the HSM

request security pki generate-certificate-request	Creates a CSR locally	Contacts the HSM for private key signing while creating the CSR. Digest has to be SHA-256
request security pki local-certificate enroll	Creates a CSR locally. Sends the CSR to the CA server and receives a certificate	Contacts the HSM for private key signing while creating the CSR. Sends the CSR to the CA server and receives a certificate. Digest has to be SHA-256
request security pki local-certificate export	Exported local certificate to other device	Not possible as key pair not present locally
request security pki key-pair export	Exported locally present key pair to other device	Not possible as key pair not present locally
request security pki local-certificate generate-self-signed	Generates self signed certificate	Contacts HSM for signing and then generates self signed certificate
show security pki local-certificate	Shows local certificate present on device	Shows keypair is generated locally or at cloud HSM

request security pki local-certificate enroll scep

IN THIS SECTION

Syntax

```
request security pki local-certificate enroll scep
    ca-profile  ca-profile name
    certificate-id certificate-id-name
    challenge-password challenge-password
    digest (sha-1 | sha-256)
    domain-name domain-name
    email email-address
     ip-address ip-address
    ipv6-address ipv6-address
    scep-digest-algorithm (md5 | sha-1)
    scep-encryption-algorithm (des | des3)
    subject subject-distinguished-name
```

Release Information

Command introduced in Junos OS Release 9.1. Serial number (SN) option added to the subject string output field in Junos OS Release 12.1X45. `scep` keyword and `ipv6-address` option added in Junos OS Release 15.1X49-D40.

Starting in Junos OS Release 20.1R1 on vSRX 3.0, you can safeguard the private keys used by PKID and IKED using Microsoft Azure Key Vault hardware security module (HSM) service. You can establish a PKI based VPN tunnel using the keypairs generated at the HSM. The hub `certificate-id` option under certificate-id is not available for configuration after generating HSM key-pair.

Starting in Junos OS Release 20.4R1 on vSRX 3.0, you can safeguard the private keys used by PKID and IKED using AWS Key Management Service (KMS). You can establish a PKI based VPN tunnel using the keypairs generated by the KMS. The hub `certificate-id` option under certificate-id is not available for configuration after generating PKI key-pair.

Description

Enroll and install a local digital certificate online by using Simple Certificate Enrollment Protocol (SCEP).

If you enter the `request security pki local-certificate enroll` command without specifying the `scep` or `cmpv2` keyword, SCEP is the default method for enrolling a local certificate.

Options

ca-profile *ca-profile-name*	CA profile name.
certificate-id *certificate-id-name*	Name of the local digital certificate and the public/private key pair.
challenge-password *password*	Password set by the administrator and normally obtained from the SCEP enrollment webpage of the CA. The password is maximum 256 characters in length. You can enforce the limit to the required characters.
digest (sha-1 | sha-256)	Hash algorithm used for signing RSA certificates, either SHA-1 or SHA-256. SHA-1 is the default.
domain-name *domain-name*	Fully qualified domain name (FQDN). The FQDN provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.
email *email-address*	E-mail address of the certificate holder.
ip-address *ip-address*	IP address of the router.
ipv6-address *ipv6-address*	IPv6 address of the router for the alternate subject.
scep-digest-algorithm (md5 | sha-1)	Hash algorithm digest, either MD5 or SHA-1; SHA-1 is the default.
scep-encryption-algorithm (des | des3)	Encryption algorithm, either DES or DES3; DES3 is the default.
subject *subject-distinguished-name*	Distinguished Name (DN) format that contains the domain component, common name, department, serial number, company name, state, and country in the following format: DC, CN, OU, O, SN, L, ST, C. • `DC`—Domain component • `CN`—Common name • `OU`—Organizational unit name • `O`—Organization name • `SN`—Serial number of the device If you define SN in the subject field without the serial number, then the serial number is read directly from the device and added to the certificate signing request (CSR).

- ST—State
- C—Country

Required Privilege Level

maintenance and security

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

command-name

```
user@host> request security pki local-certificate enroll scep certificate-id r3-entrust-scep ca-
profile entrust domain-name router3.example.net subject
"CN=router3,OU=Engineering,O=example,C=US" challenge-password 123
```

```
Certificate enrollment has started. To view the status of your enrollment, check the public key
infrastructure log (pkid) log file at /var/log/pkid. Please save the challenge-password for
revoking this certificate in future.  Note that this password is not stored on the router.
```

Sample Output

Sample output for vSRX 3.0

```
user@host> request security pki generate-key-pair certificate-id example
```

```
Generated key pair example, key size 2048 bits
```

```
user@host> request security pki local-certificate enroll certificate-id ?
```

```
Possible completions:
<certificate-id> Certificate identifier
example
```

```
user@host> request security pki generate-key-pair certificate-id Hub
```

```
error: Failed to generate key pair at HSM. Found a key with the same name at HSM. Use a
different certificate id next time. Refer to PKID logs for more details
```

SEE ALSO

- request security pki local-certificate enroll cmpv2
- show security pki local-certificate (View)
- clear security pki local-certificate (Device)

RELATED DOCUMENTATION

- What is Azure Key Vault?

Example: Configure an IPsec VPN Between Two vSRX Instances

IN THIS SECTION

This example shows how to configure an IPsec VPN between two instances of vSRX in Microsoft Azure.

Before You Begin

Ensure that you have installed and launched a vSRX instance in Microsoft Azure virtual network.

See SRX Site-to-Site VPN Configuration Generator and How to troubleshoot a VPN tunnel that is down or not active for additional information.

Overview

You can use an IPsec VPN to secure traffic between two VNETs in Microsoft Azure using two vSRX instances.

vSRX IPsec VPN Configuration

IN THIS SECTION

vSRX1 VPN Configuration

Step-by-Step Procedure

To configure IPsec VPN on vSRX1:

1. Log in to the vSRX1 in configuration edit mode (see *Configure vSRX Using the CLI*).

2. Set the IP addresses for vSRX1 interfaces.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.10/24
set interfaces ge-0/0/1 unit 0 family inet address 10.10.10.10/24
set interfaces st0 unit 1 family inet address 10.0.250.10/24
```

3. Set up the untrust security zone.

```
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces st0.1
```

4. Set up the trust security zone.

```
set security zone trust host-inbound-traffic system-services https
set security zone trust host-inbound-traffic system-services ssh
set security zone trust host-inbound-traffic system-services ping
set security security-zone trust interfaces ge-0/0/1.0
```

5. Configure IKE.

```
set security ike proposal ike-phase1-proposalA authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposalA dh-group group2
set security ike proposal ike-phase1-proposalA authentication-algorithm sha-256
set security ike proposal ike-phase1-proposalA encryption-algorithm aes-256-cbc
set security ike proposal ike-phase1-proposalA lifetime-seconds 1800
set security ike policy ike-phase1-policyA mode aggressive
set security ike policy ike-phase1-policyA proposals ike-phase1-proposalA
set security ike policy ike-phase1-policyA pre-shared-key ascii-text <preshared-key>
set security ike gateway gw-siteB ike-policy ike-phase1-policyA
set security ike gateway gw-siteB address 198.51.100.10
set security ike gateway gw-siteB local-identity user-at-hostname "source@example.net"
set security ike gateway gw-siteB remote-identity user-at-hostname "dest@example.net"
set security ike gateway gw-siteB external-interface ge-0/0/0.0
```

NOTE: Be sure to replace `198.51.100.10` in this example with the correct public IP address.

6. Configure IPsec.

```
set security ipsec proposal ipsec-proposalA protocol esp
set security ipsec proposal ipsec-proposalA authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-proposalA encryption-algorithm aes-256-cbc
set security ipsec policy ipsec-policy-siteB proposals ipsec-proposalA
set security ipsec vpn ike-vpn-siteB bind-interface st0.1
set security ipsec vpn ike-vpn-siteB ike gateway gw-siteB
set security ipsec vpn ike-vpn-siteB ike ipsec-policy ike-phase1-policyA
set security ipsec vpn ike-vpn-siteB establish-tunnels immediately
```

7. Configure routing.

```
set routing-instances siteA-vr1 instance-type virtual-router
set routing-instances siteA-vr1 interface ge-0/0/0.0
set routing-instances siteA-vr1 interface ge-0/0/1.0
set routing-instances siteA-vr1 interface st0.1
set routing-instances siteA-vr1 routing-options static route 0.0.0.0/0 next-hop 10.0.0.1
set routing-instances siteA-vr1 routing-options static route 10.20.20.0/24 next-hop st0.1
commit
```

vSRX2 VPN Configuration

Step-by-Step Procedure

To configure IPsec VPN on vSRX2:

1. Log in to the vSRX2 in configuration edit mode (See *Configure vSRX Using the CLI*.

2. Set the IP addresses for the vSRX2 interfaces.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.0.10/24
set interfaces ge-0/0/1 unit 0 family inet address 10.20.20.10/24
set interfaces st0 unit 1 family inet address 10.0.250.20/24
```

3. Set up the untrust security zone.

```
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces st0.1
```

4. Set up the trust security zone.

```
set security zones security-zone trust host-inbound-traffic system-services https
set security zones security-zone trust host-inbound-traffic system-services ssh
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
```

5. Configure IKE.

```
set security ike proposal ike-phase1-proposalA authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposalA dh-group group2
set security ike proposal ike-phase1-proposalA authentication-algorithm sha-256
set security ike proposal ike-phase1-proposalA encryption-algorithm aes-256-cbc
set security ike proposal ike-phase1-proposalA lifetime-seconds 1800
set security ike policy ike-phase1-policyA mode aggressive
set security ike policy ike-phase1-policyA proposals ike-phase1-proposalA
set security ike policy ike-phase1-policyA pre-shared-key ascii-text preshared-key
set security ike gateway gw-siteB ike-policy ike-phase1-policyA
set security ike gateway gw-siteB address 203.0.113.10
set security ike gateway gw-siteB local-identity user-at-hostname "dest@example.net"
set security ike gateway gw-siteB remote-identity user-at-hostname "source@example.net"
set security ike gateway gw-siteB external-interface ge-0/0/0.0
```

NOTE: Be sure to replace `203.0.113.10` in this example with the correct public IP address. Also note that the SiteB local-identity and remote-identity should be in contrast with the SiteA local-identity and remote-identity.

6. Configure IPsec.

```
set security ipsec proposal ipsec-proposalA protocol esp
set security ipsec proposal ipsec-proposalA authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-proposalA encryption-algorithm aes-256-cbc
set security ipsec policy ipsec-policy-siteB proposals ipsec-proposalA
set security ipsec vpn ike-vpn-siteB bind-interface st0.1
set security ipsec vpn ike-vpn-siteB ike gateway gw-siteB
set security ipsec vpn ike-vpn-siteB ike ipsec-policy ike-phase1-policyA
set security ipsec vpn ike-vpn-siteB establish-tunnels immediately
```

7. Configure routing.

```
set routing-instances siteA-vr1 instance-type virtual-router
set routing-instances siteA-vr1 interface ge-0/0/0.0
set routing-instances siteA-vr1 interface ge-0/0/1.0
set routing-instances siteA-vr1 interface st0.1
set routing-instances siteA-vr1 routing-options static route 0.0.0.0/0 next-hop 10.0.0.1
set routing-instances siteA-vr1 routing-options static route 10.20.20.0/24 next-hop st0.1
commit
```

Verification

IN THIS SECTION

Verify Active VPN Tunnels

Purpose

Verify that the tunnel is up on both vSRX instances.

Action

```
root@> show security ipsec security-associations
```

```
Total active tunnels: 1
ID      Algorithm                SPI      Life:sec/kb    Mon lsys Port  Gateway
<131074 ESP:aes--cbc--256/sha1 de836105 1504/ unlim -- root 4500 52.200.89.XXX
>131074 ESP:aes--cbc--256/sha1 b349bc84 1504/ unlim -- root 4500 52.200.89.XXX
```

RELATED DOCUMENTATION

- IPsec VPN Overview
- Application Firewall Overview

Example: Configure an IPsec VPN Between a vSRX and Virtual Network Gateway in Microsoft Azure

IN THIS SECTION

This example shows how to configure an IPsec VPN between a vSRX instance and a virtual network gateway in Microsoft Azure.

Before You Begin

Ensure that you have installed and launched a vSRX instance in Microsoft Azure virtual network.

See SRX Site-to-Site VPN Configuration Generator and How to troubleshoot a VPN tunnel that is down or not active for additional information.

Overview

You can use an IPsec VPN to secure traffic between two VNETs in Microsoft Azure, with one vSRX protecting one VNet and the Azure virtual network gateway protecting the other VNet.

vSRX IPsec VPN Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure IPsec VPN on vSRX:

1. Log in to the vSRX in configuration edit mode (see *Configure vSRX Using the CLI*).
2. Set the IP addresses for vSRX interfaces.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.10/24
set interfaces ge-0/0/1 unit 0 family inet address 10.10.10.10/24
set interfaces st0 unit 1 family inet address 10.0.250.10/24
```

3. Set up the untrust security zone.

```
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces st0.1
```

4. Set up the trust security zone.

```
set security zone trust host-inbound-traffic system-services https
set security zone trust host-inbound-traffic system-services ssh
```

```
set security zone trust host-inbound-traffic system-services ping
set security security-zone trust interfaces ge-0/0/1.0
```

5. Configure IKE.

```
set security ike proposal ike-phase1-proposalA authentication-method pre-shared-keys
set security ike proposal ike-phase1-proposalA dh-group group2
set security ike proposal ike-phase1-proposalA authentication-algorithm sha-256
set security ike proposal ike-phase1-proposalA encryption-algorithm aes-256-cbc
set security ike policy ike-phase1-policyA mode main
set security ike policy ike-phase1-policyA proposals ike-phase1-proposalA
set security ike policy ike-phase1-policyA pre-shared-key ascii-text <preshared-key>
set security ike gateway gw-siteB ike-policy ike-phase1-policyA
set security ike gateway gw-siteB address 52.175.210.65
set security ike gateway gw-siteB version v2-only
set security ike gateway gw-siteB external-interface ge-0/0/0.0
```

NOTE: Be sure to replace `52.175.210.65` in this example with the correct public IP address.

6. Configure IPsec.

The following example illustrates a vSRX IPsec configuration using the CBC encryption algorithm:

```
set security ipsec proposal ipsec-proposalA protocol esp
set security ipsec proposal ipsec-proposalA authentication-algorithm hmac-sha1-96
set security ipsec proposal ipsec-proposalA encryption-algorithm aes-256-cbc
set security ipsec proposal ipsec-proposalA lifetime-seconds 7200
set security ipsec proposal ipsec-proposalA lifetime-kilobytes 102400000
set security ipsec policy ipsec-policy-siteB proposals ipsec-proposalA
set security ipsec vpn ike-vpn-siteB bind-interface st0.1
set security ipsec vpn ike-vpn-siteB ike gateway gw-siteB
set security ipsec vpn ike-vpn-siteB ike ipsec-policy ike-phase1-policyA
set security ipsec vpn ike-vpn-siteB establish-tunnels immediately
```

If required, you can use AES-GCM as the encryption algorithm in the vSRX IPsec configuration instead of CBC:

```
set security ipsec proposal ipsec-proposalA protocol esp
set security ipsec proposal ipsec-proposalA encryption-algorithm aes-256-gcm
```

```
set security ipsec proposal ipsec-proposalA lifetime-seconds 7200
set security ipsec proposal ipsec-proposalA lifetime-kilobytes 102400000
set security ipsec policy ipsec-policy-siteB proposals ipsec-proposalA
set security ipsec vpn ike-vpn-siteB bind-interface st0.1
set security ipsec vpn ike-vpn-siteB ike gateway gw-siteB
set security ipsec vpn ike-vpn-siteB ike ipsec-policy ike-phase1-policyA
set security ipsec vpn ike-vpn-siteB establish-tunnels immediately
```

7. Configure routing.

```
set routing-instances siteA-vr1 instance-type virtual-router
set routing-instances siteA-vr1 interface ge-0/0/0.0
set routing-instances siteA-vr1 interface ge-0/0/1.0
set routing-instances siteA-vr1 interface st0.1
set routing-instances siteA-vr1 routing-options static route 0.0.0.0/0 next-hop 10.0.0.1
set routing-instances siteA-vr1 routing-options static route 10.20.20.0/24 next-hop st0.1
commit
```

Microsoft Azure Virtual Network Gateway Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

1. To configure the Microsoft Azure virtual network gateway, refer to the following Microsoft Azure procedure:

   Configure IPsec/IKE policy for S2S VPN or VNet-to-VNet connections

   Ensure the IPSec IKE parameters in Microsoft Azure virtual network gateway match the vSRX IPSec IKE parameters when the site-to-site VPN connection is formed.

2. Verify Active VPN Tunnels.

Verify that the tunnel is up between the vSRX instance and the Azure virtual network gateway.

```
root@> show security ike security-associations
```

```
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
8290401 UP     b1adf15fc3dfe0b0  89cc2a12cb7e3cd7  IKEv2          52.175.210.65
```

```
root@> show security ipsec security-associations
```

```
Total active tunnels: 1
  ID    Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
  <131073 ESP:aes-gcm-256/None c0e154e2 5567/  102399997 - root 4500 52.175.210.65
  >131073 ESP:aes-gcm-256/None 383bd606 5567/  102399997 - root 4500 52.175.210.65
```

RELATED DOCUMENTATION

- IPsec VPN Overview
- Application Firewall Overview

Example: Configure Juniper Sky ATP for vSRX

IN THIS SECTION

This example shows how to configure Juniper Sky™ Advanced Threat Prevention (Juniper Sky ATP) on a vSRX instance that is deployed in a virtual private cloud (VPC).

Before You Begin

Ensure that you have installed and launched a vSRX instance in a VPC.

Overview

You can use Juniper Sky ATP, a cloud-based solution, along with vSRX to protect all hosts in your network against evolving security threats.

Juniper Sky ATP Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure Juniper Sky ATP on a vSRX instance:

1. Log in to the vSRX instance using SSH and start the CLI.

```
root@% cli
root@>
```

2. Enter configuration mode.

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

3. Set up the correct data interface for the active advanced antimalware (AAMW) service instead of using the default fxp0 interface.

```
root@# set services advanced-anti-malware connection source-interface ge-0/0/0.0
```

4. Configure NAT.

```
root@# set security nat source rule-set rs1 from zone trust
root@# set security nat source rule-set rs1 to zone untrust
root@# set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
root@# set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
root@# set security nat source rule-set rs1 rule r1 then source-nat interface
```

5. Set up virtual routing instance for the correct data interface for AAMW service.

```
root@# set routing-instances vsrx-vr1 instance-type virtual-router
root@# set routing-instances vsrx-vr1 routing-options static route 0.0.0.0/0 next-hop 10.4.1.1
root@# set routing-instances vsrx-vr1 interface ge-0/0/0.0
root@# set routing-instances vsrx-vr1 interface ge-0/0/1.0
```

6. Verify the configuration.

```
root@# commit check
configuration check succeeds
```

7. Commit the configuration to activate it on the vSRX instance.

