Juniper Networks | Engineering Simplicity

vSRX Deployment Guide for VMware

Published
2020-12-28

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

vSRX Deployment Guide for VMware

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

Use this guide to install the vSRX Virtual Firewall on VMware. This guide also includes basic vSRX configuration and management procedures.

After completing the installation and basic configuration procedures covered in this guide, refer to the Junos OS documentation for information about further software configuration.

1
CHAPTER

Overview

Understand vSRX with VMware

IN THIS SECTION

This section presents an overview of vSRX on VMware

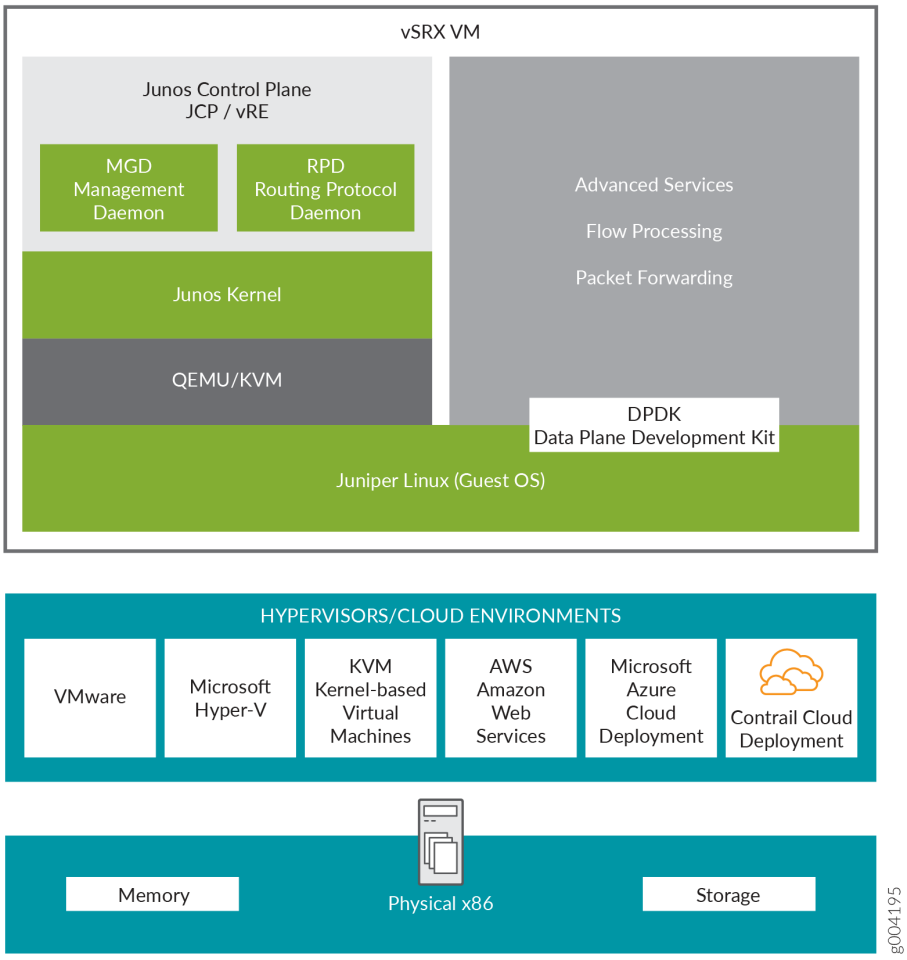
vSRX Overview

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 1 on page 3 shows the high-level architecture.

Figure 1: 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.