```
root@# commit
commit complete
```

8. Optionally, you can verify the configuration by running the following show commands in the configuration mode:

   - show services advanced-anti-malware connection | display set
   - show security nat | display set
   - show routing-instances vsrx-vr1 | display set

RELATED DOCUMENTATION

Juniper Sky Advanced Threat Prevention Administration Guide

PART 8

vSRX Deployment for Google Cloud Platform

CHAPTER 25

Overview

IN THIS CHAPTER

Understand vSRX Deployment with Google Cloud

IN THIS SECTION

Understand vSRX Deployment with Google Cloud Platform

IN THIS SECTION

Google Cloud Platform (GCP) is a public cloud service provided by Google. Like Amazon Web Service (AWS) and Microsoft Azure, GCP offers a suite of products and services that allow you to build and host applications and websites, store data, and analyze data on Google's scalable infrastructure. A pay-as-you-go model is delivered and saves you from building your own private cloud using dedicated hardware.

Google's virtual private cloud (VPC) gives you the flexibility to scale and control how workloads connect regionally and globally. When you connect your on-premises or remote resources to GCP, you will have

global access to your VPCs without needing to replicate connectivity or administrative policies in each region.

vSRX in a public cloud can be used for protecting service VMs from public Internet or protecting VMs in different subnets, or used as VPN Gateways.

Like AWS, GCP allows you to build your own VPCs on top of Google's public infrastructure. Unlike AWS, GCP uses KVM instead of modified Xen as the hypervisor for VM management.

In a Google cloud, vSRX instances run on top of Google VPCs. A Google VPC has the following properties:

- Provides a global private communication space.
- Supports multitenancy in an organization.
- Provides private communication between Google Cloud Platform (GCP) resources, such as Computing Engine and Cloud Storage.
- Provides security for configuration access using identify and access management (IAM).
- Extensible across hybrid environments.

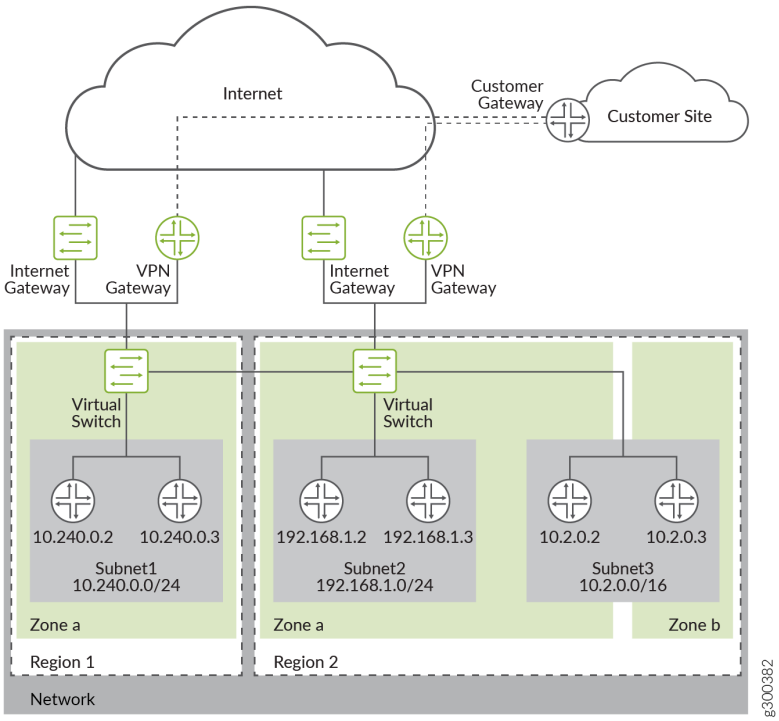
When you create a resource in GCP, you choose a network and subnet. For resources other than instance templates, you also select a zone or a region. Selecting a zone implicitly selects its parent region. Because subnets are regional objects, the region you select for a resource determines the subnets it can use.

The process of creating an instance involves selecting a zone, a network, and a subnet. The subnets available for selection are restricted to those in the selected region. GCP assigns the instance an IP address from the range of available addresses in the subnet.

The process of creating a managed instance group involves selecting a zone or region, depending on the group type, and an instance template. The instance templates available for selection are restricted to those whose defined subnets are in the same region selected for the managed instance group. Instance templates are global resources. The process of creating an instance template involves selecting a network and a subnet. If you select an auto-mode network, you can choose "auto subnet" to defer subnet selection to one that is available in the selected region of any managed instance group that would use the template, because auto-mode networks have a subnet in every region by definition.

An example of a typical Google VPC is shown in Figure 117 on page 488.

Figure 117: Example of a Google VPC



The vSRX instance is launched with multiple virtual interfaces in VPC subnets. The first interface (fxp0) will be the management interface. It is connected to the Internet gateway for public access. You can use SSH to access the interface and manage the virtual device with Junos CLI, just as you can with SRX Series devices. The subsequent interfaces are revenue ports. They are managed by the flowd process running on Linux and handle all the traffic. On GCP, a maximum of 8 network interfaces are allowed per vSRX instance.

Some of the initial provisioning parameters for first boot are host name, root password, SSH public key, management interface (fxp0) IP address, and default gateway IP address.

Starting in Junos OS Release 19.2R1, vSRX instances with 2 vCPUs, 4-GB memory, and 19-GB disk space are supported on GCP.

Manage Access to Instances

To create and manage instances, you can use a variety of tools, including the Google Cloud Platform Console, the gcloud command-line tool, and the REST API. To configure applications on your instances, connect to the instance using SSH for Linux instances.

You can manage access to your instances using one of the following methods:

- **Linux instance:**
  - Manage instance access using OS login, which allows you to associate SSH keys with your Google account or G Suite account and manage administrator or non-administrator access to instances through identity and access management (IAM) roles. If you connect to your instances using the gcloud command-line tool or SSH from the console, Compute Engine can automatically generate SSH keys for you and apply them to your Google account or G Suite account.
  - Manage your SSH keys in project or instance metadata, which grants administrator access to instances with metadata access that do not use OS Login. If you connect to your instances using the gcloud command-line tool or SSH from the console, Compute Engine can automatically generate SSH keys for you and apply them to project metadata.
- **Windows Server instances**—Create a password for a Windows Server instance.

Access Instances

After you configure access to your instances, you can connect to your instances using one of several options. For more information about connecting your instances, see Connecting to instances.

Requirements for vSRX on Google Cloud Platform

IN THIS SECTION

Google Compute Engine Instance Types

To create a vSRX instance, you need to choose a machine type. The machine type specifies a particular collection of virtualized hardware resources available to a VM instance, including the memory size, vCPU count, and maximum disk capacity.

Google Compute Engine allows you to use predefined machine or instances types or customized machine or instance types based on your needs. Table 83 on page 490 below shows the predefined machine types available in Google Compute Engine.

Table 83: Google Compute Engine Instance Types

Machine Name	Description	vCPUs	Memory (GB)	vSRX 3.0 Instance	Maximum number of Persistent Disks	Maximum total Persistent Disk Size (TB)	RSS Type
n1-standard-4	Standard machine type with 4 vCPUs and 15 GB of memory	4	15	VSRX-4CPU-15G memory	16	64	SWRSS
n1-standard-8	Standard machine type with 8 vCPUs and 30 GB of memory	8	30	VSRX-8CPU-30G memory	16	64	SWRSS
n1-standard-16	Standard machine type with 16 vCPUs and 60 GB of memory	16	60	VSRX-16CPU-60G memory	16	64	SWRSS

A single Google Compute Engine instance supports up to eight network interfaces. If you want to configure eight interfaces, choose n1-standard-8 or a larger machine type. After choosing the machine type, define the networking attributes and SSH Keys for the VM. For more information on network interfaces, see Creating instances with multiple network interfaces.

vSRX Support for Google Cloud

Starting in Junos OS Release 19.2R1, vSRX with 1 Junos Control Plane (JCP) vCPU, 1 data plane vCPU, and 4 GB of vRAM is supported.

vSRX Specifications for GCP

IN THIS SECTION

This topic provides details about hardware and software requirements for deploying vSRX with Google.

Minimum System Requirements for Google Cloud Platform

Table 84 on page 491 lists the minimum system requirements and the Junos OS release in which a particular software specification was introduced for vSRX instances to be deployed on GCP.

Table 84: Minimum System Requirements for vSRX on GCP

Component	Specification	Release Introduced
Memory	4 GB	Junos OS Release 19.2R1
Disk space	19-GB IDE drive	Junos OS Release 19.2R1
vCPUs	1 Junos Control Plane (JCP) vCPU and 1 data plane vCPU	Junos OS Release 19.2R1
vNICs	2-8 vNICs • Virtio • SR-IOV is not supported by GCP.	Junos OS Release 19.2R1
Software feature license	For more information, see Flex Software Subscription Model and Juniper Flex Program Support for Juniper Products.	NA

Table 84: Minimum System Requirements for vSRX on GCP *(Continued)*

Component	Specification	Release Introduced
Software packaging	Google Compute Engine has specific requirements for the bootable image that is imported to Google cloud space. For more information, see https://cloud.google.com/compute/docs/ images/import-existing-image#create_ image_file. For initial deployment, the .img file is used and for software upgrade, the .tgz image is used.	NA

Interface Mapping for vSRX on Google Cloud

Each network adapter defined for a vSRX is mapped to a specific interface, depending on whether the vSRX instance is a standalone VM or one of a cluster pair for high availability. The interface names and mappings in vSRX are shown in Table 85 on page 492.

Note the following:

- In standalone mode:
  - fxp0 is the out-of-band management interface.
  - ge-0/0/0 is the first traffic (revenue) interface.

Table 85 on page 492 shows the interface names and mappings for a standalone vSRX on Google cloud.

Table 85: Interface Names for a Standalone vSRX on GCP

Network Adapter	Interface Name in Junos OS for vSRX
1	fxp0
2	ge-0/0/0
3	ge-0/0/1

Table 85: Interface Names for a Standalone vSRX on GCP *(Continued)*

Network Adapter	Interface Name in Junos OS for vSRX
4	ge-0/0/2
5	ge-0/0/3
6	ge-0/0/4
7	ge-0/0/5
8	ge-0/0/6

vSRX Default Settings on GCP

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.

Table 86 on page 493 lists the factory-default settings for security policies on the vSRX instance.

Table 86: Factory-Default Settings for Security Policies

Source Zone	Destination Zone	Policy Action
trust	untrust	permit
trust	trust	permit
untrust	trust	deny

CHAPTER 26

Install vSRX in Google Cloud

IN THIS CHAPTER

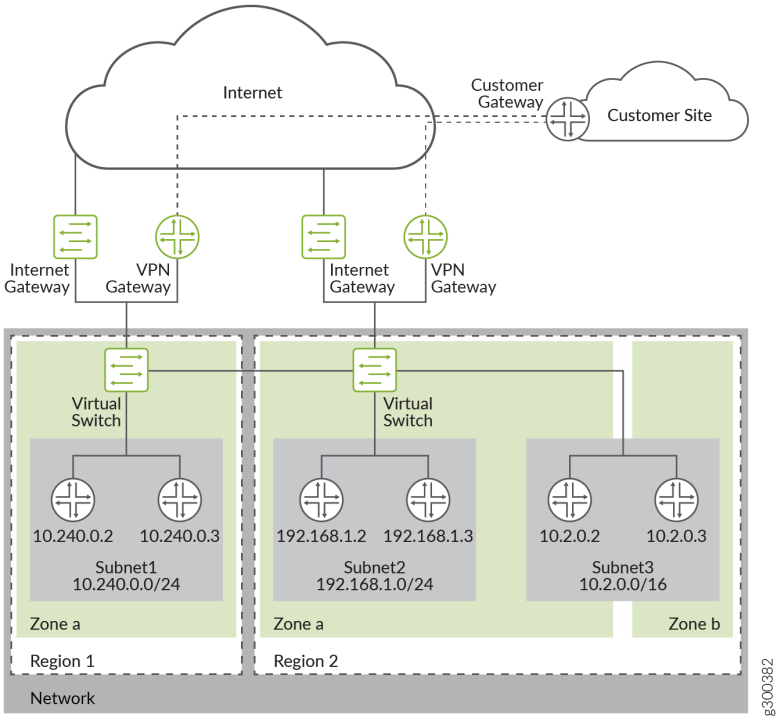
Prepare to setup vSRX Deployment on GCP

IN THIS SECTION

Before you begin, you need a Google account and an identity and access management (IAM) role, with all required permissions to access, create, modify, and delete Compute Engine Instances and Storage Service, and Google's VPC objects. You should also create access keys and corresponding secret access keys, certificates, and account identifiers.

Figure 118 on page 495 shows an example of how you can deploy vSRX to provide security for applications running in a private subnet of Google VPC.

Figure 118: Example of a Google VPC



You need to set up the vSRX 3.0 Firewall on Google Cloud Platform to deploy a vSRX 3.0 firewall on a Google Cloud Computer Engine instance on the Google Cloud Platform (GCP).

Before you deploy vSRX 3.0, you must create your project networks and subnetworks, and plan networks and IP address assignments for the vSRX interfaces. During the deployment, you must choose from the existing networks and subnetworks.

Subnetworks—You must create subnetworks in each VPC networks in specific region in which you plan to deploy the vSRX. A VPC Networks can add subnetworks in different region. These subnetworks are all internal network in GCP.

- **IP Address**—You need to assign IP address ranges when you create interface subnetworks.
- **Range**—The range for a network subnet cannot overlap with others.
- **External IP Address**—During vSRX deployment you can choose to enable or disable an external IP address when you create a network interface for the vSRX, by default, an ephemeral IP address is auto-assigned. You can also specify a static address when creating a network interface.

- **Management Interface**—The first network interface added to a vSRX is mapped to fxp0 on the vSRX.
  - Enable IP forwarding
  - This interface has an external IP address.
  - On vSRX, DHCP is enabled to fxp0 by default.
  - You can change the ephemeral IP address given during deployment to a static IP address, after you complete the deployment.
- **Interface Order**—First network interface is mapped to fxp0, second network interface is mapped to ge-0/0/0, 3rd network interface is mapped to ge-0/0/1.
- **Number of vSRX Interfaces**
  - The maximum number of virtual interfaces allowed per vSRX instance is 8.
  - To create a vSRX instance, you have to specify the machine type. The machine type specifies a particular collection of virtualized hardware resources available to a VM instance, including the memory size, virtual CPU count, and maximum disk capacity.
  - **Default VPC Network**—There is default network in a GCP project, you can delete the default network if unused. By default, 5 networks in a project. You can request additional networks for your project.
  - **Firewall Rules**—You must create a GCP firewall rules to allows access for management connection.

Before you begin, ensure to have the following ready:

- Google Cloud Platform Account Planning
- SSH Key Pair
- Virtual Private Cloud (VPC) Network Planning

Step 1: Google Cloud Platform Account Planning

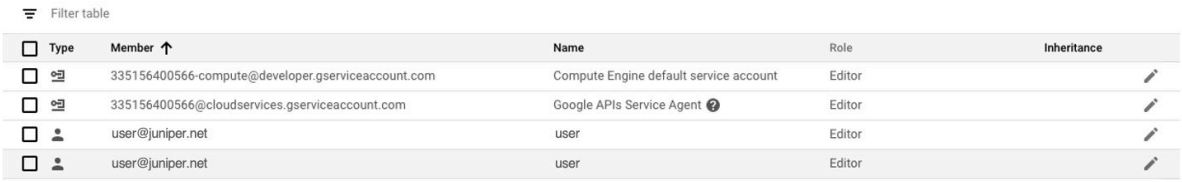
Before you begin deploying vSRX VM, review the licensing information and collect the information you'll need for the configuration process.

1. Understand your vSRX license requirements.
2. Determine private IP address for your management and other interfaces.
3. Get required permissions for the GCP account.
   - GCP user account with a linked e-mail address

- Identity and access management (IAM) roles as Compute Viewer, Storage Object Viewer,and Monitoring Metric Writer.

Accounts and Permissions—Ensure you have proper accounts and permissions before your deploy vSRX 3.0 on a Google Computer Engine instance. Sample account roles and IAM permissions are shown in Figure 119 on page 497

Figure 119: Sample Account Roles and IAM Permissions



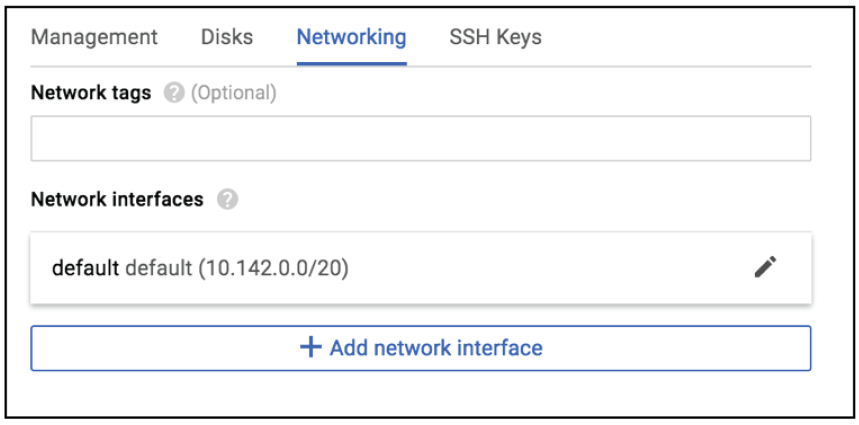
Step 2: Define Network Attributes and Generate SSH Key Pair for Authentication

The procedure below provides you steps to define network attributes and generate your own SSH Key pairs to allow your first time login:

1. After choosing the machine type, you must define networking attributes in the advanced options for the VM.

   Click the **VM instances** tab on the home page and then click the **Networking** tab as shown in Figure 120 on page 497. Update the networking attributes and add the required interfaces.

Figure 120: Define Network Attributes



You can add up to 8 interfaces for each vSRX instance.

NOTE: You cannot choose virtual interface type. GCP supports only the VirtIO interface type. SR-IOV is not supported in GCP.

2. vSRX manages authentication for first login only through RSA SSH key authentication. Password is not allowed, so you cannot log into vSRX through console on GCP web. Root login without password is not allowed. So you must generate your own SSH Key before your deploy a vSRX instance in Google Compute Engine.

   Generate the public key and the private key. Create an SSH key pair and store the SSH Key in the default location for your operation system.

   - If you are using Linux or MacOS: Use ssh-keygen to create the key pair in your .ssh directory. Run the **ssh-keygen -t rsa -f ~/.ssh/gcp-user-1 -C gcp-user** command. Here **gcp-user-1** is name of key file and **gcp-user** is username.

     NOTE: It is mandatory to use “gcp-user” as username when you login to the vSRX for the first time vSRX.

   - If you are using Windows: Use PuTTYgen to create the key pair.

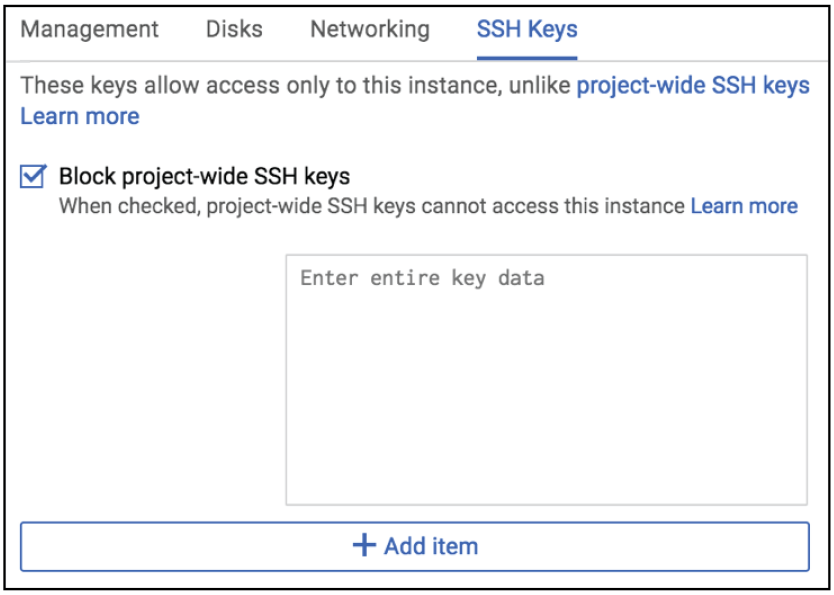
3. Copy your public key in a text editor. You need to paste it later while deploying vSRX in the GCP Marketplace.

4. Block project-wide SSH keys and specify an SSH key for each vSRX instance.

   Click the **SSH Keys** tab on the **VM instances** page as shown in Figure 121 on page 499.

NOTE: The SSH key is used by the public key authentication for the first login. As a security measurement, you must block project-wide SSH keys and specify an SSH key for each vSRX instance.

Figure 121: Block Project-Wide SSH Keys



5. Save your private key in .ppk format. You need this key later to authenticate the vSRX instance.

Step 3: Plan Google Virtual Private Cloud (VPC) Network

Prepare the virtual private cloud (VPC) networks in Google Cloud Platform. You must create virtual private networks, rules, and subnetworks and configure interfaces before you start deploying the vSRX on GCP which involves:

1. Log in to the Google Cloud console.
2. **VPC Networks**—You must create a custom network specifically for each vSRX network interface. In the left navigation area, click **VPC network** under **NETWORKING**.
3. On the top pane, click **CREATE VPC NETWORK**.
4. Enter a name for the network.
5. Create a subnetwork with the following details and click **Create**.
   - **Name**—Name of the subnetwork.

- **IP Address**—Assign an IP address range for creating interface subnetworks. This range is used for your internal network, so ensure that the address range does not overlap with other subnets.
- **Region**—Select the region where you want to launch your vSRX VM.
- **Private Google Access**—Retain the default value **Off**.
- **Flow logs**—Retain the default value **Off**.

Deploy vSRX in Google Cloud Platform

IN THIS SECTION

The following procedures describe how to deploy vSRX in the Google Virtual Private Cloud (VPC):

- Deploy the vSRX Firewall from Google Cloud Platform Marketplace.
- Use custom private image to deploy the vSRX Firewall from the GCP portal.
- Use cloud-init to deploy the vSRX Firewall through gcloud using CLI.

Deploy the vSRX Firewall from Marketplace Launcher

You can use the Google Cloud Platform Marketplace to deploy your vSRX3.0 with licenses as avirtual machine(VM) running on a Google Compute Engine instance.

Before you deploy the vSRX, you must create or choose a project in your organization and create any networks and subnets that will connect to the firewall. You cannot attach multiple network interfaces to the same VPC network. Every interface you create must have a dedicated network with at least one subnet.

This topic provide your step to deploy a vSRX Firewall from the Google Cloud Platform Marketplace Launcher.

1. Log in to the Google Cloud Platform console.
2. In the left navigation area, select **Marketplace**.

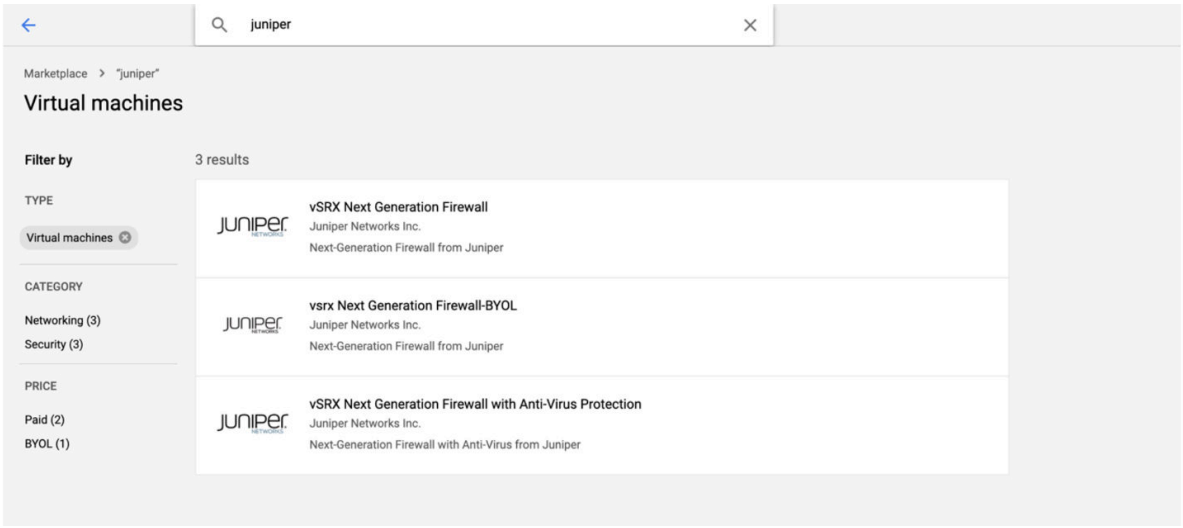
3. Locate the vSRX listing in the Marketplace.

   In the Search box, type 'Juniper' or 'vSRX' and click one of the following options based on your licensing requirements as shown in Figure 122 on page 501.

   The images are available from cloud:

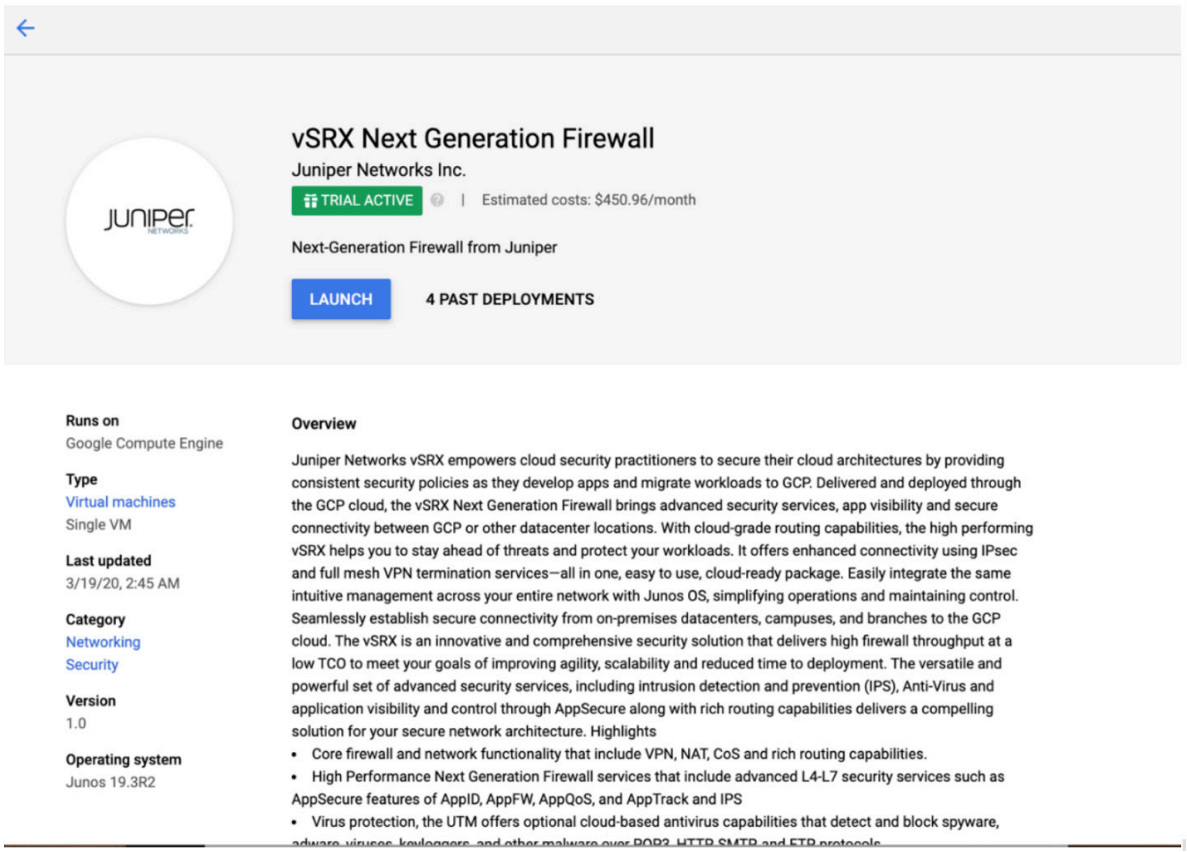
   - vSRX Next Generation Firewall
   - vsrx Next Generation Firewall-BYOL
   - vSRX Next Generation Firewall with Anti-Virus Protection

Figure 122: Locate vSRX Listing in the GCP Marketplace



4. Click **Launch** on Compute Engine. The deployment page appears as shown in Figure 123 on page 502.

Figure 123: Launch vSRX Instance in GCP from Marketplace



5. Name the instance and choose resources.

   Provide the details for the vSRX VM:

   - **Deployment Name**—Enter a unique name for your vSRX VM.
   - **Machine type**—Select a machine type based on the system requirements for your license.
   - **SSH key**—Paste your public SSH key that you created earlier.
     - Paste the key after the text gcp-user:

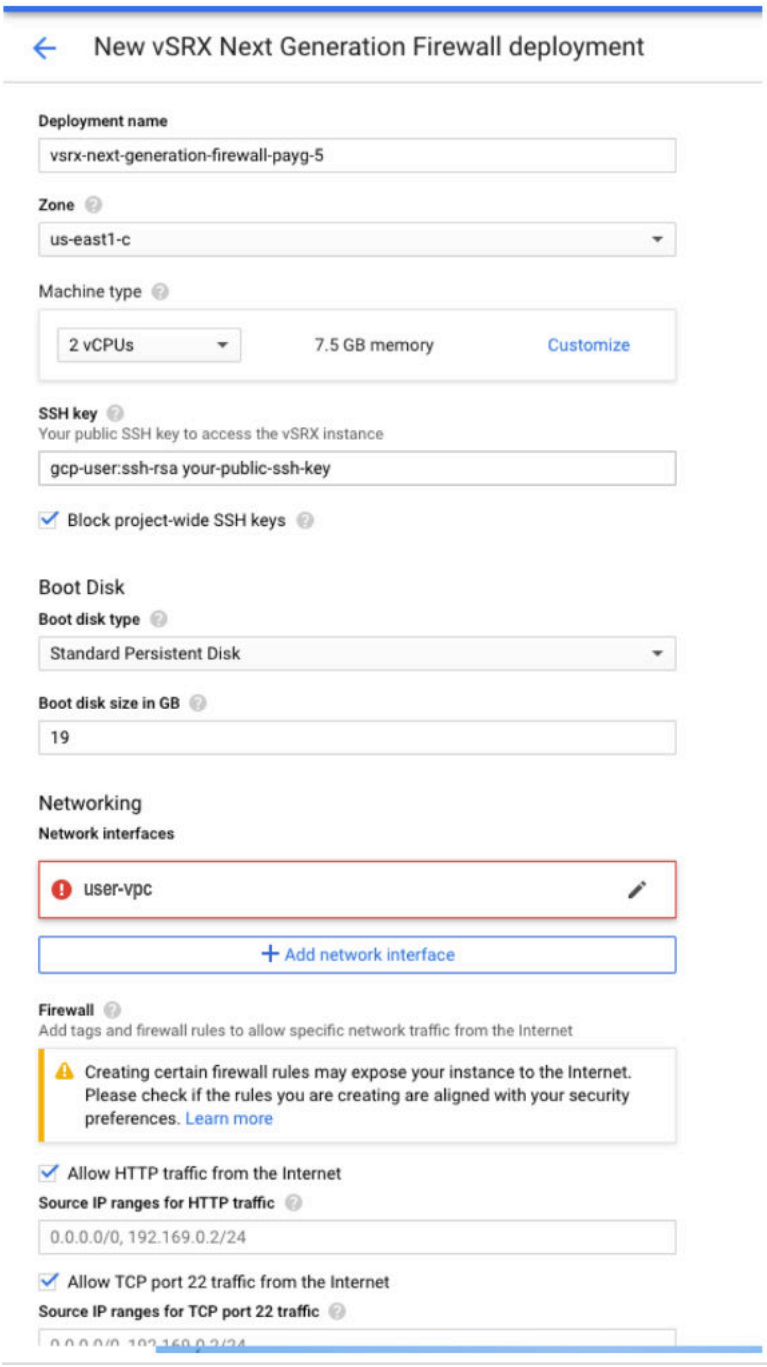
       NOTE: It is mandatory to use “gcp-user” as username when you login to the vSRX for the first time vSRX.

     - Select the Block project-wide SSH keys option.

- **Network interfaces**—Select the VPC network and the subnets. Note that you can add only those subnets that you've created for the selected zone for this vSRX VM.
- **IP Forwarding**—Retain the default value On. This is a mandatory requirement for the vSRX VM.
- **Enable External IP**—Select the ephemeral option. This setting allows the GCP to provide an ephemeral IP address to act as the external IP address.
- **Allow HTTP traffic from the Internet**—Retain the default value as selected. We recommend not providing HTTP access unless absolutely necessary.
- **Allow TCP port 22 traffic from the Internet**—Retain the default value as selected. F or security reasons, we recommend that you limit the SSH access only to the specific IP address to access the vSRX

Name the instance and choose resources as shown in Figure 124 on page 504.

Figure 124: Name vSRX Instance and Choose Resources in GCP Marketplace



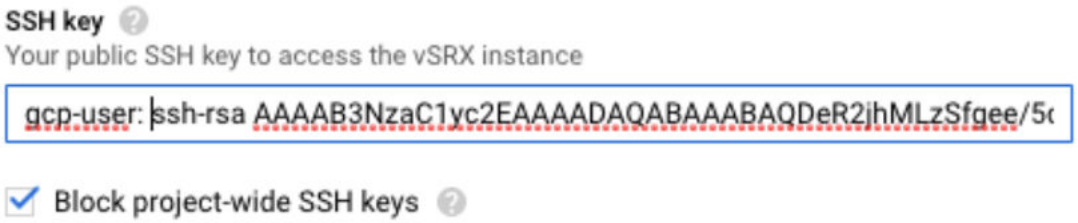
a. Choose a **Deployment Name**. The name must be unique and cannot conflict with any other deployment in the project.

b. Select a zone.

c. Select a machine type.

d. Set the SSH Key as shown in Figure 125 on page 505.

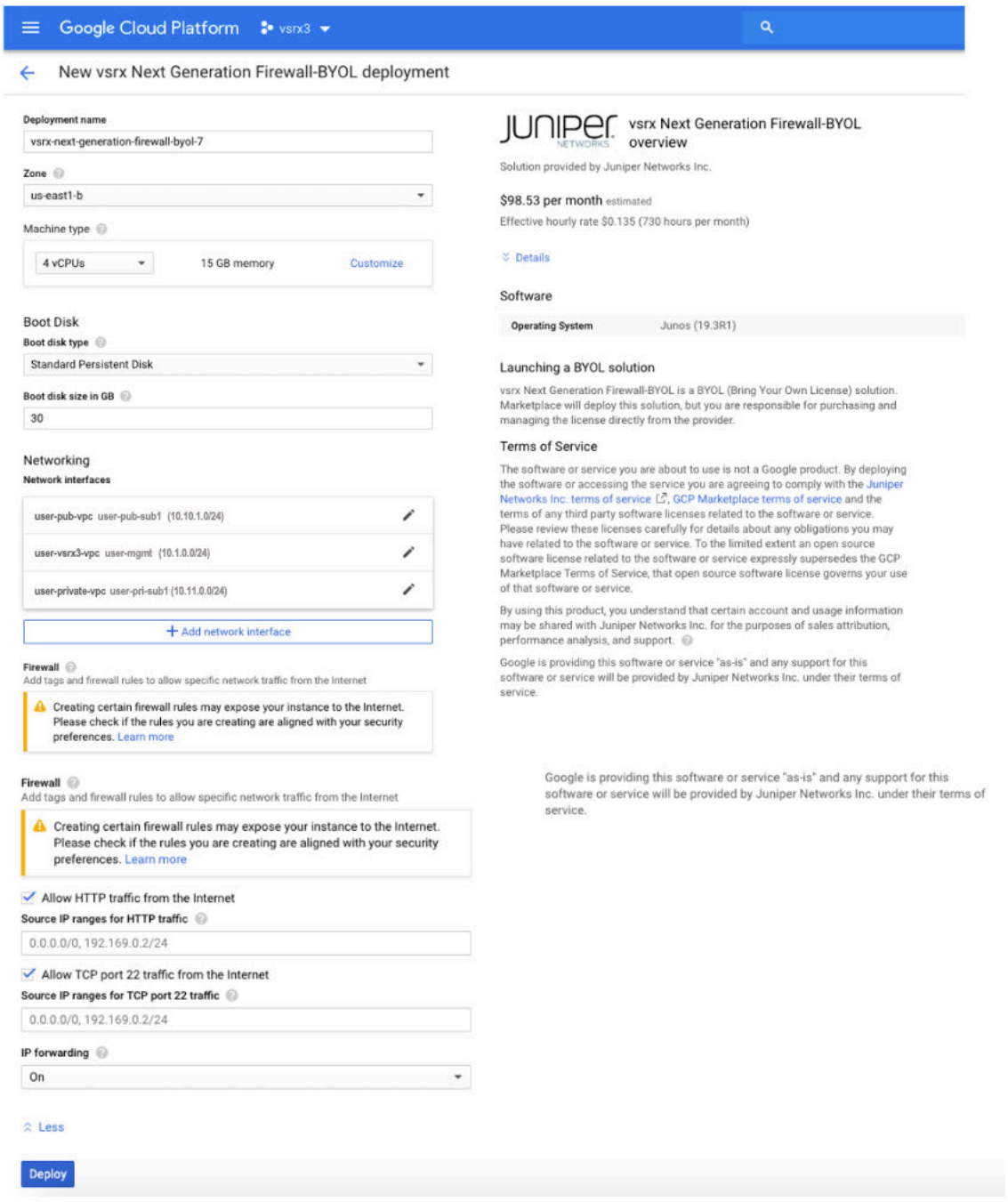
Figure 125: SSH Key



e. Configure the network and subnet.

f. Leave **IP forwarding** 'on' (mandatory for vSRX deployments) as shown in Figure 126 on page 506.

Figure 126: IP Forwarding Configuration



6. Accept GCP Marketplace **Terms of Service**.
7. Click **Deploy**.

The system shows the progress of your vSRX deployment. It displays a message indicating the successful completion of the deployment and sends you an e-mail notification for the same.

8. Click your VM to view the details. You can view your VM details by navigating to the Compute Engine under **COMPUTE** in the left navigation area.

   Make note of the external IP address, shown under Network interfaces. You'll need this address later to log on to your vSRX instance using the CLI.

9. Logging in to a vSRX Instance.

   In GCP deployments, vSRX instances provide the following capabilities by default to enhance security:

   - Allows you to login only through SSH.
   - cloud-init is used to setup SSH key login.
   - SSH password login is disabled for root account.

   > **NOTE**: Root login using SSH password is be disabled by default.

   Use an SSH client to log in to a vSRX instance for the first time. To log in, specify the location where you saved the SSH key pair file for the user account, and the IP address assigned to the vSRX management interface (fxp0).

   > **NOTE**: Root login using a Junos OS password is disabled by default. You can configure other users after the initial Junos OS setup phase.

   If you do not have the key pair filename and the IP address, use these steps to view the key pair name and IP for a vSRX instance:

   a. In the GCP portal, select **Instances**.

   b. Select the vSRX instance, and select **eth0** in the Description tab to view the IP address for the fxp0 management interface.

   c. Click **Connect** above the list of instances to view the SSH key pair filename.

   To configure the basic settings for the vSRX instance, see *Configure vSRX Using the CLI*.

   > **NOTE**: gcloud connect to vSRX is not supported. Always use ssh with user provided key to connect to vSRX after instance is up.

Deploy the vSRX Instance from GCP Portal Using Custom Private Image

IN THIS SECTION

You can also use your custom private image to deploy the vSRX instead of deploying an image from GCP marketplace. Firstly you need upload the private image to Google Cloud storage, then create compute image in GCP, and then deploy vSRX on Google Compute Engine.

Watch the video Deploying vSRX Virtual Firewalls on Google Cloud Platform to understand how you can deploy vSRX instances from GCP.

Upload vSRX Image to Google Cloud Storage

To upload vSRX image to Google Cloud Storage:

1. Prepare the private vSRX image file.

   A custom image is a boot disk image that is private to you. To import a disk image to Google Compute Engine, the image file must meet the following requirements.

   - Disk image filename must be **disk.raw**.
   - RAW image file must have a size in an increment of 1 GB. For example, the file must be either 10 GB or 11 GB but not 10.5 GB.
   - Compressed file must be a .tar.gz file that uses gzip compression and the GNU tar format.

   To use .qcow2 vSRX image to generate .tar.gz file follow below steps to process the upload.

   a. Convert .qcow2 to "disk.raw" (disk.raw is the dedicate name for google cloud deployment).

   ```
   qemu-img convert -f qcow2 -O raw junos-vsrx3-x86-64-19.2I-20190115_dev_common.0.1057.qcow2 disk.raw
   ```

   b. Compress to .tgz file.

   ```
   tar -czf vsrx-0115.tar.gz disk.raw
   ```

2. Upload image to Google Cloud Storage. You can upload your custom private image in two ways:
   - Upload image through SDK shell
   - Upload image from Google Cloud Platform portal

Upload image through SDK shell:

Install Google Cloud SDK on Ubuntu.

You must install Google Cloud SDK on your operation system. below is the sample to install it on Ubuntu.

For more information on Google Cloud SDK installation on Ubuntu, see https://cloud.google.com/sdk/docs/quickstart-debian-ubuntu and for Gcloud command-line tool overview, see https://cloud.google.com/sdk/gcloud/.

To upload image through SDK shell:

1. Create google cloud storage.

   ```
   gs://vsrx-image
   ```

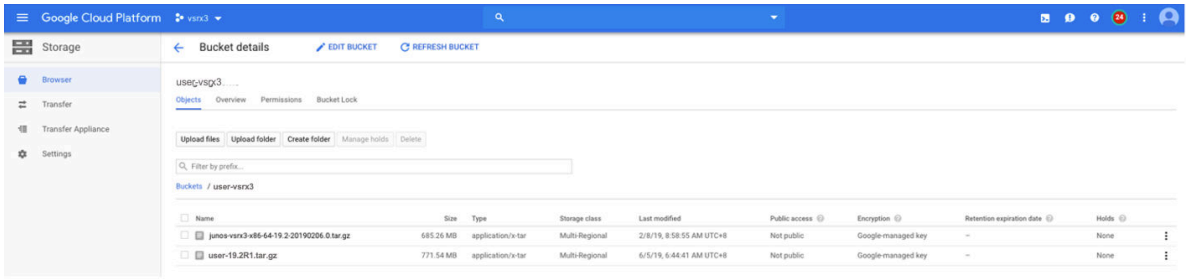
2. Copy disk.raw to cloud storage.

   ```
   gsutil cp vsrx-0115.tar.gz gs://vsrx-image
   ```

To upload image from Google Cloud Platform portal.

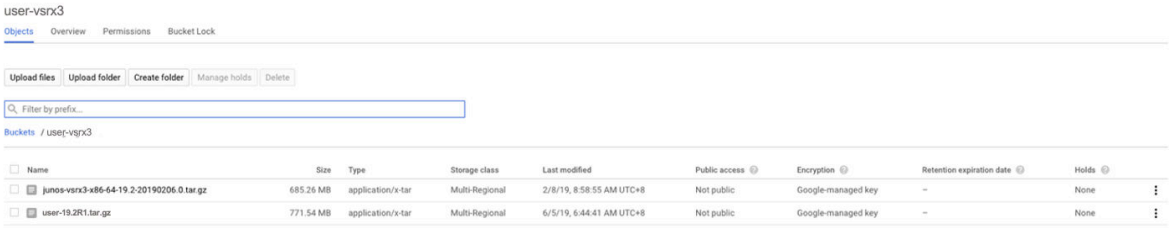
1. Click **Storage->Create Bucket->Upload files** as shown in Figure 127 on page 509.

Figure 127: vSRX Image Upload from GCP Portal



2. Check the private image is available in Google Cloud Storage by selecting **Storage -> Bucket detail** in Google Cloud Platform web as shown in Figure 128 on page 510.

Figure 128: View Private Images in GCP Portal



Create vSRX Image

After you upload the vSRX image file to GCP storage you need to create GCP compute image for vSRX deployment.

1. Create image in cloud.

   A sample to create vSRX image using the package ready in GCP project storage is shown below. The option of 'multi_ip_subnet' is mandatory.

   ```
   gcloud compute images create vsrx-0115 '--guest-os-features=multi_ip_subnet' --source-uri=gs://vsrx-image/vsrx-0115.tar.gz
   ```

2. Check the private image is available in Google Cloud Compute Engine.

   ```
   root@cnrd-ubuntu173:~# gcloud compute images list | grep vsrx3-194* vsrx-0115. vsrx3-218606 READY
   ```

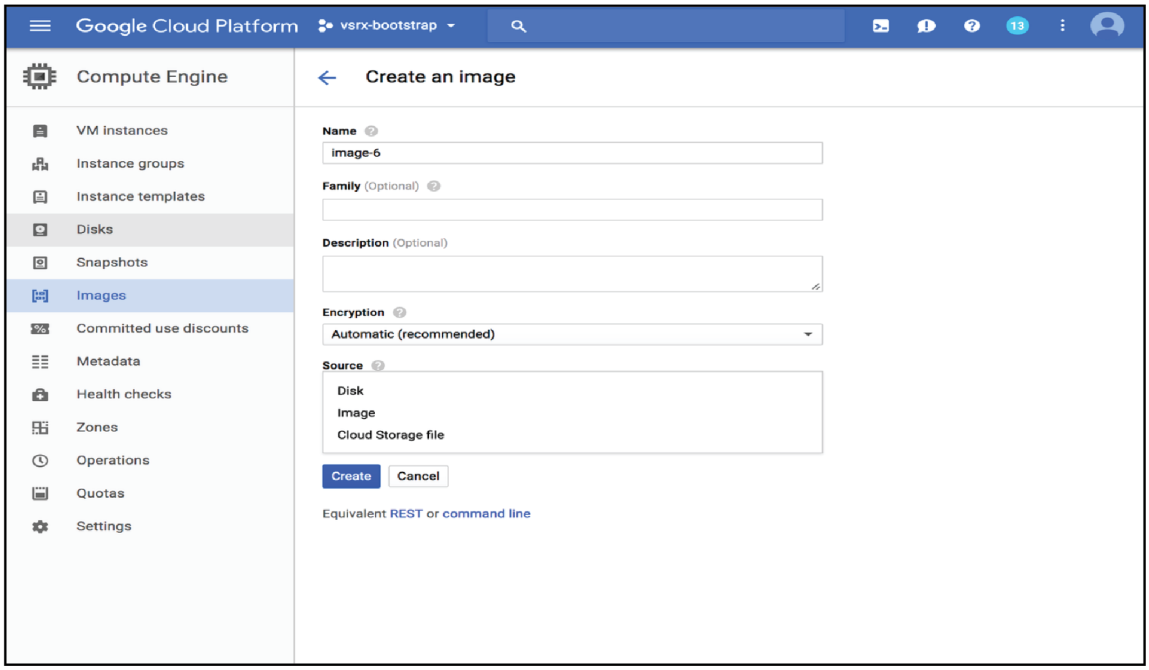
Using Google Console

You can rename the image file using the Google console as well.

1. Log in to your Google account and open the **Google Cloud Platform** home page.

2. Click the**images** option on the **Google Cloud Platform** page. The **Create an image** page opens as shown in Figure 129 on page 511

Figure 129: Google Cloud Platform Image Creation Page

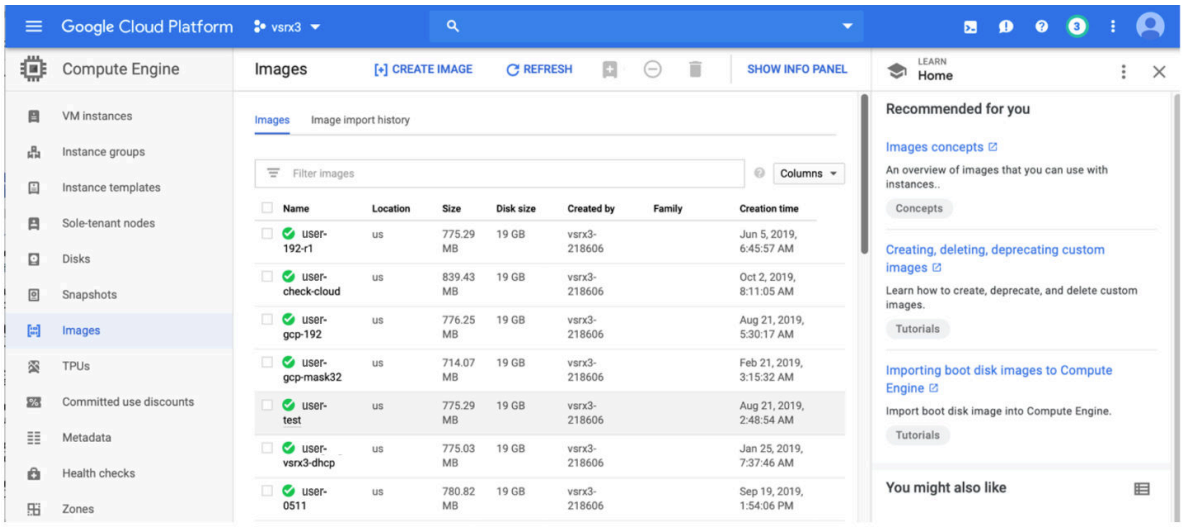


3. Fill in the required details in the **Create an image** page and click **Create**.

NOTE: It is mandatory to use “gcp-user” as username when you login to the vSRX for the first time vSRX.

4. Check the private image that available in Google Cloud Compute Engine. On Google Cloud Platform web, click **Compute Engine->Images** as shown in Figure 130 on page 512.

Figure 130: Check Private Image in Google Cloud Compute Engine



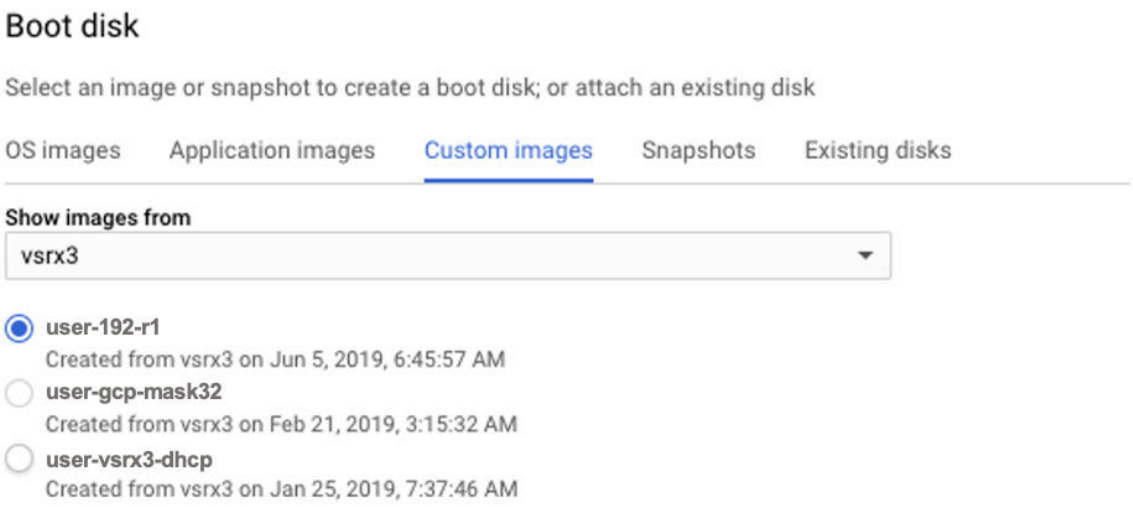
Deploy the vSRX Firewall from GCP Portal

You can follow below steps to deploy a vSRX instance:

1. Login Google Cloud Platform portal, go to **Compute Engine -> VM instances** and click **CREATE INSTANCE**.
2. Configure a vSRX instance.
   - **Name**—Specify a unique name to the instance.
   - **Region**—Select proper region you want to deploy the vSRX on, you must already create subnet in same region in proper VPC networks.
   - **Machine configuration** —Choose correct machine type.
   - **Container** —Uncheck

- **Boot Disk**—Choose the private image in **Custom Images** tab as shown in Figure 131 on page 513. You must already upload the private image to Google Cloud Storage.

Figure 131: Boot Disk from Custom Images



- **Identity and API access**—Set default
- **Firewall / Management** —Set default
- **Firewall / Security**—Paste your SSH Key pair here. Details please reference “Prepare to setup vSRX on GCP – SSH Key”.
- **Firewall / Disks**—Set default
- **Firewall / Networking**:

Table 87: Firewall Networking

Firewall / Networking	Details
Hostname	Optional, you can specify tags for the instance used for route configuration.
Network Interfaces	Default

	You can set interfaces to existing VPC networks and subnet in same region. Interface number, Interface order and manage interface setting.

3. Click **Create**
4. Logging in to a vSRX Instance.

   In GCP deployments, vSRX instances provide the following capabilities by default to enhance security:

   - Allows you to login only through SSH.
   - SSH password login is disabled for root account.

   NOTE: Root login using a Junos OS password or SSH password is disabled by default. You can configure other users after the initial Junos OS setup phase.

   Use an SSH client to log in to a vSRX instance for the first time. To log in, specify the location where you saved the SSH key pair file for the user account, and the IP address assigned to the vSRX management interface (fxp0).

   NOTE: It is mandatory to use “gcp-user” as username when you login to the vSRX for the first time vSRX.

   If you do not have the key pair filename and the IP address, use these steps to view the key pair name and IP for a vSRX instance:

   a. In the GCP portal, select **Instances**.

   b. Select the vSRX instance, and select **eth0** in the Description tab to view the IP address for the fxp0 management interface.

   c. Click **Connect** above the list of instances to view the SSH key pair filename.

   To configure the basic settings for the vSRX instance, see *Configure vSRX Using the CLI*.

   NOTE: gcloud connect to vSRX is not supported. Always use ssh with user provided key to connect to vSRX after instance is up.

Deploy the vSRX Firewall Using Cloud-init

vSRX supports cloud-init. Cloud-init is an open-source multi-distribution package that handles early initialization of a cloud instance. It allows user to customize VM instance with attributes like hostname and default IP on the first boot. Cloud-init is particularly useful when user wants to deploy large number of VM instances in the data center using automation tools.

Some of the initial provisioning parameters for first boot are:

- Hostname
- Root password
- SSH public key

  NOTE: for the ssh key file, it needs to be in the format "<username>:<key value>" as required by google cloud. Something like this:

- Management interface (fxp0) IP
- Default gateway IP

You can deploy vSRX Firewall using cloud-init in two ways:

- From Google SDK
- To deploy vSRX with cloud-init from Google portal, see "Deploy the vSRX Firewall from GCP Portal" on page 512. To add user-data to have cloud init enabled specify the metadata.

GCE supports cloud-init type instance configuration. To launch instance with user data, use the command below as an example.

Figure 132: Sample Cloud-Init Configuration

```
gcloud compute instances create vsrx-cloudinit-001 --image vsrx-0115 \
--zone=us-west1-b \
--network-interface address=,network=vpc-1-mgt,subnet=subnet-1-uswest1-5  \
--network-interface address=,network=vpc-untrust-global,subnet=subnet-6-
uswest1-16,private-network-ip=10.16.16.113 \
--network-interface no-address,network=vpc-trust-regional,subnet=subnet-7-
uswest1-26,private-network-ip=10.26.26.113 \
--machine-type=n1-standard-4 --can-ip-forward \
--metadata-from-file user-data=junos.conf,ssh-keys=gcp-user.pub
```

Please note the following points:

- junos.conf is configuration file with '#junos-config' in content
- gcp-user.pub is ssh public key
- vSRX 3.0 supports RSA key pair only
- For the SSH key file, it needs to be in the format **<username>:<key value>** as required by Google cloud. Refer the sample SSH key file below.

```
root@cnrd-kvmsrv37:~# cat gcp-user.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDeR2jhMLzSfgee/5cnduTa+13yVLKbTa/
OFnZSHQsZoA5LKHIXs/TbyooZTX5PnfNr6hx2Iyxjaodu01kT0UJ87wps8n9BH74DP6x0YK07OaZZl5T/
5Iso9fXRCzl9+go9vKzNKhqXmqKUc3Fl6hTX2QzQbtrwN2twLzCxz+OSliCoobJr+/
8wPcvI6fUbL6FRTgE1zC1HBlDKspK7x47YDYPJlUcyMhRtGvxd319jrx5i96mZq850+
dCfZkHSipT09hFRtk8C4MsOaKsw3RWUCY5LCPekrutrLLfhMKh88onv4ud7gXOklSwgVVod49aY2FfiaACMAVoaomfYXwe
P
gcp-user



ssh -i <private-key> gcp-user@<vSRX management public ip>
```

- In junos.conf, please remove the “gcp-default” block in your user data. They will shadow the one created by vSRX init script. Refer the sample junos config

```
    #junos-config
security {
    policies {
        default-policy {
            permit-all;
        }
    }
    zones {
        security-zone trust {
            interfaces {
                ge-0/0/0.0;
            }
        }
        security-zone untrust {
            interfaces {
                ge-0/0/1.0;
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                10.0.0.10/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                10.0.1.10/24;
            }
        }
    }
}
```

NOTE: gcloud connect to vSRX is not supported. Always use ssh with user provided key to connect to vSRX after instance is up.

RELATED DOCUMENTATION

Deploying vSRX Virtual Firewalls on Google Cloud Platform

Upgrade the Junos OS for vSRX Software Release

You can upgrade the Junos OS for vSRX software using the CLI. Upgrading or downgrading Junos OS can take several hours, depending on the size and configuration of the network. Download the desired Junos OS Release for the **vSRX.tgz** file from the Juniper Networks website.

You also can upgrade using J-Web (see J-Web) or the Junos Space Network Management Platform (see Junos Space).

For the procedure on upgrading a specific Junos OS for vSRX software release, see the *Migration, Upgrade, and Downgrade Instructions* topic in the release-specific *vSRX Release Notes* available on the vSRX TechLibrary webpage.

Secure Data with vSRX 3.0 Using GCP KMS (HSM)

IN THIS SECTION

This topic describes the integration of vSRX 3.0 with GCP (Google Cloud Platform) Key management Service (KMS) for securing confidential information such as private keys that must be stored within a FIPS boundary. GCP provides support for KMS that is used by applications such as vSRX 3.0 to safeguard and to manage cryptographic keys.

Overview

A wrapper library is available in Junos to enable VPN and other applications (such as mgd) to integrate and communicate with cloud-based KMS. This wrapper library provides interface to Key Management Service (KMS) using PKCS#11 APIs. Junos applications use this wrapper library with updated support for GCP to communicate with KMS. To support PKCS#11 APIs, GCP team provides Juniper a library which acts as an intermediary between Junos applications and Cloud KMS service. This library is added as part of vSRX 3.0 Junos package. There is no action needed from you to enable the libraries.

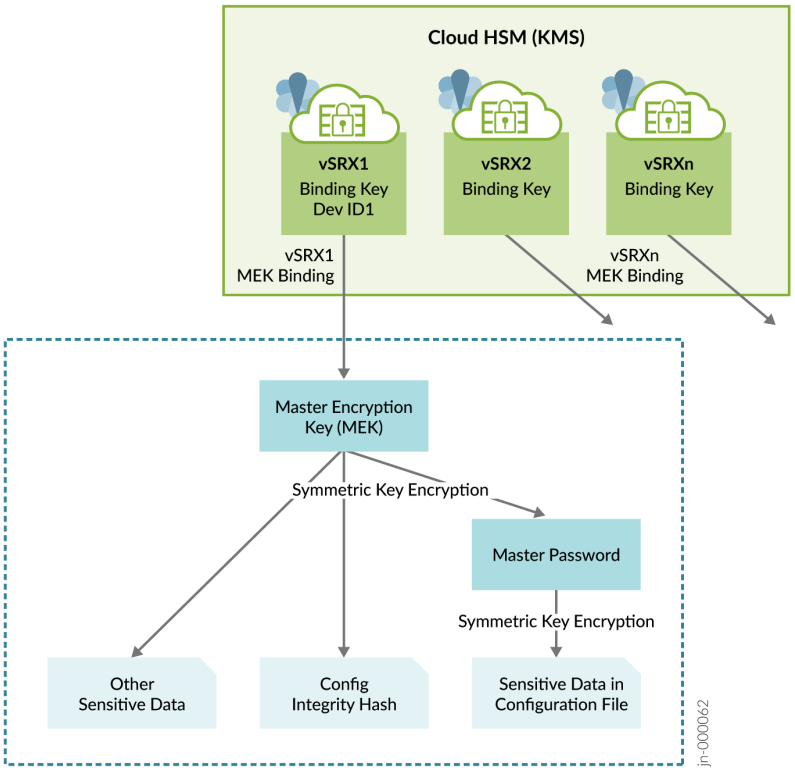
After enabling the KMS service, you need to specify the Master Encryption Key (MEK) using the `request security hsm master-encryption-password set plain-text-password` command. vSRX 3.0 then creates RSA 2048 key pair Master binding Key (MBK) in KMS and encrypts MEK using MBK in KMS. MEK is then used as a key for encrypting data at rest such as hash of configuration, private key pair files and master-password file.

vSRX with GCP KMS has the following limitations:

- vSRX uses management interface to access KMS service. If management interface is not enabled or configured, KMS service cannot be used from vSRX.
- SSL Proxy, Sky-ATP, IDP Signature download or any other module using certificate-based connections will not work when HSM is enabled.
- RSA Key Pair with a Key ID can be generated only once. It cannot be used for another Key Pair to generate or create request, for a deleted Key ID, or for another new Key request.

Figure 1 illustrates the inventory of keys in vSRX 3.0.

Figure 133: Supply of Keys in vSRX 3.0



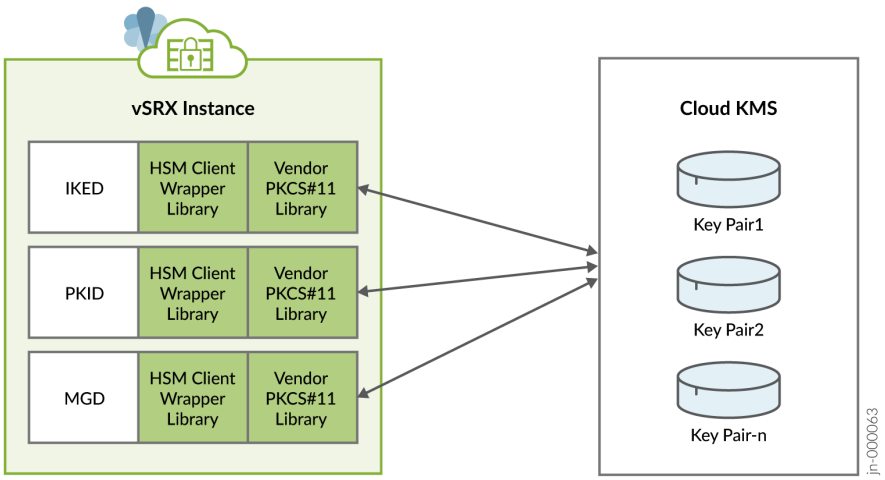
Support for generating Public Key Infrastructure (PKI) key-pairs in GCP cloud KMS is enabled and any request such as RSA SIGN, which needs private key of the generated key-pair is be sent to GCP cloud KMS. Specifically, the following operations have been offloaded to the KMS:

- Private key signing during Certificate Signing Request (CSR) creation in PKI Daemon (PKID) running on the device.
- Private key signing during verification of the certificate received from the CA server in PKID.
- Private key signing during IKE negotiations at Key Management Daemon (KMD) which is the IKE Daemon running on the device.

All the VPN applications (PKID and KMD) will use wrapper library to communicate with the KMS service to create, manage and execute crypto operations on the RSA keys.

Figure 2 illustrates how VPN applications accessing KMS service.

Figure 134: VPN Applications Accessing KMS Service



You can secure data at rest and achieve configuration integrity with vSRX 3.0 using GCP KMS. Perform the steps given in this topic to setup GCP Cloud KMS service and Key Ring for vSRX 3.0.

Key Ring is a component in KMS service where keys created by Junos applications are going to reside. A key ring organizes keys in a specific Google Cloud location and allows you to manage access control on groups of keys. A key ring's name does not need to be unique across a Google Cloud project, but must be unique within a given location. After creation, a key ring cannot be deleted. Key rings do not incur storage costs.

Integrate GCP KMS with vSRX 3.0

To enable and setup vSRX 3.0 to access KMS on GCP.

1. Launch vSRX 3.0 instance in GCP. See Deploying vSRX Virtual Firewalls on Google Cloud Platform and Deploy vSRX in Google Cloud Platform.

2. Setup GCP KMS for vSRX 3.0.

   Before you can enable vSRX 3.0 to communicate with the KMS service, you need to ensure vSRX 3.0 instance is authenticated and authorized to access GCP Cloud KMS service. To setup GCP environment or account do the following:

   a. Create a service account.

A service account is a special type of Google account intended to represent a non-human user such as virtual machines(VMs), that needs to authenticate and be authorized to access data in Google APIs.

vSRX 3.0 uses the PKCS#11 library provided by GCP to access Cloud KMS service. The library uses service accounts to authenticate using service account credentials.

i. To create a new service account to use with vSRX 3.0 to access Cloud KMS, see

   Getting Started with Authentication. If you already have a service account then, see Authenticating as a service account.

ii. Create IAM role for the service account to enable access for vSRX 3.0 instance.

   Once you have service account setup, grant the account a role or roles with the following IAM permissions:

   - cloudkms.cryptoKeys.list on all configured KeyRings.
   - cloudkms.cryptoKeyVersions.list on all CryptoKeys in each configured KeyRing.
   - cloudkms.cryptoKeyVersions.viewPublicKey for all asymmetric keys contained within all configured KeyRings.
   - cloudkms.cryptoKeyVersions.use to Decrypt or cloudkms.cryptoKeyVersions.use to sign for any keys to be used for decryption or signing.
   - cloudkms.cryptoKeys.create if you intend to create keys.
   - cloudkms.cryptoKeyVersions.destroy if you intend to destroy keys.

   You can also use pre-defined groups of IAM roles as listed below to grant service account the needed permissions. For more information about roles associated for each of the above groups, see Permissions and roles

iii. Attach IAM role to vSRX 3.0 instance either from GUI or using GCP CLI.

   After you have service account created and granted needed IAM roles as mentioned above, you can either create a new vSRX 3.0 instance using this service account or set an existing vSRX 3.0 instance to use the service account.

   For more information, see Creating and enabling service accounts for instances.

iv. Create Key Ring

   After granting required access for vSRX 3.0 instance to communicate with KMS, you need to create Key Ring, which is a component in KMS service where keys created by vSRX 3.0 will reside.

Key Ring can be created using gcloud or from console. For more information, see Create a key ring

.

Additionally, GCP KMS does not allow creation of a key with an ID which was already used and created earlier. GCP KMS also does not allow deletion of existing key and creating another key with same name.

> **NOTE**: Key ring can be created in one specific region, dual-regional or multi-regional locations. Location refers to the datacenter in which your keys are going to be saved. If you use one specific region key is located in that location only. In case of dual regions, keys are replicated to other regions and same implies for multi-regional locations. For more information, see Cloud KMS locations.
>
> After you create Key Ring, please note down the resource ID for the key ring as it is needed for input into vSRX 3.0 using CLI. GCP PKCS#11 KMS library on vSRX 3.0 will use this resource ID to communicate with KMS. Name of Key created in Key ring can contain letters, numbers, underscores (_), and hyphens (-).

3. Provide GCP Key Ring resource information using the `request security hsm set gcp project <name_of_project> location <location_of_key_ring> key-ring <name_of_key_ring>` command. For more information, see Getting a Cloud KMS resource ID

4. After enabling the KMS service, you need to specify the Master Encryption Key (MEK) using the `request security hsm master-encryption-password set plain-text-password` command on vSRX 3.0.

   Once you specify the MEK, vSRX 3.0 creates the RSA 2048 key pair (MBK) in KMS and encrypts MEK using Master binding Key (MBK) in KMS. MEK is then used as a key for encrypting data at rest such as hash of configuration, private key pair files and master-password file.

5. Change the Master Encryption Password.

   If you want to change the master encryption password then you can run the `request security hsm master-encryption-password set plain-text-password` command from operational mode:

   > **NOTE**: It is recommended that no configuration changes are made while you are changing the master encryption password.

   The system checks if the master encryption password is already configured. If master encryption password is configured, then you are prompted to enter the current master encryption password.

   The entered master encryption password is validated against the current master encryption password to make sure these master encryption passwords match. If the validation succeeds, you will

be prompted to enter the new master encryption password as plain text. You will be asked to enter the key twice to validate the password.

The system then proceeds to re-encrypt the sensitive data with the new master encryption password. You must wait for this process of re-encryption to complete before attempting to change the master encryption password again.

If the encrypted master encryption password file is lost or corrupted, the system will not be able to decrypt the sensitive data. The system can only be recovered by re-importing the sensitive data in clear text, and re-encrypting them.

6. Check HSM status using the `show security hsm status` command to check if KMS is enabled and reachable, also displays the Resource ID of Key Ring being used, Master binding Key (MBK), and Master Encryption Key (MEK) status.

Verify the Status of the HSM

IN THIS SECTION

Purpose

To check connectivity with HSM.

Action

You can use the `show security hsm status` command to verify the status of the HSM. The following information is displayed:

- If HSM is enabled and reachable or disabled
- Is Master Binding Key (RSA Key pair) created in HSM
- Is Master Encryption Key configured - master encryption password status (set or not set)
- Cloud vendor Information

show security hsm status

IN THIS SECTION

Syntax