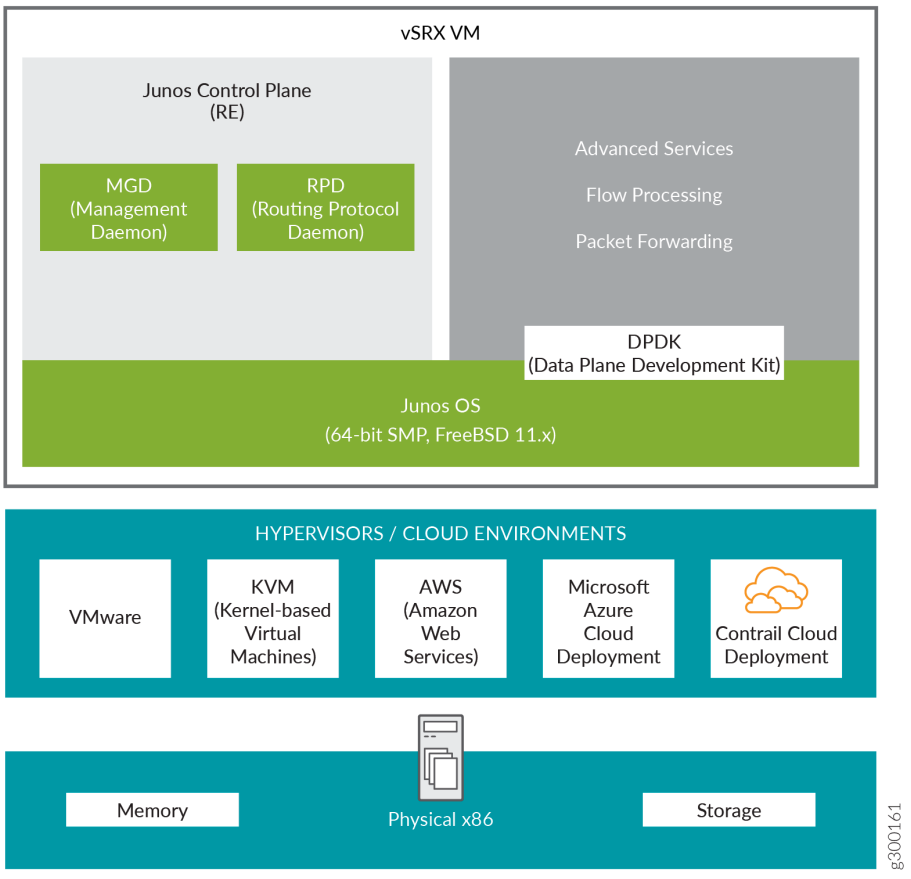
In vSRX 3.0 architecture, FreeBSD 11.x is used as the guest OS and the Routing Engine and Packet Forwarding Engine runs on FreeBSD 11.x as single virtual machine for improved performance and scalability. vSRX 3.0 uses DPDK to process the data packets in the data plane. A direct Junos upgrade from vSRX 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 2 on page 4 shows the high-level software architecture for vSRX 3.0

Figure 2: 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

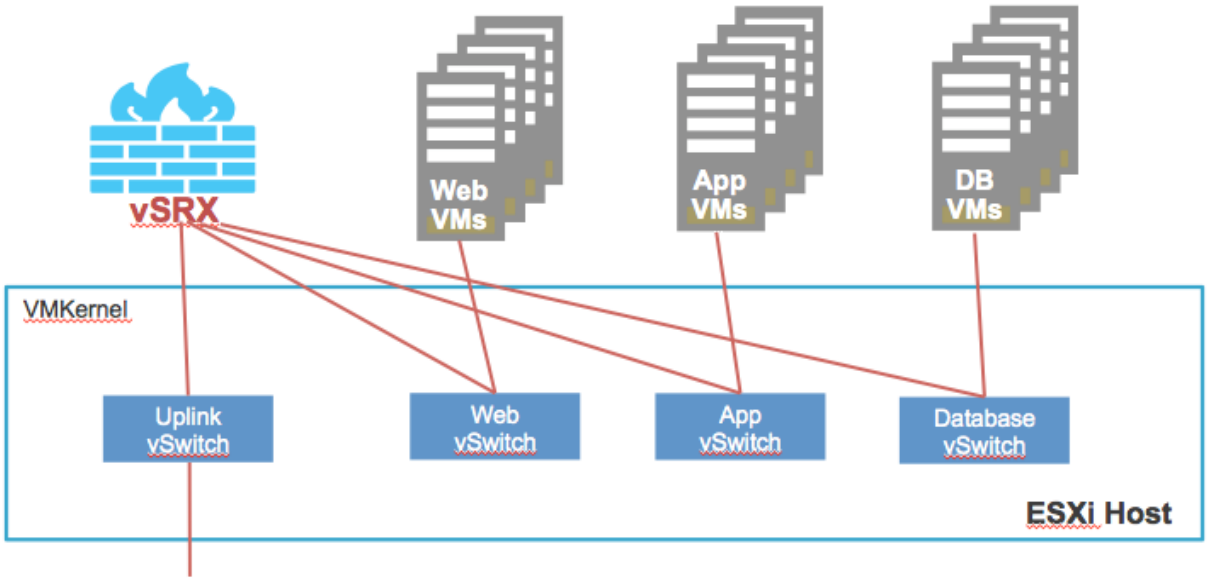
VMware vSphere is a virtualization environment for systems supporting the x86 architecture. VMware ESXi® is the hypervisor used to create and run virtual machines (VMs) and virtual appliances on a host machine. The VMware vCenter Server® is a service that manages the resources of multiple ESXi hosts.