```
show security hsm status
```

Release Information

Command introduced in Junos OS Release 19.4R1.

Description

Display the current status of the Hardware Security Module (HSM). You can use this `show security hsm status` command to check the status of HSM, master binding key, master encryption password, and cloud vendor details.

Options

This command has no options.

Required Privilege Level

security

Output Fields

Table 88 on page 526 lists the output fields for the `show security hsm status` command.

Table 88: show security hsm status Output Fields

Field Name	Field Description
`Enabled`	Specifies whether HSM is enabled or disabled.
`Master Binding Key`	Displays the HSM's Master Binding Key status whether it is created or not created in HSM. HSM generates cryptographic keys and encrypts them so that those can only be decrypted by the HSM. This process is know as binding. Each HSM has a master binding key, which is also know as storage root key.
`Master Encryption Key`	Displays Master Encryption configuration status whether it is set or not set. The encrypted data and the hash of the configuration is protected by vSRX using Microsoft Key Vault (HSM) service.
`Cloud vendor Details`	Displays the details specific to the cloud vendor.

Sample Output

show security hsm status (HSM status command output when vSRX initially boots up but GCP KMS feature is not enabled)

```
user@host>              show security hsm status
```

```
HSM Status:
  Accessible: no
  Master Binding Key: not-created
  Master Encryption Key: not-configured
```

show security hsm status (HSM status command output after successful integration with GCP KMS)

```
user@host>              show security hsm status
```

```
HSM Status:
  Accessible: yes
  Master Binding Key: not-created
  Master Encryption Key: not-configured
 GCP Key Ring: projects/example-project-98765/locations/us-central1/keyRings/example-ring
```

request security hsm master-encryption-password

IN THIS SECTION

Syntax

```
request security hsm master-encryption-password set plain-text-password
```

Release Information

Command introduced in Junos OS Release 19.4R1.

Description

Use this command to set or replace the password (in plain text).

Options

plain-text-password	Set or replace the password (in plain text).

Required Privilege Level

maintenance

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security hsm master-encryption-password set plain-text-password

```
user@host>              request security hsm master-encryption-password set plain-text-password
```

```
   Enter new master encryption password:
   Repeat new master encryption password:
   Binding password with HSM
   Master encryption password is bound to HSM
   Encoding master password ..
   Successfully encoded master password
   Deleting all previous local certificates, keypairs and certificate requests
```

PART 9

vSRX Deployment for IBM Cloud

CHAPTER 27

Overview

IN THIS CHAPTER

vSRX Overview

SUMMARY

In this topic you learn about vSRX architecture and its benefits.

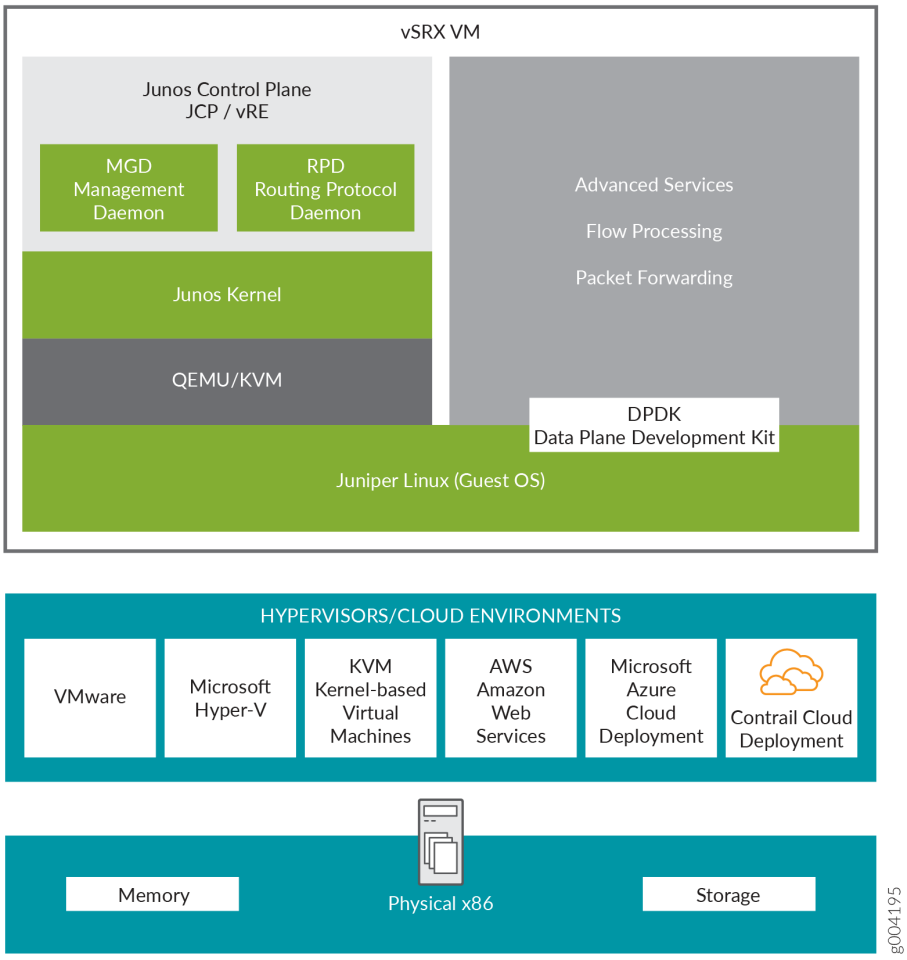
IN THIS SECTION

vSRX is a virtual security appliance that provides security and networking services at the perimeter or edge in virtualized private or public *cloud* environments. vSRX runs as a virtual machine (*VM*) on a standard x86 server. vSRX is built on the Junos operating system (Junos OS) and delivers networking and security features similar to those available on the software releases for the SRX Series Services Gateways.

The vSRX provides you with a complete Next-Generation Firewall (NGFW) solution, including core firewall, VPN, NAT, advanced Layer 4 through Layer 7 security services such as Application Security, intrusion detection and prevention (IPS), and UTM features including Enhanced Web Filtering and Anti-Virus. Combined with Sky ATP, the vSRX offers a cloud-based advanced anti-malware service with dynamic analysis to protect against sophisticated malware, and provides built-in machine learning to improve verdict efficacy and decrease time to remediation.

Figure 135 on page 531 shows the high-level architecture.

Figure 135: vSRX Architecture



vSRX includes the Junos control plane (JCP) and the packet forwarding engine (PFE) components that make up the data plane. vSRX uses one virtual CPU (vCPU) for the JCP and at least one vCPU for the PFE. Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, multi-core vSRX supports scaling vCPUs and virtual RAM (vRAM). Additional vCPUs are applied to the data plane to increase performance.

Benefits

vSRX on standard x86 servers enables you to quickly introduce new services, deliver customized services to customers, and scale security services based on dynamic needs. vSRX is ideal for public, private, and hybrid cloud environments.

Some of the key benefits of vSRX in a virtualized private or public cloud multitenant environment include:

- *Stateful firewall* protection at the tenant edge
- Faster deployment of virtual firewalls into new sites
- Ability to run on top of various hypervisors and public cloud infrastructures
- Full routing, *VPN*, core security, and networking capabilities
- Application security features (including IPS and App-Secure)
- Content security features (including Anti Virus, Web Filtering, Anti Spam, and Content Filtering)
- Centralized management with Junos Space Security Director and local management with J-Web Interface
- Juniper Networks Sky Advanced Threat Prevention (Sky ATP) integration

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, multi-core vSRX supports scaling vCPUs and virtual RAM (vRAM). Additional vCPUs are applied to the data plane to increase performance.

Getting Started with Juniper vSRX on IBM Cloud

IN THIS SECTION

IBM Cloud™ Juniper vSRX allows you to route private and public network traffic selectively, through a full-featured, enterprise-level firewall that is powered by Junos OS software features, such as full routing stacks, QoS and traffic sharing, policy-based routing, and VPN.

NOTE: For a list of known limitations with IBM Cloud™ Juniper vSRX Gateway, see Known limitations.

Overview of vSRX in IBM Cloud

IN THIS SECTION

The vSRX provides performance, ease of configuration, and maintenance advantages with the simplicity of running on a bare metal server. The hardware is sized to handle the routing and security load associated with multiple VLANs, and it can be ordered with redundant network links and redundant RAID arrays. All vSRX features are customer-managed.

The IBM Cloud™ Juniper vSRX is offered in two different modes: standalone mode or High Availability (HA) cluster.

For additional documentation for IBM Cloud™ Juniper vSRX, see Supplemental Documentation.

The vSRX deploys to protect your environment from external and internal threats by filtering private- and public-facing traffic. Customers can manage the vSRX themselves by defining policies and rules that allow or deny (among other actions) inbound or outbound network traffic, thereby protecting their applications from internal and external approaches. Both IPv4 and IPv6 stacks are supported in a stateful manner.

Connect your on-site data center or office to the IBM Cloud using VPN tunneling by provisioning your vSRX as a network gateway device. Remote access IPsec VPN also is supported.

For a detailed configurations on VPN, see VPN.

With the vSRX gateway appliance, you can provision application and database servers without public network interfaces, and still allow your servers access to the Internet using source NAT. For enhanced security, you can protect your servers behind the gateway device, using destination NAT.

You can set up dynamic routing using BGP, which allows you to announce your own public IP space to the IBM Cloud routers.

A VLAN (virtual local area network) is a mechanism that segregates a physical network into many virtual segments. For convenience, traffic from multiple selected VLANs can be delivered through a single network cable, using a process commonly called "trunking."

vSRX is managed in two different interfaces: The vSRX server(s) and the Gateway Appliance fixture. Servers in an associated VLAN can be reached from other VLANs only by going through your vSRX; it is not possible to circumvent the vSRX unless you bypass or disassociate the VLAN.

By default, a new Gateway Appliance is associated with two non-removable "transit" VLANs, one each for your public and private networks. These networks typically are used for administration, and they can be secured by vSRX commands separately. The vSRX can manage VLANs that are associated with it through the Gateway Appliance (only).

For information on how to manage VLANs from the **Gateway Appliances Details** screen, see Manage VLANs.

IBM© Cloud offers several firewalls to choose from. See Exploring firewalls section that provides comparison of the supported firewall solutions to help you choose the one that is right for you.

Benefits of vSRX in IBM Cloud

vSRX support in IBM Cloud offers you the following benefits:

- You can use an IPsec site-to-site VPN tunnel for secure communication from your enterprise data center or office to your IBM Cloud network.
- Empowers you with greater flexibility to build connectivity between multi-tiered applications running on different isolated networks.
- BGP offers more flexibility for custom private network configurations, when you're using a mix of tunnels and Direct Link solutions.
- The Gateway Appliance provides an interface (GUI and API) for selecting the VLANs you want to associate with your vSRX. Associating a VLAN with a Gateway Appliance reroutes (or "trunks") that VLAN and all of its subnets to your vSRX, gives you control over filtering, forwarding, and protection.

Choosing a vSRX license

There are two license types available for your IBM Cloud™ Juniper vSRX:

- Standard
- Content Security Bundle (CSB)

Each license includes a different set of features and options, and the following table outlines the differences.

NOTE: You can specify your license type when ordering your vSRX, as well as change the license, see Gateway Appliance Details.

License Type	Features
Standard	• Core security: firewall, ALG, screens, user firewall • IPsec VPN (site-to-site VPN) • NAT • CoS • Routing services: BGP, OSPF, DHCP, J-Flow, IPv4 • Foundation: Static routing, management (J-Web, CLI, and NETCONF), on-box logging, diagnostics

(Continued)

License Type	Features

Content Security Bundle (CSB)—Includes all Standard features, along with the additional features listed in the next column.	• AppSecure
	  • Application Tracking (AppTrack)
	  • Application Firewall (AppFW)
	  • Application Quality of Service (AppQoS)
	  • Advanced policy-based routing (APBR)
	  • Application Quality of Experience (AppQoE)
	• User Firewall
	• IPS
	• UTM
	  • Anti Virus
	  • Anti Spam
	  • Web Filtering
	  • Content Filtering
	• SSL Proxy
	  • SSL Forward Proxy
	  • SSL Reverse Proxy
	  • SSL Decrypting Mirror

Ordering a vSRX

You can order your IBM Cloud™ Juniper vSRX by performing the following procedure:

1. From your browser, open the Gateway Appliances page in the IBM Cloud catalog and log in to your account.

   You can also get to this page by logging in to the IBM Cloud UI console and selecting **Classic Infrastructure** > **Network** > **Gateway appliance**. Alternatively, from the IBM Cloud catalog, select the **Network category** then choose the **Gateway appliance** tile.

2. Choose **Juniper vSRX (up to 1 Gbps)** or **Juniper vSRX (up to 10 Gbps)** under **Gateway Vendor**.
3. Choose your license type from **License add-ons**, either Standard or CSB. See "Choosing a vSRX license" on page 534 section for information on the features offered with each license.
4. From the **Gateway appliance** section, enter your **Host name** and **Domain** name. These fields are already be populated with default information, so ensure that the values are correct.
5. Check the **High Availability** option if needed, then select a data center **Location**, and the specific **Pod** you want from the menu.

   NOTE: Only pods that already have an associated VLAN are displayed here. If you want to provision your gateway appliance in a pod you don't see listed, first create a VLAN there.

6. From the **Configuration** section, choose your processor's RAM. You can also define an SSH key, if you want to use it to authenticate access to your new Gateway.

   The appropriate processor is chosen for you based on the license version you selected in step two. However, you can choose different RAM configurations.

7. From the **Storage disks** section, choose the options that meet your storage requirements. Reserve more than the default disk setting if you plan to run network diagnostics that generate detailed logs.

   RAID0 and RAID1 options are available for added protection against data loss, as are hot spares (backup components that can be placed into service immediately when a primary component fails). You can have up to four disks per vSRX. "Disk size" with a RAID configuration is the usable disk size, as RAID configurations are mirrored.

8. From the **Network interface** section, select your **Uplink port speeds**. The default selection is a single interface, but there are redundant and private only options as well. Choose the one that best fits your needs.

   The Network Interface **Add Ons** section allows you to select an IPv6 address if required, and shows you any additional included default options.

9. Review your selections, check that you have read the Third Party Service Agreements, then click **Create**. The order is verified automatically.

After your order is approved, the provisioning of your IBM Cloud™ Juniper vSRX Gateway starts automatically. When the provisioning process is complete, the new vSRX appears in the Gateway Appliances list page. Click the gateway name to open the Gateway Details page. The IP addresses, login username, and password for the device appear. Remember that after you order and configure your gateway from the IBM Cloud catalog, you must also configure the device itself with the same settings.

Junos OS Features Supported on vSRX

SUMMARY

This topic provides details of the Junos OS features supported and not supported on vSRX.

IN THIS SECTION

SRX Series Features Supported on vSRX

vSRX inherits most of the branch SRX Series features with the following considerations shown in Table 89 on page 538.

To determine the Junos OS features supported on vSRX, use the Juniper Networks Feature Explorer, a Web-based application that helps you to explore and compare Junos OS feature information to find the right software release and hardware platform for your network. Find Feature Explorer at: Feature Explorer: vSRX .

Table 89: vSRX Feature Considerations

Feature	Description
IDP	The IDP feature is subscription based and must be purchased. After purchase, you can activate the IDP feature with the license key. For SRX Series IDP configuration details, see: Understanding Intrusion Detection and Prevention for SRX Series

Table 89: vSRX Feature Considerations *(Continued)*

Feature	Description	
IPSec VPNs	Starting in Junos OS Release 19.3R1, vSRX supports the following authentication algorithms and encryption algorithms: • Authentication algorithm: hmac-sha1-96 and HMAC-SHA-256-128 authentication • Encryption algorithm: aes-128-cbc Starting in Junos OS Release 20.3R1, vSRX supports 10,000 IPsec VPN tunnels. To support the increased number of IPsec VPN tunnels, a minimum of 19 vCPUs are required. Out of the 19 vCPUs, 3 vCPUs must be dedicated to RE. You must run the `request system software add optional://junos-ike.tgz` command the first time you wish to enable increased IPsec tunnel capacity. For subsequent software upgrades of the instance, the junos-ike package is upgraded automatically from the new Junos OS releases installed in the instance. DH group15, group16, group21 is also added when we install junos-ike package. If chassis cluster is enabled then run this command on both the nodes. You can configure the number of vCPUs allocated to Junos Routing Engine using the `set security forwarding-options resource-manager cpu re <value>`. **NOTE**: 64 G memory is required to support 10000 tunnels in PMI mode. [See show security ipsec security-associations, show security ike tunnel-map, and show security ipsec tunnel-distribution.]	
IPsec VPN - Tunnel Scaling on vSRX	Types of Tunnels	Number of tunnels supported
	Site-Site VPN tunnels	2000
	AutoVPN tunnels	10,000
	IKE SA (Site-to-site)	2000
	IKE SA (AutoVPN)	10,000
	IKE SA (Site-to-site + AutoVPN)	10,000

Table 89: vSRX Feature Considerations *(Continued)*

Feature	Description	
	IPSec SA pairs (Site-to-site)	10,000 With 2000 IKE SAs, we can have 10,000 IPSec SA.
	IPSec SA pairs (AutoVPN)	10,000
	Site-to-site + AutoVPN IPSec SA pairs	2000 Site-to-site 8000 AutoVPN
	Site-to-site + AutoVPN tunnels	2000 Site-to-site 8000 AutoVPN
ISSU	ISSU is not supported.	
Logical Systems	Starting in Junos OS Release 20.1R1, you can configure logical systems and tenant systems on vSRX and vSRX 3.0 instances. With Junos OS, you can partition a single security device into multiple logical devices that can perform independent tasks. Each logical system has its own discrete administrative domain, logical interfaces, routing instances, security firewall and other security features. See Logical Systems Overview.	

Table 89: vSRX Feature Considerations *(Continued)*

Feature	Description
PowerMode IPsec	Starting in Junos OS Release 20.1R1, vSRX 3.0 instances support PowerMode IPsec that provides IPsec performance improvements using Vector Packet Processing (VPP) and Intel AES-NI instructions. PowerMode IPsec is a small software block inside the SRX PFE (SRX Packet Forwarding Engine) that is activated when PowerMode is enabled. Supported Features in PowerMode IPsec • IPsec functionality • Traffic selectors • Secure tunnel interface (st0) • All control plane IKE functionality • Auto VPN with traffic selector • Auto VPN with routing protocol • IPv6 • Stateful Layer 4 firewall • High-Availability • NAT-T Non-Supported Features in PowerMode IPsec • NAT • IPsec in IPsec • GTP/SCTP firewall • Application firewall/AppSecure • QoS • Nested tunnel • Screen • Multicast

Table 89: vSRX Feature Considerations *(Continued)*

Feature	Description
	• Host traffic
Tenant Systems	Starting in Junos OS Release 20.1R1, you can configure tenant systems on vSRX and vSRX 3.0 instances. A tenant system provides logical partitioning of the SRX device into multiple domains similar to logical systems and provides high scalability. See Tenant Systems Overview.
Transparent mode	The known behaviors for transparent mode support on vSRX are: • The default MAC learning table size is restricted to 16,383 entries. For information about configuring transparent mode for vSRX, see Layer 2 Bridging and Transparent Mode Overview.

Table 89: vSRX Feature Considerations *(Continued)*

Feature	Description
UTM	• The UTM feature is subscription based and must be purchased. After purchase, you can activate the UTM feature with the license key. • Starting in Junos OS Release 19.4R1, vSRX 3.0 instances support the Avira scan engine, which is an on-device antivirus scanning engine. See On-Device Antivirus Scan Engine. • For SRX Series UTM configuration details, see Unified Threat Management Overview. • For SRX Series UTM antispam configuration details, see Antispam Filtering Overview. • **Advanced resource management (vSRX 3.0)**—Starting in Junos OS Release 19.4R1, vSRX 3.0 manages the additional system resource requirements for UTM-and IDP-specific services by reallocating CPU cores and extra memory. These values for memory and CPU cores are not user configured. Previously, system resources such as memory and CPU cores were fixed. You can view the allocated CPU and memory for advance security services on vSRX 3.0 instance by using the `show security forward-options resource-manager settings` command. To view the flow session scaling, use the `show security monitoring` command. [See show security monitoring and show security forward-options resource-manager settings.]

Some Junos OS software features require a license to activate the feature. To understand more about vSRX Licenses, see, Licenses for vSRX. Please refer to the Licensing Guide for general information about License Management. Please refer to the product Data Sheets for further details, or contact your Juniper Account Team or Juniper Partner.

SRX Series Features Not Supported on vSRX

vSRX inherits many features from the SRX Series device product line. Table 90 on page 544 lists SRX Series features that are not applicable in a virtualized environment, that are not currently supported, or that have qualified support on vSRX.

Table 90: SRX Series Features Not Supported on vSRX

SRX Series Feature	vSRX Notes
Application Layer Gateways	
Avaya H.323	Not supported
Authentication with IC Series devices	
Layer 2 enforcement in UAC deployments	Not supported **NOTE**: UAC-IDP and UAC-UTM also are not supported.
Chassis cluster support **NOTE**: Support for chassis clustering to provide network node redundancy is only available on a vSRX deployment in Contrail, VMware, KVM, and Windows Hyper-V Server 2016.	
Chassis cluster for VirtIO driver	Only supported with KVM **NOTE**: The link status of VirtIO interfaces is always reported as UP, so a vSRX chassis cluster cannot receive link up and link down messages from VirtIO interfaces.
Dual control links	Not supported
In-band and low-impact cluster upgrades	Not supported
LAG and LACP (Layer 2 and Layer 3)	Not supported
Layer 2 Ethernet switching	Not supported
Low-latency firewall	Not supported
Class of service	
High-priority queue on SPC	Not supported

Table 90: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
Tunnels	Only GRE and IP-IP tunnels supported **NOTE**: A vSRX VM deployed on Microsoft Azure Cloud does not support GRE and multicast.
Data plane security log messages (stream mode)	
TLS protocol	Not supported
Diagnostic tools	
Flow monitoring cflowd version 9	Not supported
Ping Ethernet (CFM)	Not supported
Traceroute Ethernet (CFM)	Not supported
DNS proxy	
Dynamic DNS	Not supported
Ethernet link aggregation	
LACP in standalone or chassis cluster mode	Not supported
Layer 3 LAG on routed ports	Not supported
Static LAG in standalone or chassis cluster mode	Not supported
Ethernet link fault management	

Table 90: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
Physical interface (encapsulations) • ethernet-ccc • ethernet-tcc • extended-vlan-ccc • extended-vlan-tcc	Not supported
Interface family • ccc, tcc • ethernet-switching	Not supported
Flow-based and packet-based processing	
End-to-end packet debugging	Not supported
Network processor bundling	
Services offloading	
Interfaces	
Aggregated Ethernet interface	Not supported
IEEE 802.1X dynamic VLAN assignment	Not supported
IEEE 802.1X MAC bypass	Not supported
IEEE 802.1X port-based authentication control with multisupplicant support	Not supported

Table 90: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
Interleaving using MLFR	Not supported
PoE	Not supported
PPP interface	Not supported
PPPoE-based radio-to-router protocol	Not supported
PPPoE interface **NOTE**: Starting in Junos OS Release 15.1X49-D100 and Junos OS Release 17.4R1, the vSRX supports Point-to-Point Protocol over Ethernet (PPPoE) interface.	Not supported
Promiscuous mode on interfaces	Only supported if enabled on the hypervisor
IPSec and VPNs	
Acadia - Clientless VPN	Not supported
DVPN	Not supported
Hardware IPsec (bulk crypto) Cavium/RMI	Not supported
IPsec tunnel termination in routing instances	Supported on virtual router only
Multicast for AutoVPN	Not supported
IPv6 support	
DS-Lite concentrator (also called Address Family Transition Router [AFTR])	Not supported

Table 90: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
DS-Lite initiator (aka B4)	Not supported
J-Web	
Enhanced routing configuration	Not supported
New Setup wizard (for new configurations)	Not supported
PPPoE wizard	Not supported
Remote VPN wizard	Not supported
Rescue link on dashboard	Not supported
UTM configuration for Kaspersky antivirus and the default Web filtering profile	Not supported
Log file formats for system (control plane) logs	
Binary format (binary)	Not supported
WELF	Not supported
Miscellaneous	
GPRS **NOTE**: Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, vSRX supports GPRS.	Not supported
Hardware acceleration	Not supported

Table 90: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
Outbound SSH	Not supported
Remote instance access	Not supported
USB modem	Not supported
Wireless LAN	Not supported
MPLS	
Crcuit cross-connect (CCC) and translational cross-connect (TCC)	Not supported
Layer 2 VPNs for Ethernet connections	Only if promiscuous mode is enabled on the hypervisor
Network Address Translation	
Maximize persistent NAT bindings	Not supported
Packet capture	
Packet capture	Only supported on physical interfaces and tunnel interfaces, such as *gr*, *ip*, and *st0*. Packet capture is not supported on redundant Ethernet interfaces (*reth*).
Routing	
BGP extensions for IPv6	Not supported
BGP Flowspec	Not supported
BGP route reflector	Not supported

Table 90: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
CRTP	Not supported
Switching	
Layer 3 Q-in-Q VLAN tagging	Not supported
Transparent mode	
UTM	Not supported
Unified threat management	
Express AV	Not supported
Kaspersky AV	Not supported
Upgrading and rebooting	
Autorecovery	Not supported
Boot instance configuration	Not supported
Boot instance recovery	Not supported
Dual-root partitioning	Not supported
OS rollback	Not supported
User interfaces	
NSM	Not supported

Table 90: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
SRC application	Not supported
Junos Space Virtual Director	Only supported with VMware

CHAPTER 28

Installing and Configuring vSRX in IBM

IN THIS CHAPTER

Performing vSRX Basics in IBM Cloud

IN THIS SECTION

Viewing all gateway appliances

The Gateway Appliances page in the IBM Cloud® console is where you can view and access all network gateway appliances, including IBM Virtual Router Appliances and IBM Juniper vSRX Standard.

Perform the following procedure to access the **Gateway Appliances** page in the IBM Cloud console:

1. From your browser, open the IBM Cloud catalog and log in to your account.
2. Select the **Menu** from the top left, then click **Classic Infrastructure**.
3. Choose **Network** > **Gateway Appliances**.

Viewing gateway appliance details

Network gateways are used to control network traffic on a VLAN that is regularly controlled by a router. Within the Gateway Appliance Details page on the IBM Cloud console, you can associate, disassociate, route and bypass VLANs associated with a network gateway.

Perform the following procedure to go to the **Gateway Appliance Details** page.

1. From your browser, open the IBM Cloud catalog and log in to your account.
2. Select the **Menu** from the top left, then click **Classic Infrastructure**.
3. Choose **Network** > **Gateway Appliances**.
4. Click the name of the network gateway you want to view to access the Gateway Appliance Details page. Use the Bulk Actions feature to take action on multiple VLANs at the same time.

Renaming a gateway appliance

Network gateways are given unique names that assist users in their identification. At any time, you can change a gateway name using the instructions here. It is recommended that you use a consistent naming convention to more easily identify gateways.

Perform the following procedure to rename a network gateway:

1. Access the Gateway Appliance Details page in the IBM Cloud console.
2. Click the **Actions** menu and select **Rename Gateway**.
3. Enter the new gateway name in the **Gateway Name** field.
4. Click **OK** to save the change.

After changing a gateway appliance's name, the name immediately changes at the top of the Gateway Appliance Details page. You can change the gateway name at any time by repeating these steps.

> **NOTE**: Changing the name of the gateway appliance in the IBM Cloud console does not automatically change the hostname within the Virtual Router Appliance or any DNS entries that you might have. Changing the hostname requires manual intervention.

Canceling a gateway appliance

You can cancel your gateway appliance at any time by following these instructions.

1. From your browser, open the IBM Cloud catalog and log in to your account.
2. Select the **Menu** from the top left, then click **Classic Infrastructure**.
3. Choose **Network** > **Gateway Appliances**.
4. Click the Gateway Appliance name to open the Gateway Appliance Details page.
5. From the Hardware section, click the name of the hardware member to open the server details page.
6. Select **Actions** > **Cancel device** and follow the prompts to cancel the gateway appliance.

> **NOTE**: For Highly Available server pairs, you must select and cancel both server members listed in the Hardware section on the Gateway Appliance Details page to cancel the gateway.

After you cancel the gateway appliance, the server(s) are reclaimed at the next billing cycle. For example, if you cancel the server(s) on September 8, the service is available until it is reclaimed on October 1.

You can verify if your gateway appliance is in the process of being canceled by viewing the Gateway Appliance Details page. Gateways in the process of being canceled show as Cancel pending.

> **NOTE**: If necessary, you can expedite the process by opening a case with IBM Support and requesting that the gateway appliance be reclaimed immediately. This process can take 24 to 48 hours.

Performing additional vSRX tasks

IN THIS SECTION

You can configure and maintain your IBM Cloud™ Juniper vSRX in a variety of ways, either through a remote console session through SSH or by logging into the Juniper web management GUI.

NOTE: Configuring the vSRX outside of its shell and interface may produce unexpected results and is not recommended.

Accessing the device using SSH

You can access either the vSRX or the host (Ubuntu) using SSH through a private IP address if you're on IBM Cloud VPN. Additionally, you can access the vSRX through a public IP address as well.

1. Go to Gateway Appliance Details screen and get the Public gateway IP or Private Gateway IP.
2. Click the "eye" icon to reveal the admin user's password.
3. For a vSRX, run the command ssh admin@<gateway-ip>, then enter the admin user's password. You can also use the 'root' user ID and password.

   NOTE: For the host (Ubuntu), you can only use the root user ID and password. Also, if you do not see the "eye" icon, you may not have permission to view the password. Please check your access permissions with the account owner.

Accessing the configuration mode

You can enter the configuration mode, once a shell has been opened to the vSRX, by running the config command. You can do several things in this mode using the following commands:

- show - View configurations
- show | compare - View staged changes
- set - Stage changes
- commit check - Verify the syntax of the configuration

If you are happy with your changes, you can commit them to the active configuration by running the commands commit and then save. To leave Configuration mode run the command exit.

Accessing the Device using the Juniper web management UI

The Juniper web management GUI has been configured by default, with vSRX generated self-signed certificate. Only https is enabled on port 8443. You can access it at https://gateway-ip:8443.

Creating system users

By default, the IBM Cloud™ Juniper vSRX is configured with SSH access for the username admin. Additional users can be added with their own set of priorities. For example: **set system login user ops class operator authentication encrypted-password <CYPHER>**. In this example, ops is the username and operator is the class/permission level assigned to the user. Customized classes can be also defined as opposed to pre-defined ones.

Defining the vSRX hostname

You can set or change the vSRX hostname using the following command: set system host-name <hostname>

Configuring DNS and NTP

To configure name server resolution and NTP, run the following commands:

- set system name-server <DNS server>
- set system ntp <NTP server>

Changing the root password

You can change the root password by running the following command: **set system root-authentication plain-text-password**. This prompts you to input a new password, which is encrypted and stored in the configuration, and is not visible.

vSRX Readiness Checks in IBM Cloud

IN THIS SECTION

Checking vSRX readiness

A readiness check verifies the ability of your IBM Cloud™ Juniper vSRX to perform certain gateway actions. They include:

- OS reloads
- License upgrades
- Version upgrades

Once you run the readiness check, errors will alert you to any necessary actions you should take before beginning one of these actions, or inform you that you're ready to proceed.

To run a readiness check, perform the following procedure:

1. From your browser, open the IBM Cloud catalog and log in to your account.
2. Select the **Menu** from the top left, then click **Classic Infrastructure**.
3. Choose **Network** > **Gateway Appliances**.
4. Click the name of the vSRX you want to run a readiness check on.
5. Find the Readiness Check module on the vSRX details page.
6. Click the **Run** check button.
7. The details page for your vSRX displays again, as do the test results in the readiness check module.

> **NOTE**: Ensure the status for any action you wish to perform is Ready before beginning that action.

Readiness status

There are seven unique status conditions for the readiness check that you may encounter.

- **Unchecked**—A readiness check has not yet been run for this action.
- **Expired**—The readiness check has not run recently enough to reflect accurate results. Run a new check to see the current status.

- **Ready**—Your vSRX is ready to perform the given action.
- **Not Ready**—Your vSRX is not ready to perform the action in question. This could occur because of several reasons. Either a readiness check error occurred, or the readiness check did not complete fast enough, and timed out.

  Error messages for the issues found during the readiness check display next to the module. Click on the error codes to get more information on each error. Alternatively, you can find information about each error in the topic Understanding readiness errors.
- **Running**—The readiness check is currently running on your vSRX, and has not currently encountered any errors.
- **Incomplete**—The first member of the gateway's highly available (HA) setup failed the readiness check. As a result, the gateway could not complete the readiness check.
- **Unsupported**—The action you are attempting to check is not supported for this gateway.
- **Current**— The action you are attempting to check does not need to be performed, as the gateway already has the latest version available.

Readiness check errors you may encounter can either be common errors or version upgrade errors. The below lists provide additional information on these error codes.

To understand common errors that might occur when running readiness checks, see Common readiness errors.

Correcting readiness errors

There are two categories of errors you might encounter when performing readiness checks:

- Host (Ubuntu) SSH connectivity errors
- Gateway (vSRX) SSH connectivity errors

Many of these errors result from the fact that the gateway actions being checked require root SSH access to the private IP address for either the Ubuntu (Host) OS or the vSRX (Gateway). If a SSH connectivity check fails, then the action cannot proceed.

For details on how to ensure that the SSH session can be established, refer to Accessing the device using SSH. Note that for step 3, the example given is with the admin user. For a readiness check, substitute the root user for both the vSRX and the Hardware (host). Also, make sure you use your private IP with this procedure, not your public one.

To validate connectivity, open an SSH session to either the Ubuntu host's or vSRX's private IP using the root credentials listed in the Hardware section (for an Ubuntu host) or the vSRX section (for the gateway) of the Gateway Details page. Ensure that the SSH session can be established.

If the session cannot be established, check the potential following issues:

- **For Host (Ubuntu) SSH connectivity errors:**
  - Is the Ubuntu firewall blocking SSH access to the private IP? The firewall rules must allow SSH access to the private 10.0.0.0/8 subnet. For more information on IBM Cloud IP Ranges for the service network, see IBM Cloud IP ranges.
  - Is the root password listed on the Gateway Details page the correct password for the root user? If not, click the device link under the Hardware section and navigate to Passwords. Select **Actions** > **Edit credentials** nd change the password to match the actual root password on the Ubuntu host.
  - Is the root login disabled for the SSH server? Is the SSH server disabled or stopped?
  - Is the root user account disabled on the Ubuntu host?
- **For Gateway (vSRX) SSH connectivity errors:**
  - Is the vSRX firewall blocking SSH access to the private IP? The firewall rules must allow SSH access to the private 10.0.0.0/8 subnet. For more information on IBM Cloud IP Ranges for the service network, see IBM Cloud IP ranges.
  - Is the root password listed on the Gateway Details page the correct password for the root user? If not, click the Edit icon next to the root password and change the password to match the actual root password for the vSRX.
  - Is the root user account disabled for SSH access to the vSRX?

Managing VLANs with a gateway appliance

IN THIS SECTION

You can manage, associate, disassociate, route, and bypass VLANs with a gateway appliance. You can perform these actions from the Gateway Appliance Details page.

Associating a VLAN to a gateway appliance

A VLAN must be associated to a gateway appliance before it can be routed. VLAN association is the linking of an eligible VLAN to a network gateway so that it can be routed to a gateway appliance in the future. The process of association does not automatically route a VLAN to a gateway appliance; the VLAN continues to use front-end and back-end customer routers until it is routed to the gateway.

VLANs can be associated to only one gateway at a time and must not have a firewall. Perform the following procedure to associate a VLAN to a network gateway.

- Access the Gateway Appliance Details page in the IBM Cloud console.
- Select the VLAN you want from the **Associate a VLAN** list.
- Click the **Associate** button to associate the VLAN.

After associating a VLAN to the gateway appliance, it appears in the Associated VLANs section of the Gateway Appliance Details page. From this section, the VLAN can be routed to the gateway, or be disassociated from the gateway. Additional eligible VLANs can be associated to a gateway appliance at any time by repeating these steps.

Routing an associated VLAN

Associated VLANs are linked to a gateway appliance, but traffic in and out of the VLAN does not hit the gateway until the VLAN is routed. After an associated VLAN is routed, all front-end and back-end traffic is routed through the gateway appliance as opposed to customer routers.

Perform the following procedure to route an associated VLAN:

- Access the Gateway Appliance Details page in the IBM Cloud console.
- Select the VLAN you want from the **Associate a VLAN** list.
- Click the **Associate** button to associate the VLAN.
- Select **Route VLAN** from the Actions menu.
- Click **Yes** to route the VLAN.

After routing a VLAN, all front-end and back-end traffic moves from the customer routers to the network gateway. Additional controls related to traffic and the gateway appliance itself can be taken by accessing the gateway's management tool. Routing through the network gateway can be discontinued at any time by bypassing the gateway appliance.

Bypassing gateway appliance routing for a VLAN

After a VLAN is routed, all front-end and back-end traffic travels through the network gateway. At any time, the gateway appliance can be bypassed so that traffic returns to the front-end and back-end customer routers (FCR and BCR).

Bypassing a VLAN allows the VLAN to remain associated to the network gateway. If the VLAN should no longer be associated with the gateway appliance, see Disassociating a VLAN from a gateway appliance.

Perform the following procedure to bypass gateway routing for a VLAN:

- Access the Gateway Appliance Details page in the IBM Cloud console.
- Select the VLAN you want from the **Associate a VLAN** list.
- Select **Bypass VLAN** from the Actions menu.
- Select **Route VLAN** from the Actions menu.
- Click **Yes** to bypass the gateway.

After bypassing the network gateway, all front-end and back-end traffic routes through the FCR and BCR associated with the VLAN. The VLAN remains associated with the gateway appliance and can be routed back to the gateway appliance at any time.

Disassociating a VLAN from a gateway appliance

VLANs can be linked to one gateway appliance at a time through association. Association allows the VLAN to be routed to the gateway appliance at any time. If a VLAN should be associated to another gateway appliance, or if the VLAN should no longer be associated to its gateway, disassociation is required. Disassociation removes the "link" from the VLAN to the gateway appliance, allowing it to be associated to another gateway, if necessary.

Bypassing a VLAN allows the VLAN to remain associated to the network gateway. If the VLAN should no longer be associated with the gateway appliance, see Disassociating a VLAN from a gateway appliance.

Perform the following procedure to disassociate a VLAN from a gateway appliance:

- Access the Gateway Appliance Details page in the IBM Cloud console.
- Select the VLAN you want from the **Associate a VLAN** list.
- Select **Disassociate** from the Actions menu.
- Select **Route VLAN** from the Actions menu.
- Click **Yes** to disassociate the VLAN.

After disassociating a VLAN from a gateway appliance, the VLAN can be associated to another gateway. The VLAN can also be associated back to the gateway appliance at any time. After disassociating a VLAN from a gateway appliance, the VLAN's traffic cannot be routed through the gateway. VLANs must be associated to a gateway appliance before they can be routed.

Working with the vSRX Default Configurations

IN THIS SECTION

Understanding the vSRX default configuration

IN THIS SECTION

IBM Cloud™ Juniper vSRX devices come with following default configuration:

- SSH and Ping are permitted on both vSRX public and private gateway IP addresses
- Juniper Web Management (J-Web) UI access is permitted on HTTPS port 8443 for both public and private gateway IP addresses
- An address-set SERVICE is predefined for IBM service networks
- Two security zones: SL-PRIVATE and SL-PUBLIC are predefined.
- Access from the zone SL-PRIVATE to all services is provided by IBM and address-set SERVICE is permitted

- All other network accesses are denied

Two redundancy groups are configured are illustrated below:

Redundancy group	Redundancy group function
redundancy-group 0	Redundancy group for control plane
redundancy-group 1	Redundancy group for data plane

Priority in the redundancy group decides which vSRX node is active. By default, node 0 is active for both control plane and data plane.

Reference Default Configuration Samples

- Default Configuration of a sample 1G Standalone SR-IOV Public and Private vSRX Gateway
- Default Configuration of a sample 10G HA SR-IOV Public and Private vSRX Gateway

Importing and Exporting a vSRX Configuration

IN THIS SECTION

The IBM Cloud™ Juniper vSRX upgrade process preserves the original configuration of the vSRX throughout the entire process, as long as the required reloads are done one at a time. However, it is still strongly recommended to export and backup your vSRX configuration settings before starting the upgrade.

After the upgrade process completes for stand alone servers, you should import the original configuration you saved if you want to restore it. For High Availability configurations, you should restore the configuration manually from your exported file only if the upgrade fails or if moving between architectures. For more information on migrating 1G configurations from the legacy architecture to the current architecture, see Migrating legacy configurations to the current vSRX architecture.

Considerations

- The upgrade process for Standalone and High Availability (HA) are different. See Upgrading the vSRX.
- The J-Web interface allows you to display, edit, and upload the current configuration quickly and easily without using the Junos OS CLI. See J-Web for SRX Series Documentation for more details.
- An upgrade from the vSRX 15.1 release to a newer vSRX release, such as 19.4, results in changes to the vSRX interface mappings in the configuration file. As a result, when importing your original vSRX settings, make sure that the new “interfaces” section is not modified. There are two ways of doing this: Either import sub-sections other than the “interfaces” section, or import the entire configuration and manually restore the 19.4 SR-IOV interfaces.

The new vSRX default interface configuration for both the Linux Bridge and SR-IOV must be preserved after the import of their configurations. For example, for SR-IOV the GE interfaces have specific mappings to the host that must be preserved to enable SR-IOV. These interfaces are found in the CLI using the command show configuration interfaces. See vSRX default configurations section for more information on SR-IOV mappings. See Migrating legacy configurations to the current vSRX architecture for details on migrating 1G configurations from the legacy architecture to the current architecture.

If you prefer using the Junos OS CLI, the following contents provide different methods to export and import your configuration settings, depending on whether you want to export or import the entire configuration or just part of it. To manage the configuration settings, enter CLI mode, then run the command configure to enter configuration mode. Then to commit your changes, run the command commit.

Exporting part of the vSRX configuration

To export only part of the vSRX configuration:

1. Enter configuration mode and ensure you are at the top of the configuration tree: edit then top
2. Then run the `show <section>` command to get the current configuration, enclosed in braces.

   For example, you can run show interfaces to show all the interfaces configuration. Or, if you prefer to display the output in set mode, run the `show <section> | display set` command.

   The output should be similar to the following:

```
# show interfaces | display set
set interfaces ge-0/0/0 description PRIVATE_VLANs
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 native-vlan-id 925
set interfaces ge-0/0/0 mtu 9000
```

```
...

[edit]
```

TIP: Set mode displays the configuration as a series of configuration mode commands required to re-create the configuration. This is useful if you are not familiar with how to use configuration mode commands or if you want to cut, paste, and edit the displayed configuration.

3. Copy and save the output into your local workspace for later use.

Importing the entire vSRX configuration

The new vSRX default interface configuration for both the Linux Bridge and SR-IOV must be preserved after the import of their configurations. For example, for SR-IOV the GE interfaces have specific mappings to the host that must be preserved to enable SR-IOV. These interfaces are found in the CLI using the `show configuration interfaces` command. For more information on SR-IOV mappings, see vSRX default configuration.

To import the entire vSRX configuration:

1. After upgrading the vSRX, copy the config file you saved earlier back to the /var/tmp folder.
2. Run load override /var/tmp/backup.txt under the configuration mode to replace the entire current configuration with the content that you saved under the /var/tmp folder.

Importing part of the vSRX configuration

The new vSRX default interface configuration for both the Linux Bridge and SR-IOV must be preserved after the import of their configurations. For example, for SR-IOV the GE interfaces have specific mappings to the host that must be preserved to enable SR-IOV. These interfaces are found in the CLI using the `show configuration interfaces` command. For more information on SR-IOV mappings, see vSRX default configuration.

To import only part of the vSRX configuration:

1. From the configuration mode, run `edit <section>` to go to the configuration tree level that you want.
2. Copy the configuration settings you have saved and run the command load merge terminal relative to merge the configuration with the current one.
3. Paste the content, hit Enter to go to a new line, then type Control + D to end the input.

The output should be similar to the following:

```
# load merge terminal relative
[Type ^D at a new line to end input]
family inet {
           filter {
               input PROTECT-IN;
           }
       }
load complete

[edit interfaces lo0 unit 0]
```

Alternatively, you can also:

1. Replace the configuration instead of merging it, by deleting the configuration first with the command delete under this configuration tree level and then performing a load merge terminal relative to copy and paste your previous configuration.
2. Edit the configuration in set mode, by running load set terminal instead of load merge terminal relative. Then copy and paste the content you saved in set mode.

> **NOTE**: Ensure that you always run the `load set terminal` at the top.

Migrating Legacy Configurations to the Current vSRX Architecture

IN THIS SECTION

Migrating IBM Cloud™ Juniper vSRX configurations from the legacy to the current architecture requires careful consideration.

vSRX 18.4 deployments leverage the current architecture in most cases. This includes the vSRX 18.4 1G SR-IOV offering. The older vSRX 18.4 1G Standard offering is based on Linux Bridging and has different network configurations on the Ubuntu host, the KVM hypervisor, and in the vSRX configuration. The host and KVM settings do not require any special migration steps, as the automation process handles the configuration changes. However, if you want to import the vSRX configuration from the legacy architecture into the current vSRX configuration, you likely need to refactor some of the configuration.

Migrating 1G vSRX Standalone Configurations

IN THIS SECTION

There are some steps you potentially need to convert vSRX configuration settings on a Standalone 18.4 1G Public+Private Linux Bridge (legacy architecture) instance to a Standalone 18.4 1G Public+Private SR-IOV (current architecture) instance. You can find a sample default configuration for SR-IOV based current architecture Default Configuration of a sample 1G Standalone SR-IOV Public and Private vSRX Gateway.

The following is a sample default configuration for the Linux Bridge (legacy architecture). The example shows vSRX instances that were provisioned in different Datacenter pods. As a result, the transit VLAN's (native-vlan-id) are different.

```
## Last commit: 2020-04-16 22:48:33 UTC by root
version 18.4R1-S1.3;
system {
    login {
        class security {
            permissions [ security-control view-configuration ];
        }
        user admin {
            uid 2000;
            class super-user;
            authentication {
                encrypted-password "$6$vKPIcB3I
$XlDRg3Oto9tLa7zRPkalSfonrKUEJI7U16XX2lrke3k2sPaV.CY0CJhSBIPx5aXhqo7h1GWPhhMbv0Ce1WANO."; ##
SECRET-DATA
```

```
            }
        }
    }
    root-authentication {
        encrypted-password "$6$cbXBMc8b
$jHd6LtR4OjXvjmgubQXAlofNonk6lLbNPs35beda7ffEV4XKEUQiEf1XUA3mMvJv2V1YET3kiWBogqz8h2zB7."; ##
SECRET-DATA
    }
    services {
        ssh {
            root-login allow;
        }
        netconf {
            ssh {
                port 830;
            }
        }
        web-management {
            http {
                interface fxp0.0;
            }
            https {
                port 8443;
                system-generated-certificate;
                interface [ fxp0.0 ge-0/0/0.0 ge-0/0/1.0 ];
            }
            session {
                session-limit 100;
            }
        }
    }
    host-name asloma-e2e-tc15-18-1g-1270-sa-vsrx-vSRX;
    name-server {
        10.0.80.11;
        10.0.80.12;
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
```

```
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    ntp {
        server 10.0.77.54;
    }
}
security {
    log {
        mode stream;
        report;
    }
    address-book {
        global {
            address SL8 10.1.192.0/20;
            address SL9 10.1.160.0/20;
            address SL4 10.2.128.0/20;
            address SL5 10.1.176.0/20;
            address SL6 10.1.64.0/19;
            address SL7 10.1.96.0/19;
            address SL1 10.0.64.0/19;
            address SL2 10.1.128.0/19;
            address SL3 10.0.86.0/24;
            address SL20 10.3.80.0/20;
            address SL18 10.2.176.0/20;
            address SL19 10.3.64.0/20;
            address SL16 10.2.144.0/20;
            address SL17 10.2.48.0/20;
            address SL14 10.1.208.0/20;
            address SL15 10.2.80.0/20;
            address SL12 10.2.112.0/20;
            address SL13 10.2.160.0/20;
            address SL10 10.2.32.0/20;
            address SL11 10.2.64.0/20;
            address SL_PRIV_MGMT 10.129.33.87/32;
            address SL_PUB_MGMT 161.202.136.77/32;
            address-set SERVICE {
                address SL8;
                address SL9;
                address SL4;
                address SL5;
```

```
                address SL6;
                address SL7;
                address SL1;
                address SL2;
                address SL3;
                address SL20;
                address SL18;
                address SL19;
                address SL16;
                address SL17;
                address SL14;
                address SL15;
                address SL12;
                address SL13;
                address SL10;
                address SL11;
            }
        }
    }
    screen {
        ids-option untrust-screen {
            icmp {
                ping-death;
            }
            ip {
                source-route-option;
                tear-drop;
            }
            tcp {
                syn-flood {
                    alarm-threshold 1024;
                    attack-threshold 200;
                    source-threshold 1024;
                    destination-threshold 2048;
                    queue-size 2000; ## Warning: 'queue-size' is deprecated
                    timeout 20;
                }
                land;
            }
        }
    }
    policies {
        from-zone SL-PRIVATE to-zone SL-PRIVATE {
```

```
            policy Allow_Management {
                match {
                    source-address any;
                    destination-address [ SL_PRIV_MGMT SERVICE ];
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone SL-PUBLIC to-zone SL-PUBLIC {
            policy Allow_Management {
                match {
                    source-address any;
                    destination-address SL_PUB_MGMT;
                    application [ junos-ssh junos-https junos-http junos-icmp-ping ];
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone SL-PRIVATE {
            interfaces {
                ge-0/0/0.0 {
                    host-inbound-traffic {
                        system-services {
                            all;
                        }
                    }
                }
            }
        }
        security-zone SL-PUBLIC {
            interfaces {
                ge-0/0/1.0 {
                    host-inbound-traffic {
                        system-services {
                            all;
                        }
```

```
                    }
                }
            }
        }
    }
}
interfaces {
    ge-0/0/0 {
        description PRIVATE_VLANs;
        flexible-vlan-tagging;
        native-vlan-id 1214;
        unit 0 {
            vlan-id 1214;
            family inet {
                address 10.129.33.87/26;
            }
        }
    }
    ge-0/0/1 {
        description PUBLIC_VLAN;
        flexible-vlan-tagging;
        native-vlan-id 764;
        unit 0 {
            vlan-id 764;
            family inet {
                address 161.202.136.77/29;
            }
            family inet6 {
                address 2401:c900:1001:0210:0000:0000:0000:000a/64;
            }
        }
    }
    fxp0 {
        unit 0;
    }
    lo0 {
        unit 0 {
            family inet {
                filter {
                    input PROTECT-IN;
                }
                address 127.0.0.1/32;
            }
```

```
        }
    }
}
firewall {
    filter PROTECT-IN {
        term PING {
            from {
                destination-address {
                    161.202.136.77/32;
                    10.129.33.87/32;
                }
                protocol icmp;
            }
            then accept;
        }
        term SSH {
            from {
                destination-address {
                    161.202.136.77/32;
                    10.129.33.87/32;
                }
                protocol tcp;
                destination-port ssh;
            }
            then accept;
        }
        term WEB {
            from {
                destination-address {
                    161.202.136.77/32;
                    10.129.33.87/32;
                }
                protocol tcp;
                port 8443;
            }
            then accept;
        }
        term DNS {
            from {
                protocol udp;
                source-port 53;
            }
            then accept;
```

```
        }
    }
}
routing-options {
    static {
        route 166.9.0.0/16 next-hop 10.129.33.65;
        route 0.0.0.0/0 next-hop 161.202.136.73;
        route 161.26.0.0/16 next-hop 10.129.33.65;
        route 10.0.0.0/8 next-hop 10.129.33.65;
    }
}
```

Converting the Interface Section

In the above 1G Public+Private Standalone example, the current architecture adds aggregated interfaces ae0 and ae1. These should map to what the legacy architecture defines as ge-0/0/0 (private / ae0) and ge-0/0/1 (public / ae1). Additionally, the new architecture adds ge-0/0/2 and ge-0/0/3 to support redundancy within the vSRX interfaces. In the old architecture, redundancy existed at the host (Hypervisor) bond interfaces (bond0 private / bond1 public). In the current architecture, SR-IOV VF's that map directly to the ge interfaces are used for redundancy.

You can compare these vSRX configuration differences in vSRX Standalone interface (current architecture) and vSRX Standalone interface (legacy architecture).

Any private VLAN's that were previously configured for ge-0/0/0 need to be routed through ae0. In addition, any public VLAN's that you previously configured for ge-0/0/1 need to be routed through ae1.

Converting the Zones Section

Any default security zones that previously referenced ge-0/0/0 and ge-0/0/1 should now use the ae0.0 (SL-PRIVATE) and ae1.0 (SL-PUBLIC) interfaces. The same changes also apply to any zones that previously referenced ge-0/0/0 and ge-0/0/1.

Other Changes

- The aggregated device configuration requires the following addition in the current architecture:

  ```
  set chassis aggregated-devices ethernet device-count 10
  ```

- The JWEB configuration will also include the aggregated interfaces as well:

  ```
  set system services web-management https interface ae1.0
  ```

  ```
  set system services web-management https interface ae0.0
  ```

Migrating 1G vSRX High Availability configurations

For High Availability configurations, the main vSRX changes when importing configurations from the legacy architecture to the current architecture are small changes to the interface mappings.

The 1G SR-IOV HA configuration for the current architecture adds additional vSRX interfaces for redundancy, instead of using the host (hypervisor) bond interfaces. This is possible as the host now uses SR-IOV VF's that can be mapped directly to the vSRX interfaces. Configurations that were exported from the legacy architecture will need to take this into account if they are imported into the current architecture.

For vSRX configuration for the current architecture for 1G HA, see vSRX High Availability interfaces (current architecture) and for vSRX configuration for the legacy architecture for 1G HA, see vSRX High Availability interfaces (legacy architecture).

The extra ge-0/* and ge-7/* interfaces were added and associated with the existing reth interfaces which have been present in both the legacy and current architecture. These allow for redundancy within the vSRX configuration. Redundancy is also configured for the fab interfaces as well.

Allowing SSH and Ping to a Public Subnet

IN THIS SECTION

Allowing SSH and Ping to a Public Subnet

In this topic, learn how to configure the IBM Cloud™ Juniper vSRX Standard with a new interface, zone, and address-book. As the default action for all traffic is to drop, this guide shows how to set up traffic flows that allow all traffic within the new zone, all traffic from the new zone to the internet, and allow only SSH and ping from the internet to one subnet on the new VLAN.

In this example, the values used are - Public vlan: 1523 Public subnet: 169.47.211.152/29.

NOTE: This step-by-step assumes that a high-availability deployment of the vSRX, with a single Public VLAN and subnet.

Follow the steps listed to configure the service:

Task	Description
Create a new interface, zone, and address-book subnet	Create the tagged interface unit and security zone for the new VLAN.
Creating your new traffic flows	Create the new traffic flows to allow inbound pinging and SSH.
Confirming the output and committing the changes	Check the output to confirm what will be committed to the active configuration.

Performing vSRX Advanced Tasks in IBM Cloud

IN THIS SECTION

Working with Firewalls

The IBM Cloud™ Juniper vSRX uses the concept of security zones, where each vSRX interface is mapped to a "zone" for handling stateful firewalls. Stateless firewalls are controlled by firewall filters.

Policies are used to allow and block traffic between these defined zones, and the rules defined here are stateful.

In the IBM Cloud, a vSRX is designed to have four different security zones:

Zone	Standalone Interface	HA Interface
SL-Private (untagged)	ge-0/0/0.0 or ae0.0	reth0.0
SL-Public (untagged)	ge-0/0/1.0 or ae1.0	reth1.1
Customer-Private (tagged)	ge-0/0/0.1 or ae0.1	reth2.1
Customer-Public (tagged)	ge-0/0/1.1 or ae1.1	reth3.1

Zone Policies

Following are some of the attributes that can be defined in your policies:

- Source addresses
- Destination addresses
- Applications
- Action (permit/deny/reject/count/log)

Since this is a stateful operation, there is no need to allow return packets (in this case, the echo replies).

To configure a stateful firewall, follow these steps:

1. Create security zones and assign the respective interfaces:

   Standalone scenario:

   set security zones security-zone CUSTOMER-PRIVATE interfaces ge-0/0/0.1

   set security zones security-zone CUSTOMER-PUBLIC interfaces ge-0/0/1.1

   High Availability scenario:

   set security zones security-zone CUSTOMER-PRIVATE interfaces reth2.1

   set security zones security-zone CUSTOMER-PUBLIC interfaces reth2.1

2. Define the policy and rules between two different zones.

   The following example illustrates pinging traffic from the zone Customer-Private to Customer-Public:

   set security policies from-zone CUSTOMER-PRIVATE to-zone CUSTOMER-PUBLIC policy

set security policies from-zone CUSTOMER-PRIVATE to-zone CUSTOMER-PUBLIC policy

3. Use the following commands to allow traffic that is directed to the vSRX:
   - Standalone scenario:

     set security zones security-zone CUSTOMER-PRIVATE interfaces ge-0/0/0.0 host-inbound-traffic system-services all

   - High Availability scenario:

     set security zones security-zone CUSTOMER-PRIVATE interfaces reth2.0 host-inbound-traffic system-services all

4. To allow protocols, such as OSPF or BGP, use the following command:
   - Standalone scenario:

     set security zones security-zone trust interfaces ge-0/0/0.0 host-inbound-traffic protocols all

   - High Availability scenario:

     set security zones security-zone trust interfaces reth2.0 host-inbound-traffic protocols all

Firewall Filters

By default the IBM Cloud™ Juniper vSRX allows ping, SSH, and HTTPS to itself and drops all other traffic by applying the PROTECT-IN filter to the lo interface.

To configure a new stateless firewall, follow these steps:

1. Create the firewall filter and term (the following filter allows only ICMP and drops all other traffic)

   set firewall filter ALLOW-PING term ICMP from protocol icmp

   set firewall filter ALLOW-PING term ICMP then accept

2. Apply the filter rule to the interface (the following command applies the filter to all private network traffic)

   set interfaces ge-0/0/0 unit 0 family inet filter input ALLOW-PING

Working with sNAT

You can refer a sample configuration for sNAT on a vSRX appliance, where a private node routed behind the Gateway can communicate with the outside world at Working with sNAT

To configure NAT for the IBM Cloud™ Juniper vSRX, see Network Address Translation User Guide on the Juniper website.

Working with Failover

You can initiate failover from your primary IBM Cloud™ Juniper vSRX to a backup device, so that all control and data plane traffic is routed through the secondary gateway device after failover.

NOTE: This section is only applicable if your Juniper vSRX gateway devices are provisioned in High-Availability mode.

Perform the following procedure:

1. Login to your primary vSRX gateway device.
2. Enter CLI mode by running the command cli at the console prompt. When you enter CLI mode, the console displays the node role, either primary or secondary.
3. On the primary vSRX gateway device, run the command:

   show chassis cluster status

```
Monitor Failure codes:
     CS  Cold Sync monitoring        FL  Fabric Connection monitoring
     GR  GRES monitoring             HW  Hardware monitoring
     IF  Interface monitoring        IP  IP monitoring
     LB  Loopback monitoring         MB  Mbuf monitoring
     NH  Nexthop monitoring          NP  NPC monitoring
     SP  SPU monitoring              SM  Schedule monitoring
     CF  Config Sync monitoring

 Cluster ID: 2
 Node   Priority Status         Preempt Manual       Monitor-failures

 Redundancy group: 0 , Failover count: 1
 node0  100      primary        no      no       None
 node1  1        secondary      no      no       None
 Redundancy group: 1 , Failover count: 1
 node0  100      primary        yes     no       None
 node1  1        secondary      yes     no       None

 {primary:node0}
```

   Ensure that, for both redundancy groups, the same node is set as primary. It is possible for different nodes to be set as the primary role in different redundancy groups.

NOTE: The vSRX, by default, sets Preempt to yes for Redundancy group 1, and no for Redundancy group 0. Refer to this link to learn more about pre-emption and failover behavior.

4. Initiate failover by running the following command in the console prompt:

   `request chassis cluster failover redundancy-group <redundancy group number> node <node number>`

   Select the appropriate redundancy group number and node number from the output of the command in step two. To failover both redundancy groups, execute the previous command twice, one for each group.

5. After failover is complete, verify the console output. It should now be listed as secondary.

6. Login to the other vSRX gateway of your pair. Enter into CLI mode by again executing the command cli and then verify that the console output shows as primary.

TIP: When you enter CLI mode in your Juniper vSRX gateway device, the output will show as primary from the control plane perspective. Always check the show chassis cluster status output to determine which gateway device is primary from data plane perspective. Refer to vSRX Default Configuration to learn more about redundancy groups, as well as the control and data planes.

Working with Routing

The IBM Cloud™ Juniper vSRX is based on JunOS, giving you access to the full Juniper routing stack.

- **Static routing**—To configure static routes, run the following commands:

  Setting a default route—`set routing-options static route 0/0 next-hop <Gateway IP>`

- **Creating a static route**—Run the `set routing-options static route <PREFIX/MASK> next-hop <Gateway IP>`

- **Basic OSPF routing**—To setup basic OSPF routing, only using area 0, run the following commands using md5 authentication using the `set protocols ospf area 0 interface ge-0/0/1.0 authentication md5 0 key <key>` command.

- **Basic BGP routing**

  - To setup basic BGP routing, first define the local AS by running the `set routing-options autonomous-system 65001` command.

  - Then configure the BGP neighbor and its session attributes:

    ```
    set protocols bgp group CUSTOMER local-address 1.1.1.1
    ```

```
set protocols bgp group CUSTOMER family inet unicast
set protocols bgp group CUSTOMER family inet6 unicast
set protocols bgp group CUSTOMER peer-as 65002
set protocols bgp group CUSTOMER neighbor 2.2.2.2
```

In this example, BGP is configured for the following:

- To use source IP address of 1.1.1.1 to establish the session
- To negotiate both ipv4 and ipv6 unicast families
- To peer with a neighbor that belongs to AS 65002
- Peer neighbor IP 2.2.2.2

For more configurations, see Junos OS Documentation

Working with VPN

IN THIS SECTION

This topic details a sample configuration for a Route based VPN between two sites. In this sample configuration Server 1 (Site A) can communicate with Server 2 (Site B), and each site utilizes two phase IPSEC authentication. For more information see Working with VPN and

Sample configuration for Site A (Dallas):

```
# show security address-book global address Network-A
10.84.237.200/29;
[edit]
# show security address-book global address Network-B
10.45.53.48/29;
# show security ike
proposal IKE-PROP {
    authentication-method pre-shared-keys;
```

```
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
    lifetime-seconds 3600;
}
policy IKE-POL {
    mode main;
    proposals IKE-PROP;
    pre-shared-key ascii-text "$9$ewkMLNs2aikPdbkP5Q9CKM8"; ## SECRET-DATA
}
gateway IKE-GW {
    ike-policy IKE-POL;
    address 158.100.100.100;
    external-interface ge-0/0/1.0;
}
#show security ipsec
proposal IPSEC-PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
    lifetime-seconds 3600;
}
policy IPSEC-POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC-PROP;
}
vpn IPSEC-VPN {
    bind-interface st0.1;
    vpn-monitor;
    ike {
        gateway IKE-GW;
        ipsec-policy IPSEC-POL;
    }
    establish-tunnels immediately;
}
#show interfaces
ge-0/0/0 {
    description PRIVATE_VLANs;
    flexible-vlan-tagging;
    native-vlan-id 1121;
    unit 0 {
```

```
        vlan-id 1121;
        family inet {
            address 10.184.108.158/26;
        }
    }
    unit 10 {
        vlan-id 1811;
        family inet {
            address 10.184.237.201/29;
        }
    }
    unit 20 {
        vlan-id 1812;
        family inet {
            address 10.185.48.9/29;
        }
    }
}
st0 {
    unit 1 {
        family inet {
            address 169.254.200.0/31;
        }
    }
#show security policies
from-zone CUSTOMER-PRIVATE to-zone VPN {
    policy Custprivate-to-VPN {
        match {
            source-address any;
            destination-address Network-B;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone VPN to-zone CUSTOMER-PRIVATE {
    policy VPN-to-Custprivate {
        match {
            source-address Network-B;
            destination-address any;
            application any;
```

```
        }
        then {
            permit;
        }
    }
```

Sample configuration for Site B (London):

```
# show interfaces
ge-0/0/0 {
    description PRIVATE_VLANs;
    flexible-vlan-tagging;
    native-vlan-id 822;
    unit 0 {
        vlan-id 822;
        family inet {
            address 10.45.165.140/26;
        }
    }
    unit 10 {
        vlan-id 821;
        family inet {
            address 10.45.53.49/29;
        }
    }
}
st0 {
    unit 1 {
        family inet {
            address 169.254.200.1/31;
        }
    }
#show security ike
proposal IKE-PROP {
    authentication-method pre-shared-keys;
    dh-group group5;
    authentication-algorithm sha1;
    encryption-algorithm aes-128-cbc;
    lifetime-seconds 3600;
}
policy IKE-POL {
```

```
    mode main;
    proposals IKE-PROP;
    pre-shared-key ascii-text "$9$H.fz9A0hSe36SevW-dk.P"; ## SECRET-DATA
}
gateway IKE-GW {
    ike-policy IKE-POL;
    address 169.100.100.100;
    external-interface ge-0/0/1.0;
}
# show security ipsec
proposal IPSEC-PROP {
    protocol esp;
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm aes-128-cbc;
    lifetime-seconds 3600;
}
policy IPSEC-POL {
    perfect-forward-secrecy {
        keys group5;
    }
    proposals IPSEC-PROP;
}
vpn IPSEC-VPN {
    bind-interface st0.1;
    vpn-monitor;
    ike {
        gateway IKE-GW;
        ipsec-policy IPSEC-POL;
    }
    establish-tunnels immediately;
}
#show security zone security-zone CUSTOMER_PRIVATE
security-zone CUSTOMER-PRIVATE {
    interfaces {
        ge-0/0/0.10 {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
        }
    }
}
```

```
security-zone VPN {
    interfaces {
        st0.1;
    }
}
#show security policies from-zone CUSTOMER-PRIVATE to-zone VPN
policy Custprivate-to-VPN {
    match {
        source-address any;
        destination-address Network-A;
        application any;
    }
    then {
        permit;
    }
}
 #show security zones security-zone VPN
interfaces {
    st0.1;
}
#show security policies from-zone VPN to-zone CUSTOMER-PRIVATE
policy VPN-to-Custprivate {
    match {
        source-address Network-A;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
```

Performance Consideration

In order to achieve the best IPSEC VPN performance, use AES-GCM as the encryption algorithm for both IKE and IPSEC proposals.

For example:

```
set security ike proposal IKE-PROP encryption-algorithm aes-128-gcm

set security ipsec proposal IPSEC-PROP encryption-algorithm aes-128-gcm
```

With AES-GCM as the encryption algorithm, you don't need to specify the authentication algorithm in the same proposal. AES-GCM provides both encryption and authentication.

For more information on VPN configurations, see IPsec VPN User Guide for Security Devices and Example: Configuring a Route-Based VPN

Securing the Host Operating System

IN THIS SECTION

The IBM Cloud™ Juniper vSRX runs as a Virtual Machine on a bare-metal server installed with Ubuntu and KVM. To secure the host OS, you should ensure that no other critical services are hosted on the same OS.

SSH Access

The IBM Cloud™ Juniper vSRX can be deployed with public and private network access or private network access only. By default, password based SSH access to the public IP of the host OS will be disabled on new provisions and OS reloads. Access to the host can be achieved through the private IP address. Alternatively, key based authentication can be used to access the public IP. To do so, specify the public SSH key when placing a new Gateway order.

Some existing deployments of the IBM Cloud™ Juniper vSRX may allow password based SSH access to the public IP of the host OS. For these deployments, you can manually disable password based SSH access to the public IP of the OS by following these steps:

1. Modify /etc/ssh/sshd_config

   - Ensure the following values are set.

```
ChallengeResponseAuthentication no
PasswordAuthentication no
```

- Add the following filter rules to the end of the file.

```
Match Address 10.0.0.0/8
   Password Authentication yes
```

2. Restart the SSH service using the command **/usr/sbin/service ssh restart**.

   The procedure above ensures addresses in the private infrastructure network 10.0.0.0/8 subnet are allowed SSH access. This access is needed for actions such as: OS reloads, Cluster rebuilding, Version upgrades.

Firewalls

Implementing an Ubuntu firewall (UFW, Iptables, and so on) without required rules can cause the vSRX HA cluster to be disabled. The vSRX solution depends on heartbeat communication between the primary and secondary nodes. If the firewall rules do not allow communication between the nodes, then cluster communication will be lost.

The vSRX architecture influences the firewall rules discussed below. Details on the two architectures can be found in vSRX default configuration.

For vSRX version 18.4 HA deployments running with the legacy architecture, the following rules are required to allow cluster communication for UFW:

1. To allow protocol 47 (used for heartbeat communication) in /etc/ufw/before.rules:

   -A ufw-before-input -p 47 -j ACCEPT

2. To allow private network communication:

   ufw allow in from 10.0.0.0/8 to 10.0.0.0/8

3. To enable UFW:

   ufw enable

For vSRX versions running with the newer architecture, the firewall rules must allow multicast communication.

NOTE: In some cases, troubleshooting operations may require disabling the firewall for access to public repositories. In these cases, you should work with IBM Support to understand how to proceed.

Most Gateway actions require SSH access to the private 10.0.0.0/8 subnet for the host OS and the vSRX. Blocking this access with a firewall can cause the following actions to fail: OS reloads, Cluster rebuilding, and Version upgrades

As a result, if SSH access is disabled for the 10.0.0.0/8 subnet, you must re-enable it prior to executing any of these actions.

Configuring the Management Interfaces

The IBM Cloud™ Juniper vSRX nodes provide built-in management interfaces ("fxp0") that are not configured by default. When configured, these private interfaces can be used to communicate with the individual node, which might be useful in a high availability cluster for monitoring the status of the secondary node over SSH, ping, SNMP, and so on. Since the private IP for the vSRX floats to the primary node, it is not possible to directly access the secondary node.

Configuration of the fxp0 interface requires IPs in a subnet that is attached to the private transit VLAN for the gateway. Although the primary subnet that comes with the gateway has IPs that might be available, it is not recommended for this use. This is because the primary subnet is reserved for the gateway provisioning infrastructure, and IP collisions could occur if additional gateways are deployed in the same pod.

You can allocate a secondary subnet for the private transit VLAN, and use IPs from this subnet to configure fxp0 and the host bridge interface for PING and SSH access. To do so, perform the following procedure:

1. Order a portable private subnet and assign it to the vSRX private transit VLAN. You can find the private transit VLAN on the gateway details page.

   NOTE: Ensure the subnet includes at least 8 addresses in order to support 2 IPs for the host bridge interfaces, and 2 IPs for the vSRX fxp0 interfaces.

2. Configure the host br0:0 bridge interfaces using 2 IPs from the new subnet. For example:

   On Ubuntu host 0: ifconfig br0:0 10.177.75.140 netmask 255.255.255.248

   On Ubuntu host 1: ifconfig br0:0 10.177.75.141 netmask 255.255.255.248

3. Persist the bridge interface configurations across reboots by modifying /etc/network/interfaces on each Ubuntu host. For example:

```
auto br0:0
iface br0:0 inet static
address 10.177.75.140
netmask 255.255.255.248
post-up /sbin/ifconfig br0:0 10.177.75.140 netmask 255.255.255.248
```

4. Assign the 2 IP's to the vSRX fxp0 interface and create backup router configurations for access to the secondary node's fxp0 interface:

```
set groups node0 interfaces fxp0 unit 0 family inet address 10.177.75.138/29
set groups node1 interfaces fxp0 unit 0 family inet address 10.177.75.139/29
set groups node0 system backup-router 10.177.75.137 destination [ 0.0.0.0/1 128.0.0.0/1 ]
set groups node1 system backup-router 10.177.75.137 destination [ 0.0.0.0/1 128.0.0.0/1 ]
```

> **NOTE**: Additional information on configuring the backup router can be found in this Juniper article at: KB17161.

5. Create a static route to the subnet. For example:

```
set routing-options static route 10.177.75.136/29 next-hop 10.177.75.137
```

6. Create firewall filters to allow PING and SSH to the fxp0 management interfaces:

```
set firewall filter PROTECT-IN term PING from destination-address 10.177.75.136/29

set firewall filter PROTECT-IN term SSH from destination-address 10.177.75.136/29
```

Upgrading the vSRX in IBM Cloud

IN THIS SECTION

Upgrading

There are several methods and considerations that you must understand before upgrading your IBM Cloud® Juniper vSRX:

- vSRX version level

- Bare-metal server processor model
- Bandwidth: 1G versus 10G
- Stand-alone or High Availability (HA)

Using these factors, the following table lists whether you can use the OS reload option to upgrade your vSRX. The table also describes whether rollback is supported for the upgrade. Additional considerations include whether you need a manual vSRX configuration migration to complete the upgrade.

Reference the following table to determine if you can upgrade your vSRX using OS reload. For more information, see General upgrade considerations.

For more information on the vSRX versions listed below, see IBM Cloud Juniper vSRX supported versions.

Current vSRX Version	Processor Model and Speed	Stand-Alone or HA	Upgrade method	Rollback supported
15.1	1270v6 (All 1G deployments)	Stand-alone and HA	Not Supported	N/A
15.1	All 10G Deployments	Stand-alone and HA	Upgrading using OS Reload	Stand-alone: No HA: • Manual (not automated) rollbacks are allowed after the first server completes the OS reload. • Rollbacks are not allowed after the second server completes its OS reload.
18.4	1270v6 (Some 1G Deployments)	Stand-alone and HA	Not Supported	N/A

(Continued)

Current vSRX Version	Processor Model and Speed	Stand-Alone or HA	Upgrade method	Rollback supported
18.4	4210 (Some 1G Deployments)	Stand-alone	Upgrading using OS Reload	No
18.4	4210 (Some 1G Deployments)	HA	Upgrading using OS Reload	• Yes – If you are running version 18.4 with new architecture, manual (not automated) rollbacks are allowed after the first server completes the OS reload. For more information, see Rollback Options. • No – If you are running version 18.4 without new architecture.
18.4	All 10G Deployments	Stand-alone	Upgrading using OS Reload	No

(Continued)

Current vSRX Version	Processor Model and Speed	Stand-Alone or HA	Upgrade method	Rollback supported
18.4	All 10G Deployments	HA	Upgrading using OS Reload	• Yes – If you are running version 18.4 with new architecture, manual (not automated) rollbacks are allowed after the first server completes the OS reload. For more information, see Rollback Options. • No – If you are running version 18.4 without new architecture.
19.4 and newer	All 1G and 10G Deployments	Stand-alone and HA	Upgrading using OS Reload	Yes – Manual (not automated) rollbacks are allowed after the first server completes the OS reload. For more information, see Rollback Options.

General Upgrade Considerations

Before you perform a vSRX upgrade, be aware of the following considerations:

- You might experience network disruptions when upgrading your vSRX version. To avoid disruptions, perform the upgrade during a maintenance window that supports potential network downtime. Failover is not available until the upgrade completes, and can take several hours. For High Availability (HA) environments, your vSRX configuration settings are migrated; however, it is recommended to export your settings before the upgrade.
- For a stand-alone environment, the previous configuration is not restored, so you should export and import your configuration. For more information, see Importing and exporting a vSRX configuration.

- For a successful reload on a HA vSRX, the root password for the provisioned vSRX gateway must match the root password that is defined in the vSRX portal. In addition, you must enable root SSH login to the vSRX Private IP.

  NOTE: You defined the password in the portal when you provisioned your gateway. This might not match the current gateway password. If the password was changed after provisioning, then use SSH to connect to the vSRX gateway and change the root password to match. The Readiness Check fails if there is a password mismatch.

- Do not modify the vSRX configuration during an OS reload. The upgrade process captures a snapshot of the current vSRX cluster configuration at the beginning of the process. Therefore, modifying the vSRX configuration during the upgrade process can result in a failure, or unpredictable results. For example, automated software agents attempting to modify one or both vSRX nodes. Configurations changes can corrupt the OS reload process. Additionally, these configuration changes are not preserved if a rollback is initiated.

- Before performing an OS reload upgrade on an HA cluster, run the command show chassis cluster status. The nodes should be clustered with one node that is listed as the primary and the other as secondary. Ensure that there are no monitor failures. If the cluster is not healthy prior to the upgrade, then the upgrade can fail, causing an extended traffic outage.

  Example of a healthy cluster:

```
root@asloma-19-10g-ha1-vsrx-vSRX-Node0> show chassis cluster status
  Monitor Failure codes:
    CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring      RE  Relinquish monitoring
    IS  IRQ storm

  Cluster ID: 2
  Node   Priority Status               Preempt Manual   Monitor-failures

  Redundancy group: 0 , Failover count: 1
  node0  100      primary              no      no       None
  node1  1        secondary            no      no       None
```

```
  Redundancy group: 1 , Failover count: 1
  node0  100      primary              no      no       None
  node1  1        secondary            no      no       None

  {primary:node0}
Example of an unhealthy cluster with monitor failures:
root@asloma-tc11-15-10g-pubpriv-ha1-vsrx-vSRX-Node1> show chassis cluster status
Monitor Failure codes:
  CS  Cold Sync monitoring        FL  Fabric Connection monitoring
  GR  GRES monitoring             HW  Hardware monitoring
  IF  Interface monitoring        IP  IP monitoring
  LB  Loopback monitoring         MB  Mbuf monitoring
  NH  Nexthop monitoring          NP  NPC monitoring
  SP  SPU monitoring              SM  Schedule monitoring
  CF  Config Sync monitoring
Cluster ID: 3
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  0        lost           n/a     n/a      n/a
node1  1        primary        no      no       None

Redundancy group: 1 , Failover count: 1
node0  0        lost           n/a     n/a      n/a
node1  0        primary        no      no       CS

{primary:node1}
```

- If your IBM Cloud account has multiple vSRX gateway instances in the same pod, make sure that only one gateway is upgraded at a time. Upgrading more than one vSRX at a time can result in IP collisions, disrupt the upgrade process, and potentially cause failures.
- For HA clusters, the upgrade process requires you to disable the vSRX Chassis Cluster preemption flag for Redundancy Group 1. Therefore, after the upgrade completes, the flag is disabled, but you can enable again. Run show chassis cluster status to view the preempt setting.

Upgrading using OS Reload

IN THIS SECTION

To upgrade your vSRX using OS reload, perform the following procedure.

1. Standalone environment only: See Exporting part of the vSRX configuration.
2. Access the gateway details page, see Viewing gateway appliance details.
3. Run a readiness check for “OS reload”. See Checking vSRX readiness and address any errors that are found.
4. Perform an OS reload for each bare metal server. See Performing an OS reload.

> **NOTE**: When upgrading an HA cluster, the process will power off the node not undergoing the OS reload at the end of the upgrade process. This will transition the cluster’s primary node and any active network traffic to the newly upgraded one. Once the OS reload completes for the first node in the cluster, it is critical that the second node be left unpowered until the OS reload to upgrade that node is submitted and running. Powering the node on prior to the OS reload will cause the cluster to run with mismatched vSRX versions, potentially leading to a “split-brain” scenario where each node tries to claim primary ownership. This generally results in an outage. After the OS reload of the first node, the gateway will transition to “Upgrade Active” status.

5. Standalone environment only: Import the vSRX configuration and migrate the settings to the new architecture if necessary.

vSRX Migration Configuration Considerations

For a High Availability environment, the upgrade restores the previous vSRX configuration. No further steps are needed.

For a Standalone environment, the upgrade does not restore the previous configuration, so you should export and import your configuration. See Importing and and exporting a vSRX Configuration for more information.

Additionally, when migrating from an older version, such as 15.1, your interface mappings may have changed. This requires some modifications to the vSRX configuration after the import. See Migrating 1G vSRX standalone configurations for more information.

Rollback Options

In the standalone environment a rollback is not supported.

In the high availability environment that is upgrading from vSRX version, a rollback is supported only after the first node has been OS Reloaded and before the second node has been OS Reloaded. The

Gateway will be in an “Upgrade Active” state at this point. The following steps should be followed to rollback the first node to the previous version.

NOTE: Please be aware that a traffic disruption will occur while waiting for the secondary node to power on and for the traffic to failover to this node.

1. Power off the vSRX on the node being rolled-back (primary node) using the command virsh shutdown <domain> Wait for the node to be fully powered off before proceeding.
2. Power up the vSRX on the node that has not been rolled-back using the command virsh start <domain>. Doing so will return the primary node back to the original vSRX version.

   Before restoring the original vSRX image, rename the vSRX qcow2 file in /var/lib/libvirt/images/vSRXvM2/vSRX_Image.qcow2.backup to /var/lib/libvirt/images/vSRXvM2/vSRX_Image.qcow2 so that virsh detects the original image.
3. Run the OS reload readiness checks, see Checking vSRX readiness if necessary, and resolve any issues.
4. Perform an OS reload on the host you want to rollback to return it to the original vSRX version.

The cluster will now be running with its original configuration.

Unsupported Upgrades

Early deployments of 1G vSRX 15.1 and 18.4 gateways used a networking design based on Linux Bridging. Newer 1G deployments use a networking design based on SR-IOV. For more information, see Understanding the vSRX default configuration.

Early 1G deployments generally used an Intel 1270v6 4-Core, Sky-Lake-based processor. This processor does not support SR-IOV. Newer vSRX versions, such as 19.4, require the SR-IOV networking design. Therefore, vSRX version upgrades are not supported for deployments based on this 1270v6 processor.

To upgrade to a newer vSRX version, such as 19.4, you must place a new vSRX order. After completion, you can migrate the configuration from the old design to the new, but you must also apply some manual configuration changes to the new vSRX. For more information, see Migrating Legacy Configurations to the Current vSRX Architecture.

CHAPTER 29

Managing vSRX in IBM Cloud

IN THIS CHAPTER

vSRX Configuration and Management Tools

SUMMARY

This topic provides an overview of the various tools available to configure and manage a vSRX VM once it has been successfully deployed.

IN THIS SECTION

Understanding the Junos OS CLI and Junos Scripts

Junos OS CLI is a Juniper Networks specific command shell that runs on top of a UNIX-based operating system kernel.

Built into Junos OS, Junos script automation is an onboard toolset available on all Junos OS platforms, including routers, switches, and security devices running Junos OS (such as a vSRX instance).

You can use the Junos OS CLI and the Junos OS scripts to configure, manage, administer, and troubleshoot vSRX.

Understanding the J-Web Interface

The *J-Web* interface allows you to monitor, configure, troubleshoot, and manage vSRX instances by means of a Web browser. J-Web provides access to all the configuration statements supported by the vSRX instance.

Understanding Junos Space Security Director

As one of the Junos Space Network Management Platform applications, Junos Space Security Director helps organizations improve the reach, ease, and accuracy of security policy administration with a scalable, GUI-based management tool. Security Director automates security provisioning of a vSRX instance through one centralized Web-based interface to help administrators manage all phases of the security policy life cycle more quickly and intuitively, from policy creation to remediation.

RELATED DOCUMENTATION

- CLI User Interface Overview
- J-Web Overview
- Security Director
- Mastering Junos Automation Programming
- Spotlight Secure Threat Intelligence

Managing Security Policies for Virtual Machines Using Junos Space Security Director

SUMMARY

This topic provides you an overview of how you can manage security policies for VMs using security director.

Security Director is a Junos Space management application designed to enable quick, consistent, and accurate creation, maintenance, and application of network security policies for your security devices, including vSRX instances. With Security Director, you can configure security-related policy management including IPsec VPNs, firewall policies, NAT policies, IPS policies, and UTM policies. and push the

configurations to your security devices. These configurations use objects such as addresses, services, NAT pools, application signatures, policy profiles, VPN profiles, template definitions, and templates. These objects can be shared across multiple security configurations; shared objects can be created and used across many security policies and devices. You can create these objects prior to creating security configurations.

When you finish creating and verifying your security configurations from Security Director, you can publish these configurations and keep them ready to be pushed to all security devices, including vSRX instances, from a single interface.

The Configure tab is the workspace where all of the security configuration happens. You can configure firewall, IPS, NAT, and UTM policies; assign policies to devices; create and apply policy schedules; create and manage VPNs; and create and manage all the shared objects needed for managing your network security.

RELATED DOCUMENTATION

- Security Director

CHAPTER 30

Monitoring and Troubleshooting

IN THIS CHAPTER

Technical Support

SUMMARY

IN THIS SECTION

Getting Help and Support Information

This topic provides you details on getting technical assistance.

If you have problems or questions when using IBM Cloud Gateway Appliance (vSRX), you can search for information or ask questions by using Stack Overflow. Or post your question, then tag it with "vsrx" and "ibm-cloud".

For any technical assistance , contact IBM customer support team and then IBM team will raise tickets with Juniper JTAC. Do not raise Juniper help desk tickets directly.

For information about opening an IBM Support case, or about support levels and case severities, see Contacting Support.

RELATED DOCUMENTATION

- Junos OS Documentation
- SRX Firewall Features - User Guides

10

PART

vSRX Deployment for OCI

CHAPTER 31

Overview

IN THIS CHAPTER

Understanding vSRX Deployment in Oracle Cloud Infrastructure

IN THIS SECTION

Overview of Oracle VM Architecture

This section provides you information on the Oracle VM architecture.

Oracle Cloud Infrastructure (OCI) is a set of complementary cloud services that enable you to build and run a wide range of applications and services in a highly available hosted environment. Oracle Cloud Infrastructure offers high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premises network.

Oracle virtual machine (VM) management platform provides a fully equipped environment with all the latest benefits of virtualization technology. Oracle VM platform helps you deploy operating systems and application software within a supported virtualization environment. Oracle VM can support both 1G and 10G physical NICs.

vSRX 3.0 VM can be deployed on Oracle VM server running on X86 hardware.

vSRX with Oracle Cloud Infrastructure

vSRX 3.0 specifications for deployment in OCI are: vSRX3.0 has one RE, one virtual FPC slot, and one virtual PIC. The virtual Gigabit Ether ports (labeled as “ge-0/0/[0 – (n-1)] will be within the one PIC. The index is zero-based. Number n depends on hypervisor. The maximum number of interfaces supported on vSRX are 7.

A domain is a configurable set of resources, including memory, virtual CPUs, network devices and disk devices, in which virtual machines run. A user-domain (domU) is granted virtual resources and can be started, stopped and restarted independently of other domains and of the host server itself. vSRX as a guest virtualized operating system runs within a domain. Oracle vSRX VM guests consume resources that are allocated to the domain by the hypervisor running on the Oracle VM Server. For more information about the Oracle VM Guest Additions, see Installing and Using the Oracle VM Guest Additions.

When a virtual machine is running, it can be accessed through a console, which allows it to be used as a regular operating system. vSRX as a guest virtualized operating system runs within a VM.

SEE ALSO

- How are Network Functions Separated in Oracle VM

OCI Glossary

This section defines some common terms used in Oracle Cloud Infrastructure (OCI) configuration. Table 91 on page 604 provides a list of the common terms used in OCI.

Table 91: OCI VCN Related Terminology

Term	Description
OCI	Oracle Cloud Infrastructure, which is running Xen Hypervisor.
Oracle VM Server	A managed virtualization environment providing a lightweight, secure, server platform which runs virtual machines, also known as domains.
Oracle VM Manager	Used to manage Oracle VM Servers, virtual machines, and resources. It is comprised of a number of subcomponents, including a web browser-based user interface; and a command line interface (CLI).

Table 91: OCI VCN Related Terminology *(Continued)*

Term	Description
Oracle Compute Shapes	A shape is a resource profile that specifies the number of OCPUs and the amount of memory to be allocated to an instance in Compute Classic.
Port	The network interface on a server. This term is used interchangeably with NIC (Network Interface Card).
VLAN	A method used to virtualize networking at the switch or router for better control over network separation. VLANs are virtual networks that use identifiers to separate traffic into different networks within the switch.
VNIC	Virtual machines are assigned VNICs or virtual network interface cards, which are allocated faux MAC addresses. This allows each virtual machine to connect to a network. The VNICs are bridged interfaces that are connected to a logical network that has the Virtual Machine channel enabled. A VNIC is only ever assigned to a virtual machine. A virtual machine can have as many VNICs as required within the limitations posed by the virtualization method used. For instance, hardware virtualized virtual machines are able to support a limited number of VNICs, while paravirtualized virtual machines can have an unlimited number of VNICs.

Requirements for vSRX on Oracle Cloud Infrastructure

IN THIS SECTION

This topic provides the requirements for deploying vSRX instances on Oracle Cloud Infrastructure (OCI).

Minimum System Requirements for OCI

Table 92 on page 606 lists the minimum system requirements for vSRX instances to be deployed on OCI.

Table 92: Minimum System Requirements for vSRX

Component	Specification and Details
Memory	4 GB
Disk space	16 GB

Oracle pre-defined VM shapes that vSRX support are listed below. If you need any other VM shapes, then please contact your Juniper sales representive.

Table 93: OCI VM Shapes Supported by vSRX

Shape	OCPU	Memory (GB)	Local Disk (TB)	Network Bandwidth	Max VNICs Total: Linux
VM.Standard2.4	4	60	Block Storage only	4.1 Gbps	4
VM.Standard2.8	8	120	Block Storage only	8.2 Gbps	8

Interface Mapping for vSRX on OCI: The first network interface is used for the out-of-band management (fxp0) for vSRX.

We recommend putting revenue interfaces in routing instances as a best practice to avoid asymmetric traffic/routing, because fxp0 is part of the default (inet.0) table by default. With fxp0 as part of the default routing table, there might be two default routes needed: one for the fxp0 interface for external management access, and the other for the revenue interfaces for traffic access. Putting the revenue interfaces in a separate routing instance avoids this situation of two default routes in a single routing instance.

NOTE: Ensure that interfaces belonging to the same security zone are in the same routing instance. See KB Article - Interface must be in the same routing instance as the other interfaces in the zone.

vSRX Default Settings with OCI

Do not use the `load factory-default` command on a vSRX OCI instance. The factory-default configuration removes the OCI preconfiguration. If you must revert to factory default, ensure that you manually reconfigure preconfiguration statements before you commit the configuration; otherwise, you will lose access to the vSRX instance. See *Configure vSRX Using the CLI* for preconfiguration details.

Best Practices for Deploying vSRX

Refer the following best practices for deploying vSRX:

- Disable the source/destination check for all vSRX interfaces.
- Limit public key access permissions to 400 for key pairs.
- Ensure that there are no contradictions between OCI security groups and your vSRX configuration.

CHAPTER 32

Installing vSRX in OCI

IN THIS CHAPTER

vSRX Deployment in Oracle Cloud Infrastructure

IN THIS SECTION

The topics in this section help you launch vSRX instances in Oracle Cloud Infrastructure.

Overview

IN THIS SECTION

This topic provides you an overview and pre-requisites to deploy vSRX virtual Firewall in Oracle Cloud Infrastructure. vSRX provides security and networking services for virtualized private or public Oracle Cloud environments.

Starting in Junos OS Release 20.4R2, vSRX 3.0 is available for OCI deployments.

NOTE: vSRX 3.0 image is not available in the OCI Marketplace. You must download the vSRX 3.0 software from Juniper Support Downloads and upload into an OCI compartment.

Pre-Requisites

- Ensure you have proper accounts and permissions before you attempt to deploy the vSRX in OCI.
- Copy the .oci image to an object storage compartment in your OCI account.

  An example file name is junos-vsrx3-x86-64-xxxx.oci. After you purchase the vSRX 3.0 software you can downloaded the software from: Juniper Support page.

  NOTE: .oci image extensions are built for the vSRX images to be deployed in OCI. This is because on OCI, when the qcow2 images are deployed, the default emulation selected for the vNIC is e-1000. The .oci images of the vSRX pass the metadata needed for the emulation type to be set to virtIO upon deployment of the vSRX which ensure a better throughput.

- Create Virtual Network subnets for your deployment.

For better understanding of Oracle terminologies and their use in vSRX 3.0 deployments, see "Understanding vSRX Deployment in Oracle Cloud Infrastructure" on page 603.

Example Topology

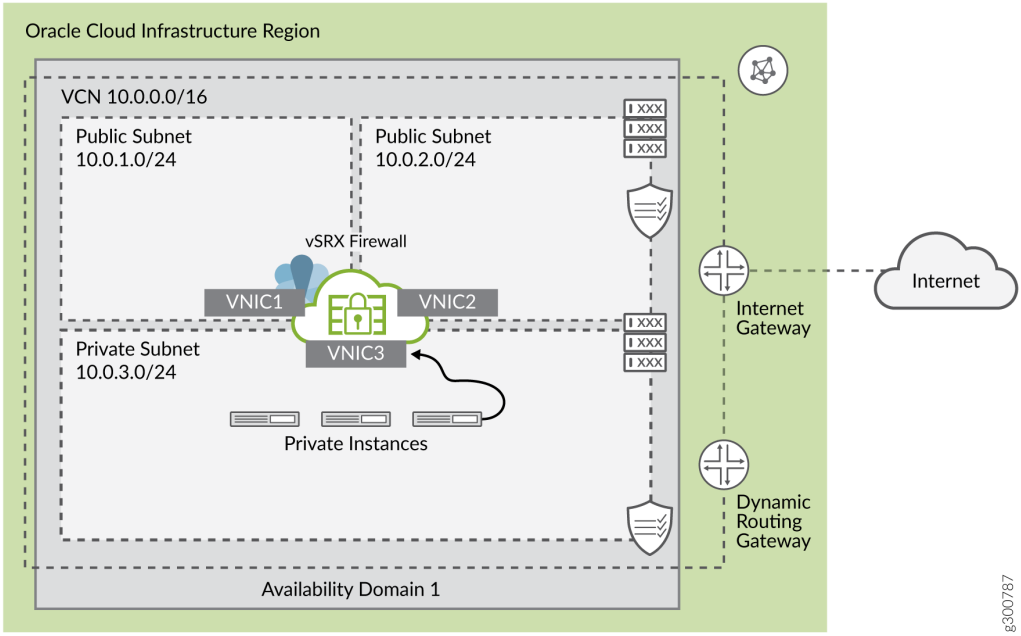
A common cloud configuration includes hosts that you want to grant access to the Internet, but you do not want anyone from outside your cloud to get access to your hosts. You can use vSRX in the OCI to NAT traffic inside the OCI from the public Internet.

The diagram shows an example VCN with three subnets:

- Public (10.0.1.0/24), for management interfaces with access to the internet through an internet gateway
- Public (10.0.2.0/24), for revenue (data) interfaces with access to the internet through an internet gateway
- Private (10.0.3.0/24), a private subnet with no access to the internet

The following topology is used as an example for this deployment.

Figure 136: Example VCN for vSRX Deployment in OCI



Launch vSRX Instances in the OCI

This topic provides details on how you can launch vSRX instances in the OCI.

1. Log in to the OCI Management Console. The Console is an intuitive, graphical interface that lets you create and manage your instances, cloud networks, and storage volumes, as well as your users and permissions. After you sign in, the console home page is displayed.
2. Choose a compartment for your resources.

   Compartments help you organize and control access to your resources. A compartment is a collection of related resources (such as cloud networks, compute instances, or block volumes) that can be accessed only by those groups that have been given permission by an administrator in your organization. For example, one compartment could contain all the servers and storage volumes that make up the production version of your company's Human Resources system. Only users with permission to that compartment can manage those servers and volumes.

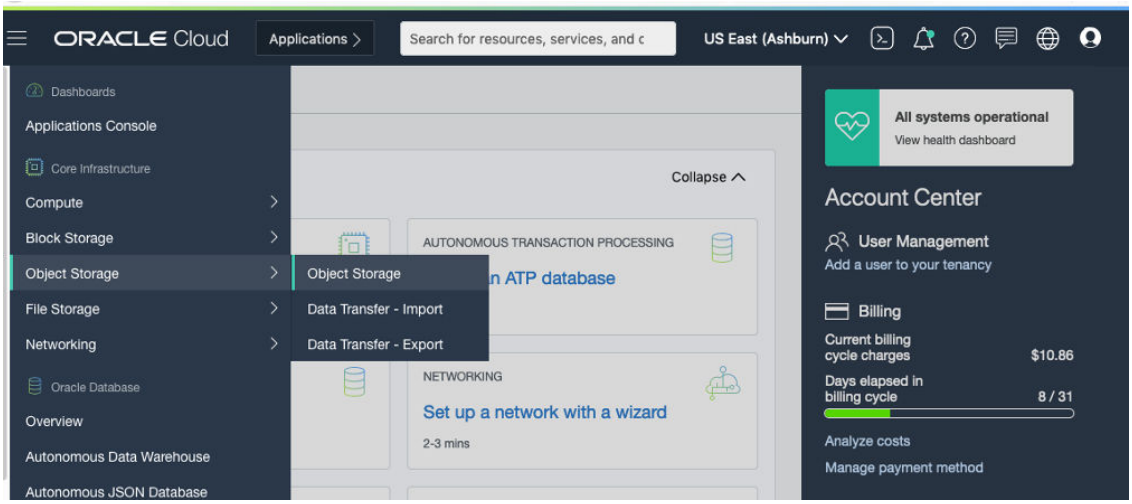
   - Open the navigation menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.

- Select the Sandbox compartment (or the compartment designated by your administrator) from the list on the left. If the Sandbox compartment does not exist, you can create. For more information, see Creating a Compartment.

3. Load the .oci onto OCI platform.

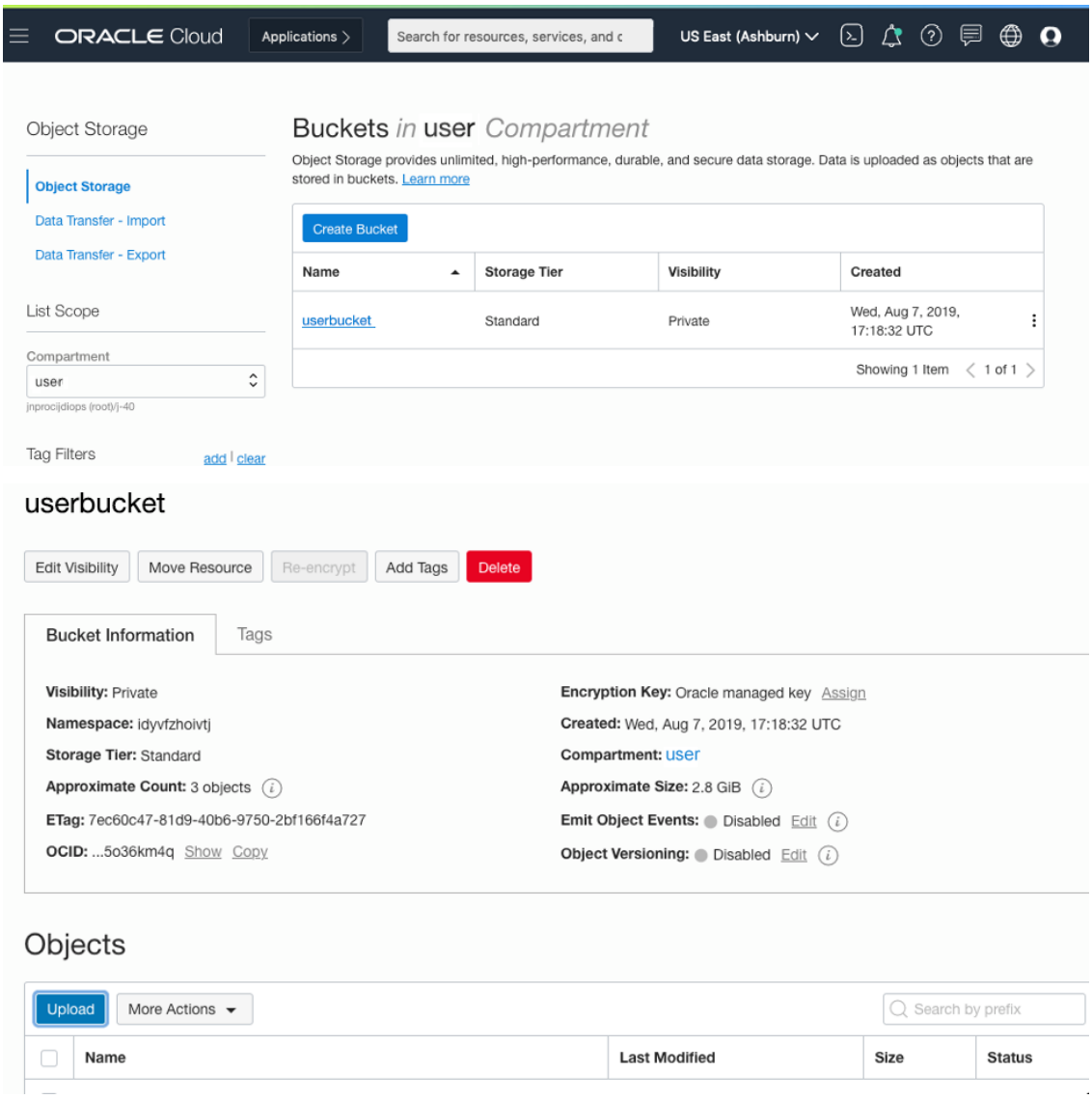
   a. From the main menu click **Object Storage**.

Figure 137: Object Storage



b. Select the compartment in which you want to create the bucket. If you have a bucket already, click the name of “your bucket”. Or create a bucket.

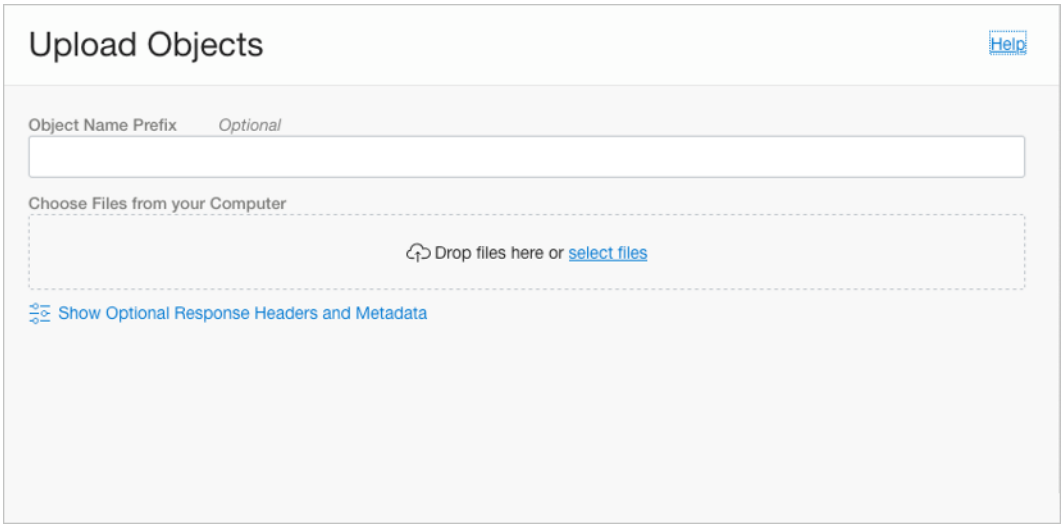
Figure 138: Create Bucket



c. Then Click **Upload Objects**.

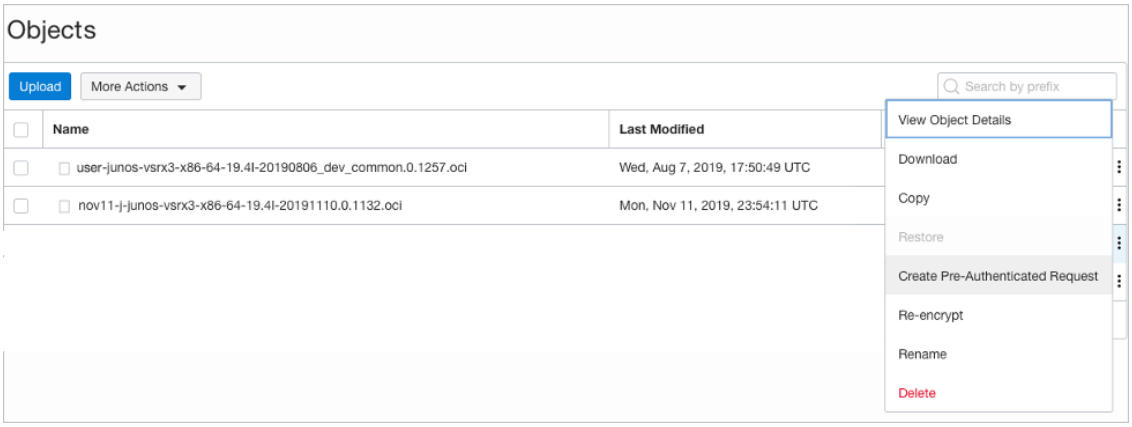
Provide the required information when a pop-up window appears.

Figure 139: Upload Objects



View Object Details: After the .oci image is loaded, choose the object right click the object and select **View Object Details**.

Figure 140: View Object Details



NOTE: There will be an URL path for this object as OCI ID, which can be used in the during importing images.

4. Create a virtual cloud network (VCN) with subnets. Multiple subnets within a single VCN network is possible.

You will then launch your instance into one of the subnets of your VCN and connect to it.

> **NOTE**: Ensure that the Sandbox compartment (or the compartment designated for you) is selected in the Compartment list on the left.

a. Open the **Navigation** menu. Under **Core Infrastructure**, go to **Networking** and click **Virtual Cloud Networks**.

b. Click **Create VCN** and enter the data for VCN Name, Compartment, select an IPv4 VCN CIDR Block, Public Subnet CIDR Block. Accept the defaults for any other fields and click **Create VCN**.

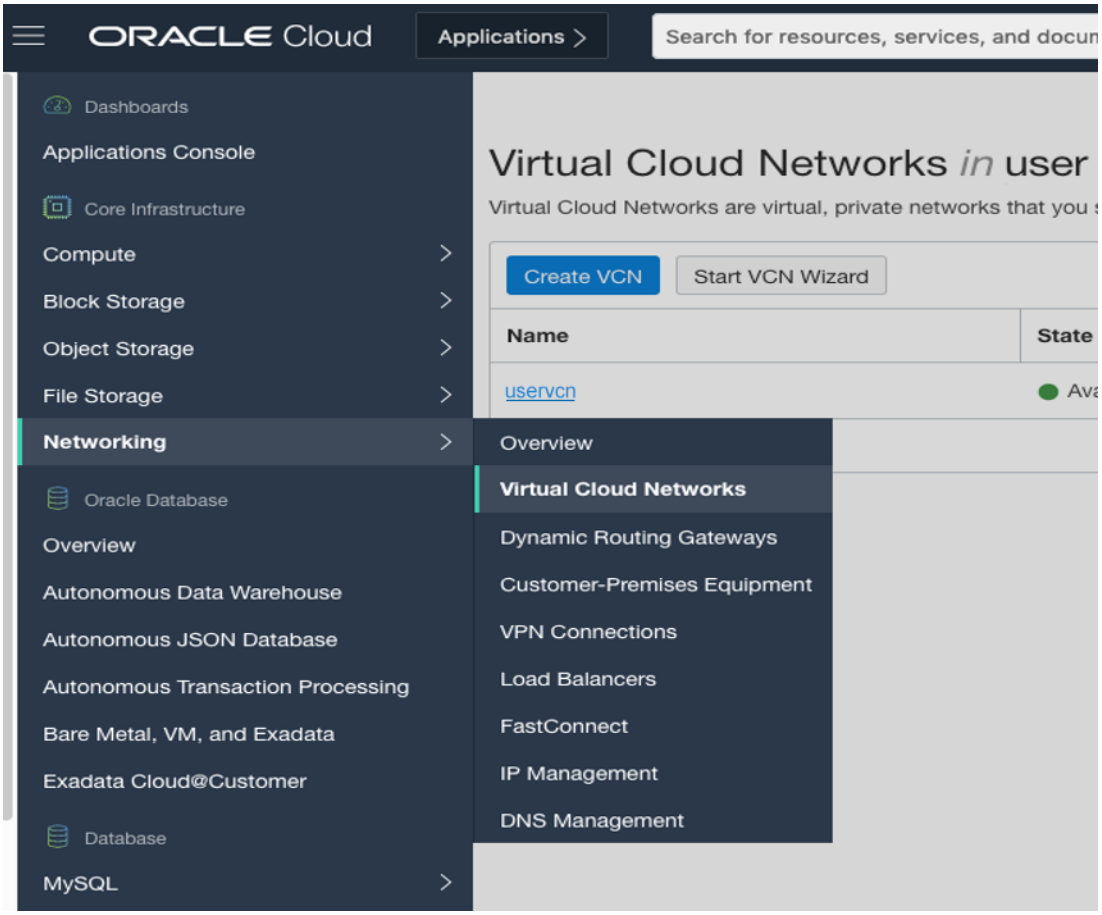

ORACLE Cloud
Applications
Search for resources, services, and docun
Dashboards
Applications Console
Core Infrastructure
Compute
Block Storage
Object Storage
File Storage
Networking
Oracle Database
Overview
Autonomous Data Warehouse
Autonomous JSON Database
Autonomous Transaction Processing
Bare Metal, VM, and Exadata
Exadata Cloud@Customer
Database
MySQL
Overview
Virtual Cloud Networks
Dynamic Routing Gateways
Customer-Premises Equipment
VPN Connections
Load Balancers
FastConnect
IP Management
DNS Management
Virtual Cloud Networks in user
Virtual Cloud Networks are virtual, private networks that you
Create VCN
Start VCN Wizard
Name
State
uservcn

Figure 141: Create Virtual Cloud Network

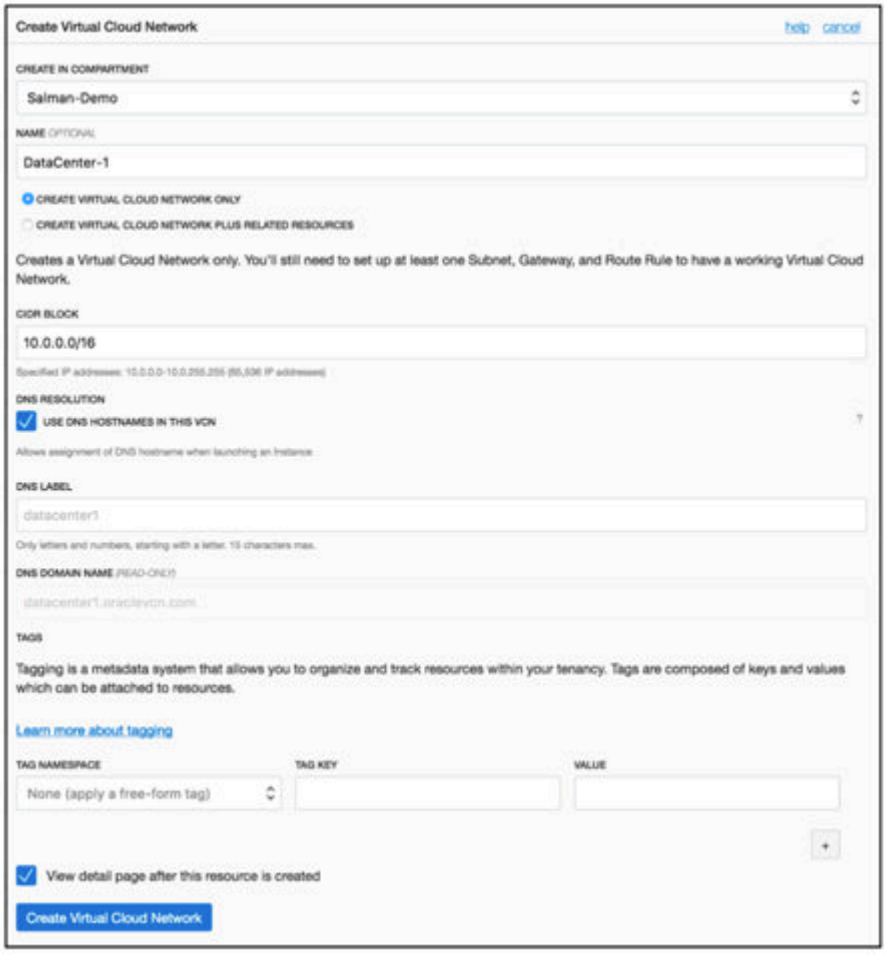
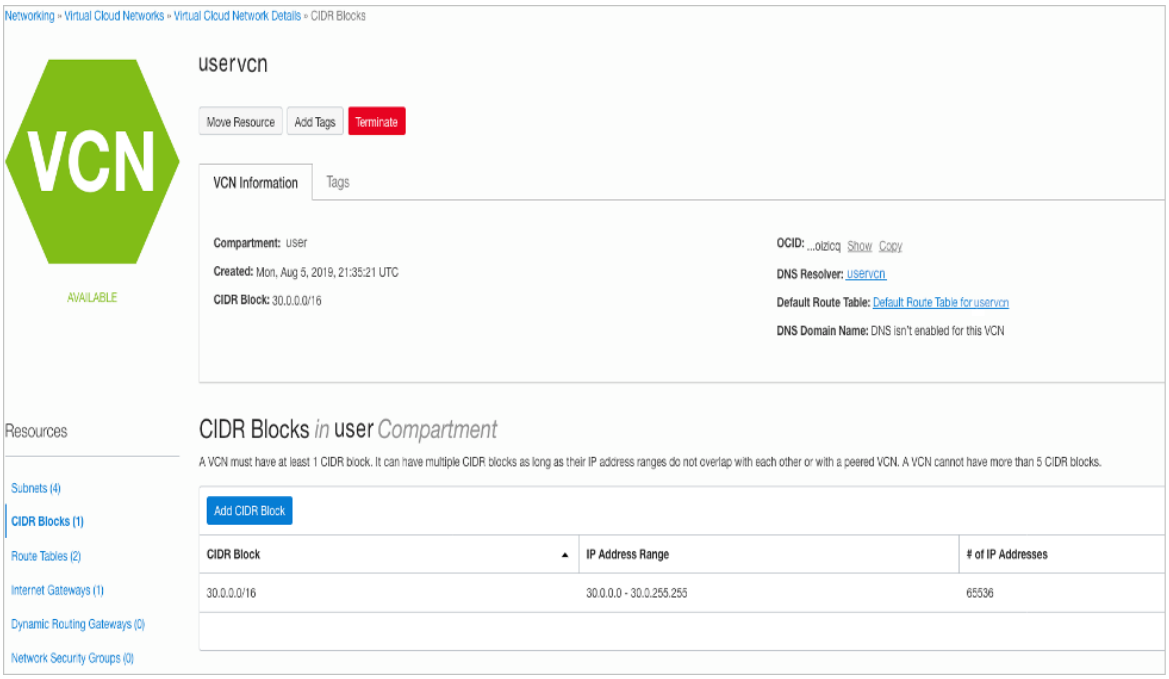


Figure 142: CIDR Block



The cloud network created will have resources such as Internet and NAT gateway, Service gateway with access to the Oracle Services Network, A regional public subnet with access to the internet gateway, and A regional private subnet with access to the NAT gateway and service gateway.

5. Create Subnets for the vSRX VCN created.

   vSRX requires two public subnets and one or more private subnets for each individual instance group. One public subnet is for the management interface (fxp0), and the other is for a revenue (data) interface. The private subnets, connected to the other vSRX interfaces, ensure that all traffic between applications on the private subnets and the internet must pass through the vSRX instance.

   a. Configure the Public Subnet (Management Interface)

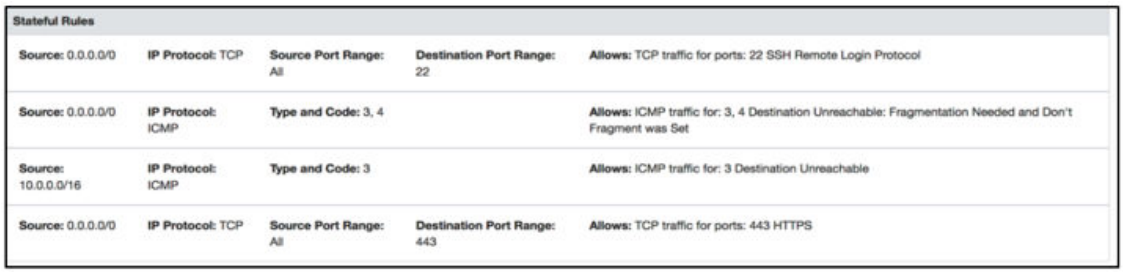
      To create this public subnet, click **Create Subnet** and define a route rule for the route table Default Route Table in which the internet gateway is configured as the route target for all traffic (0.0.0.0/0) as shown below.

Figure 143: Route Rules

For details about how to create subnets, see VCNs and Subnets.

For the subnet's security list Default Security List, create an egress rule to allow traffic to all destinations. Create ingress rules that allow access on TCP port 22 from the public internet and on TCP port 80/443 for accessing the web application from the public internet as shown below.

Figure 144: Stateful Rules (Default Security List)



b. Configure the Public Subnet (Revenue Interface)

Create this public subnet, and define a route rule for the route table Public RT in which the internet gateway is configured as the route target for all traffic (0.0.0.0/0).

For the subnet's security list Public Subnet SL, create an egress rule to allow traffic to all destinations. Create ingress rules that allow access on TCP port 80/443 for accessing the web application from the public internet and on ICMP if needed to check the connectivity as shown below.

Figure 145: Stateful Rules (Public Subnet Security List)

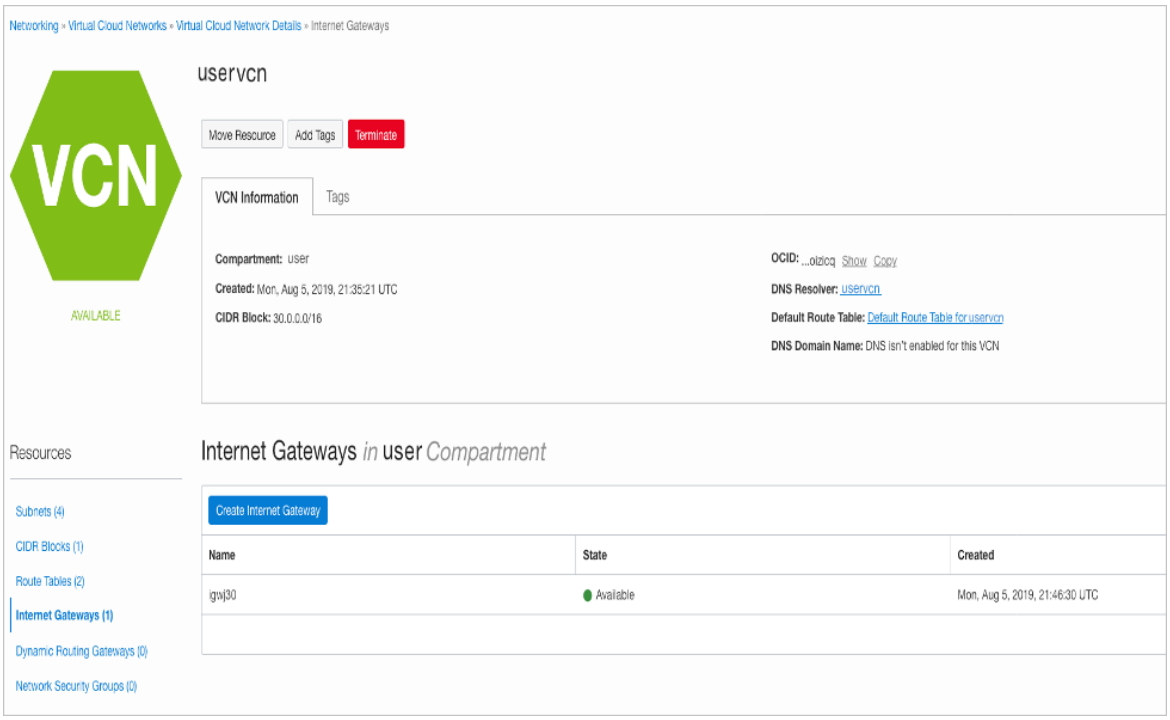
c. Configure the Private Subnet

Create this private subnet, and define a route rule for the route table Private RT in which the vSRX second vNIC's private IP address (10.0.3.3) is configured as the route target for all traffic 0.0.0.0/0.

NOTE: Configure the route rule after you create and attach the secondary VNICs.

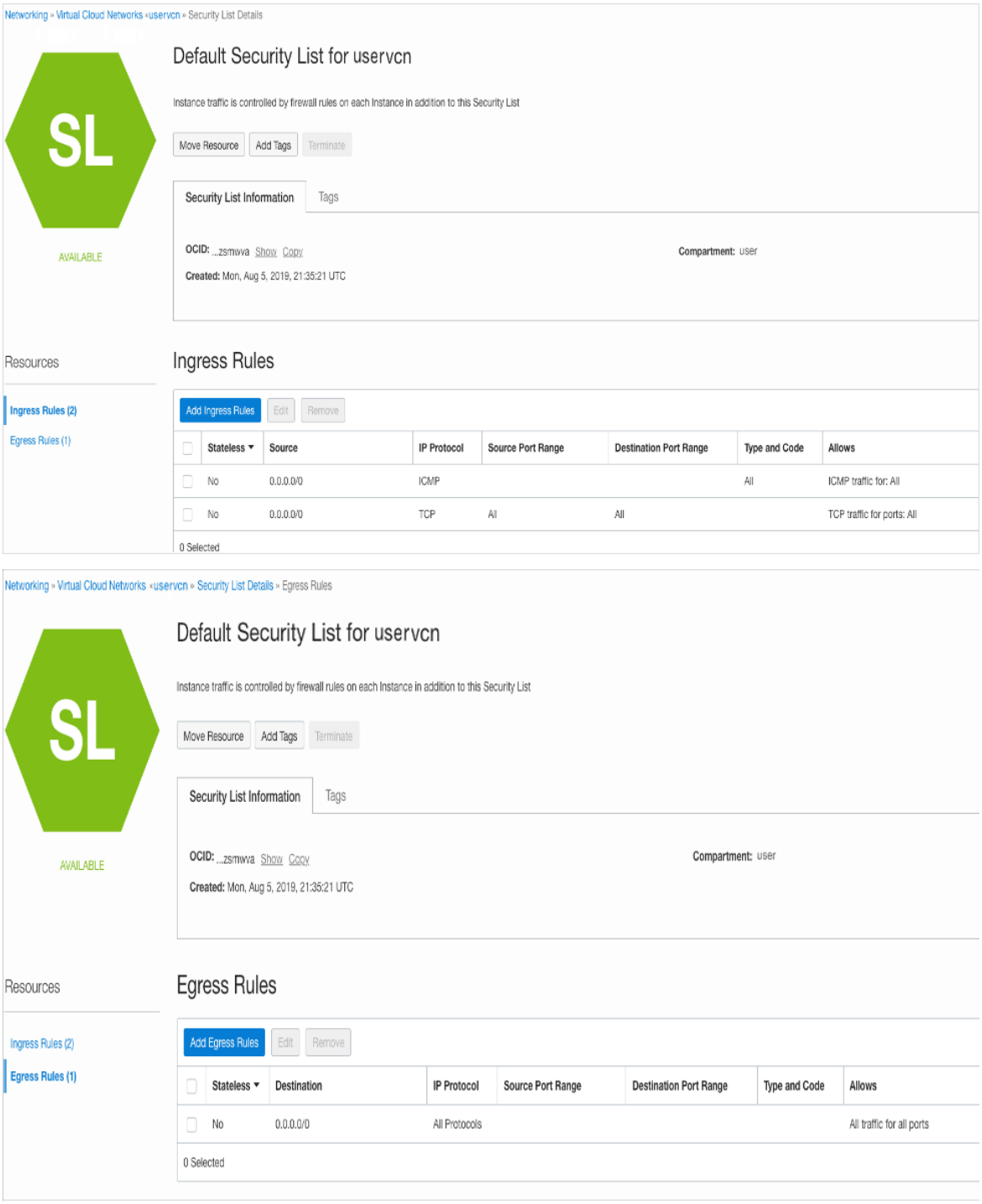
6. Create Internet Gateway. To create internet gateway click **Internet Gateways**, set an internet gateway for the vSRX to be reachable from outside.

Figure 146: Internet Gateway



7. Security list information to enable the SSH option. Select the default security list and the Ingress Rules like ICMP rule to allow ping from traffic by setting source CIDR of any any.

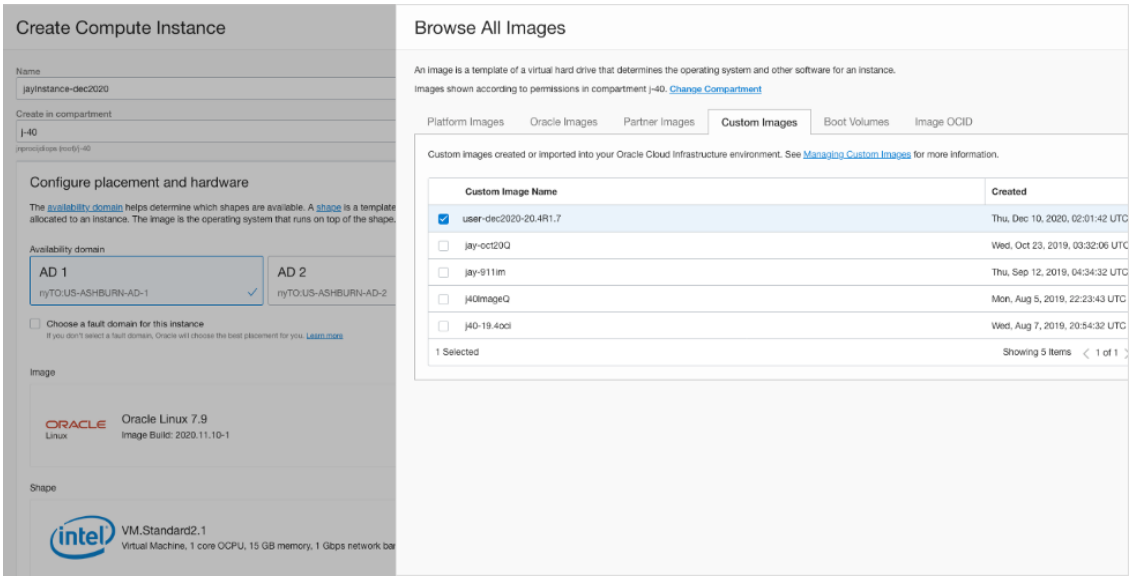
Figure 147: Security List Information

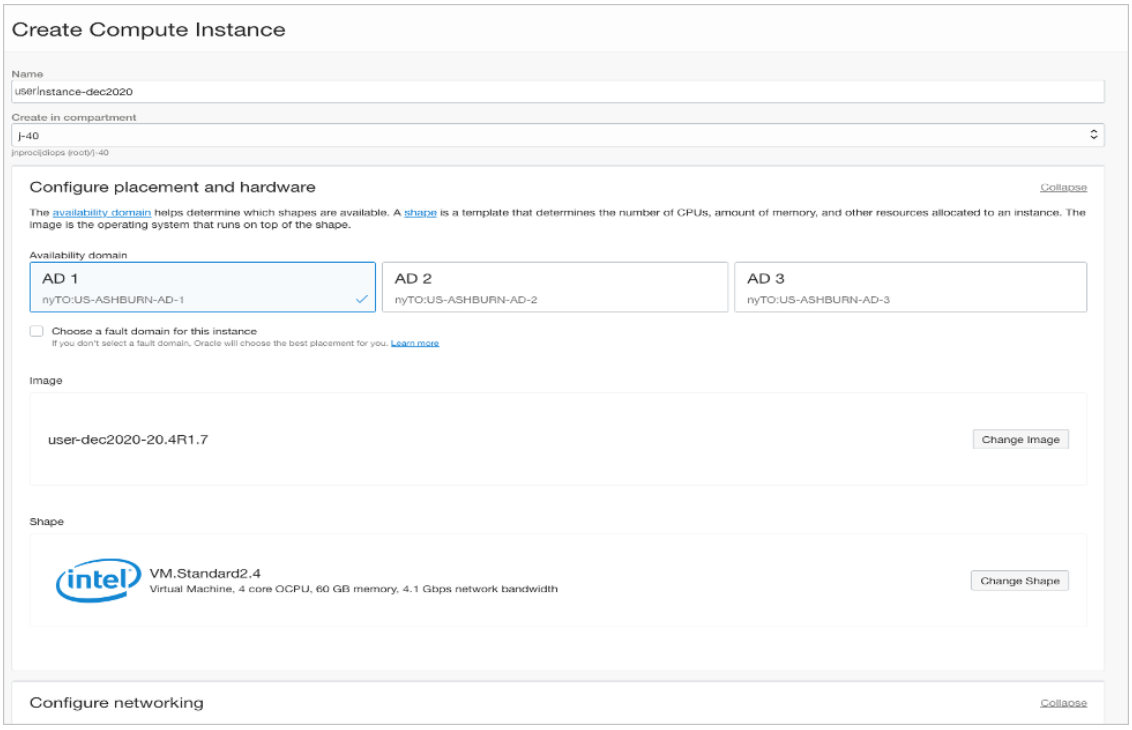


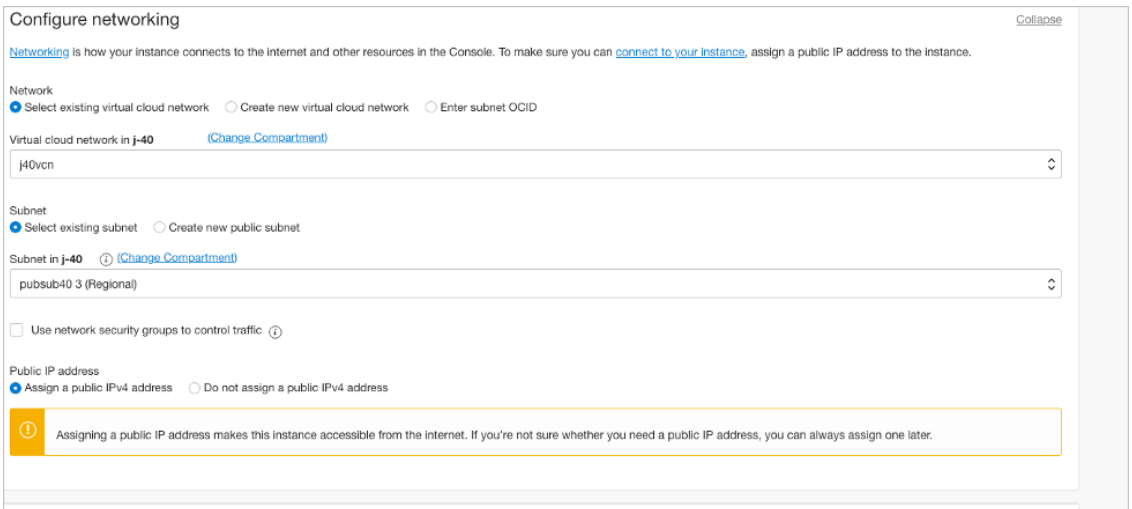
8. Create your vSRX instance in the VNC created.

   a. Open the navigation menu. Under **Core Infrastructure**, select **Compute** and click **Instances**, and then click on **Create Instance**.

b. Figure 148: Create Compute Instance







c. On the **Create Instance** page, enter the name of your instance.

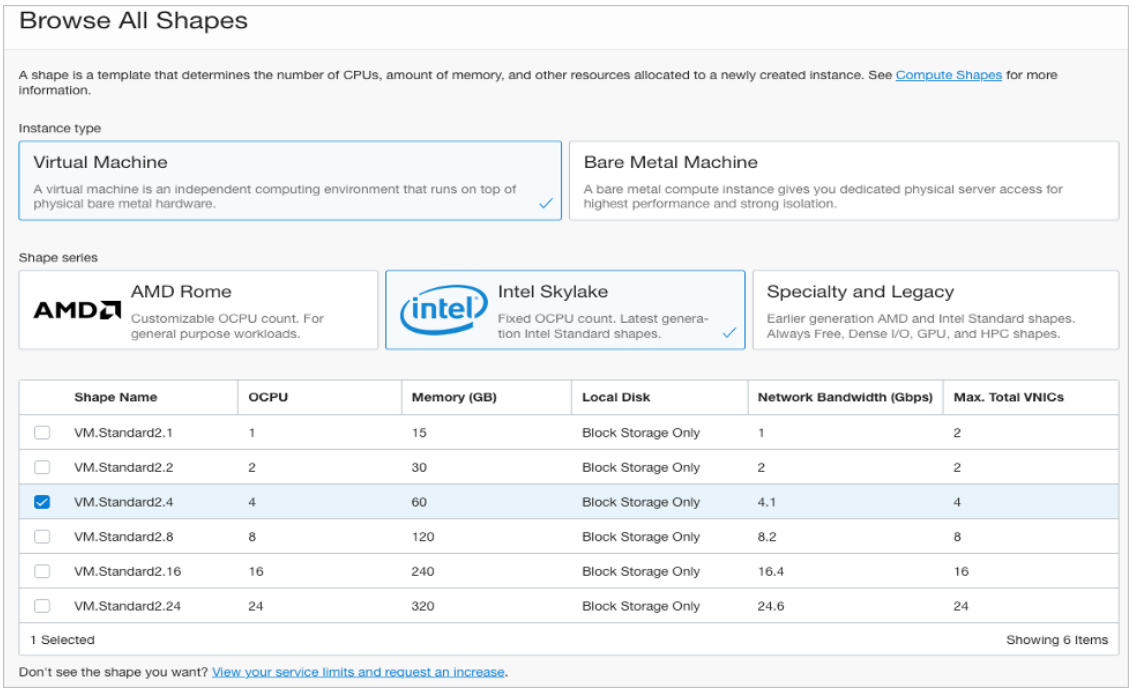
d. Choose an operating system or image source: Click **Change Image** and then click **Image Source** to select the image that you want to use. Select **Custom Images** and choose the image from the compartment. OCI vSRX image you want and then click **Select Image**.

.

Instance type – Virtual Machine.

e. Choose Instance Shape: Click **Change Shape** to select the standard predefined OCI shape. Select the VM standard 2.4 which has 4 NICs and 4 OCPUs and click **Selcect Shape**.

> **NOTE**: vSRX needs a minimum of 2 vCPUs to launch.



f. Under **Networking** tab select the virtual cloud network compartment, virtual cloud network, subnet compartments, subnet.

g. To create a public IP address for the instance, select the **Assign a public IPv4 address** option.

> **NOTE**: Accept default options for Availability Domain, Instance Type, and Instance Shape.

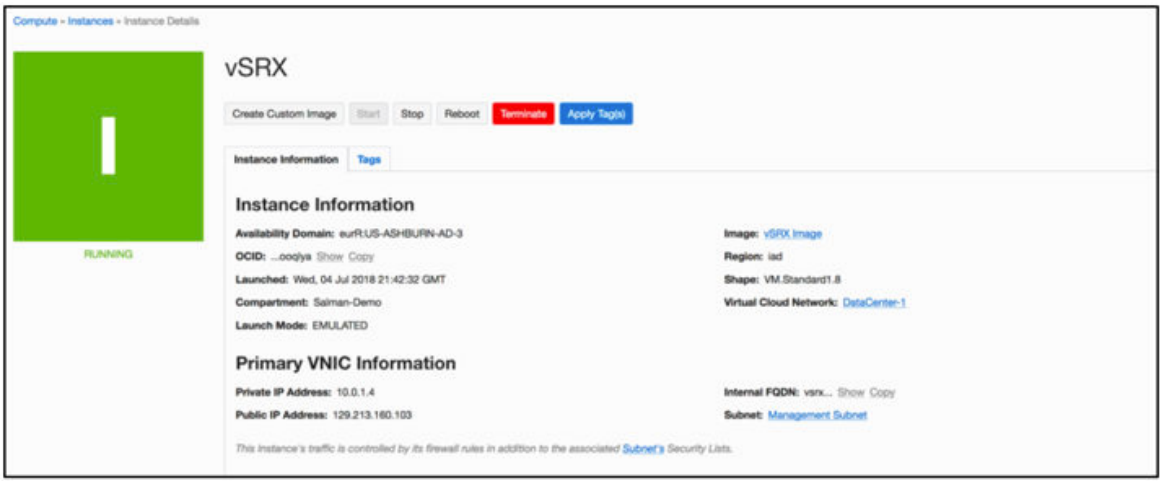
h. Add SSH keys: Under **Add SSH keys** tab, you can paste a public key by selecting the **Paste public keys** option and paste the public SSH key that was generated or you can create a new SSH key to access the vSRX and then click **Create**.

After a few minutes, we can ssh the instance using the public IP allocated for the instance (this would be displayed on the instance). Reboot the instance after adding interfaces.

The instance is displayed in the Console in a provisioning state. Expect provisioning to take several minutes before the status updates to Running. Do not refresh the page. After the instance is running, allow another few minutes for the operating system to boot before you attempt to connect. When you are ready to connect to the instance, make a note of both the public IP address and the initial password.

After the instance is provisioned, details about it appear in the instance list as shown below.

Figure 149: vSRX Instance Launched in OCI



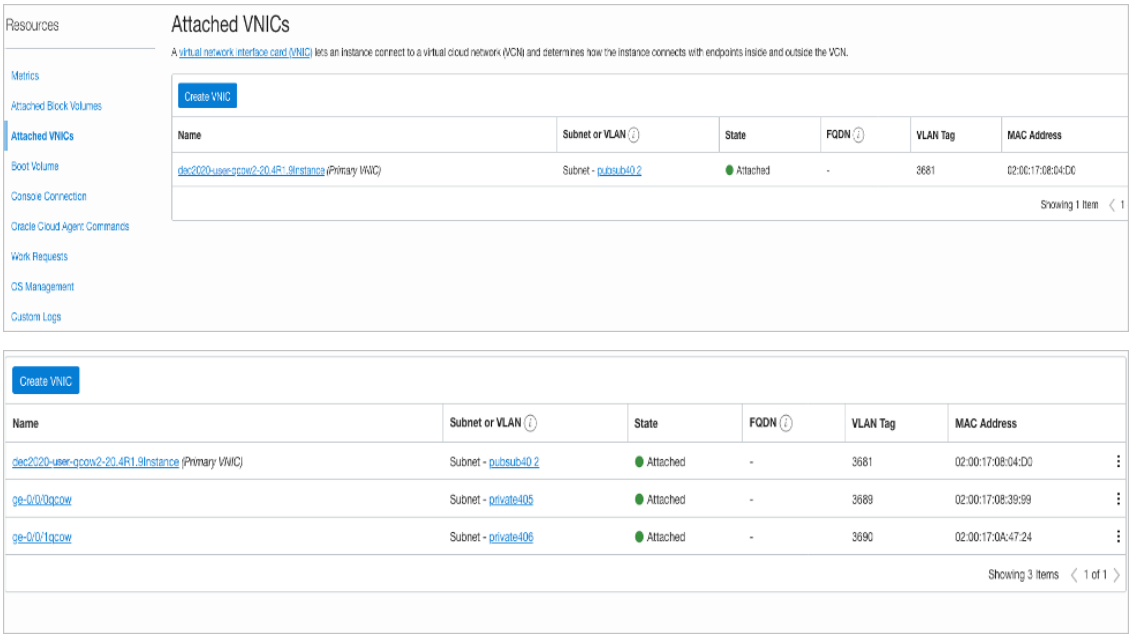
9. Adding interfaces for traffic.

Network interfaces need to be added after the instance has been created.

a. Click **Attached VNICs** and select **Create VNIC** (ge000 -public and ge001-private). Select the subnet that was created and click **Save Changes** to add VNICs to the instance.

NOTE: Order of attaching network interfaces is important. You must map the first network interface to fxp0, then the second interface to ge-0/0/0, then to ge-0/0/1 and so on.

Figure 150: Attached VNICs



10. Connect to the launched vSRX instance. Open your SSH client to access the launched vSRX instance. At first boot you can only SSH the vSRX. vSRX boots up with the default OCI configuration. Use your private key to SSH the vSRX instance.

Upgrade the Junos OS for vSRX Software Release

You can upgrade the Junos OS for vSRX software using the CLI. Upgrading or downgrading Junos OS can take several hours, depending on the size and configuration of the network. Download the desired Junos OS Release for the vSRX 3.0 upgrade tgz file from the Juniper Networks website. Example filename is **junos-install-vsrx3-x86-64-xxxxx.tgz**.

You also can upgrade using J-Web (see J-Web) or the Junos Space Network Management Platform (see Junos Space).

For the procedure on upgrading a specific Junos OS for vSRX software release, see the *Migration, Upgrade, and Downgrade Instructions* topic in the release-specific *vSRX Release Notes* available on the vSRX TechLibrary webpage.

CHAPTER 33

vSRX Licensing

IN THIS CHAPTER

Licenses for vSRX

- OCI supports Bring Your Own License (BYOL) licensing model. The BYOL license model allows you to customize your license, subscription and support to fit your needs. You can purchase BYOL from Juniper Networks or Juniper Networks authorized reseller.
- You need a license to use the software features on the vSRX. To find out the features supported on vSRX, see:
  - Supported Features on vSRX.
  - Juniper Agile Licensing Guide.
  - Flex Software License.
- To add, delete, and manage licenses, see Managing Licenses.