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

Figure 3 on page 6 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 3: Example of vSRX Deployment



vSRX Scale Up Performance

Table 1 on page 6 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 1: 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

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

vCPUs	vRAM	NICs	Junos OS Release Introduced
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

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 2 on page 8 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
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

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

vCPUs	Memory	Flow Session Capacity
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.
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 3 on page 10 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 3: 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 5.5, 6.0, 6.5	Junos OS Release 17.4R1, 18.1R1, 18.2R1, 18.3R1
	VMware ESXi 6.5	Junos OS Release 18.4R1

Table 3: Specifications for vSRX and vSRX 3.0 on VMware *(Continued)*

Component	Specification	Junos OS Release Introduced
	VMware ESXi 6.5 and 6.7 (For vSRX 3.0 only)	Junos OS Release 19.3R1
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
	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

Table 3: Specifications for vSRX and vSRX 3.0 on VMware *(Continued)*

Component	Specification	Junos OS Release Introduced
vNICs	Up to 10 vNICs • 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*. • 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 3: 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 4 on page 13 lists the specifications on the vSRX 3.0 virtual machine (VM).

Table 4: 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 4: 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

Hardware Specifications

Table 5 on page 14 lists the hardware specifications for the host machine that runs the vSRX VM.

Table 5: 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 5: Hardware Specifications for the Host Machine *(Continued)*

Component	Specification
Physical NIC support on vSRX 3.0	Support SR-IOV on X710/XL710 vSRX3.0 SR-IOV HA on I40E (X710,X740,X722 and so on) are not supported on VMware. Mellanox NIC (any ConnectX) cards are not support on VMWare. Chassis Cluster is not supported with SRIOV 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 6 on page 16 and Table 7 on page 17.

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 6 on page 16 shows the interface names and mappings for a standalone vSRX VM.

Table 6: Interface Names for a Standalone vSRX VM

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

Table 6: Interface Names for a Standalone vSRX VM *(Continued)*

Network Adapter	Interface Name in Junos OS
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 7 on page 17 shows the interface names and mappings for a pair of vSRX VMs in a cluster (node 0 and node 1).

Table 7: 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)

Table 7: Interface Names for a vSRX Cluster Pair *(Continued)*

Network Adapter	Interface Name in Junos OS
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 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 8 on page 19 lists the factory default settings for the vSRX security policies.

Table 8: 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

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 9 on page 20.

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 9: 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 9: 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. 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

Table 9: vSRX Feature Considerations *(Continued)*

Feature	Description	
	IKE SA (Site-to-site + AutoVPN)	10,000
	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 9: 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

Table 9: vSRX Feature Considerations *(Continued)*

Feature	Description
	• Multicast • 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 9: 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 10 on page 26 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 10: 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	

Table 10: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
High-priority queue on SPC	Not supported
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

Table 10: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
Ethernet link fault management	
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

Table 10: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
IEEE 802.1X port-based authentication control with multisupplicant support	Not supported
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

Table 10: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
IPv6 support	
DS-Lite concentrator (also called Address Family Transition Router [AFTR])	Not supported
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	

Table 10: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
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
Logical systems	Not supported
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	

Table 10: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
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
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	

Table 10: SRX Series Features Not Supported on vSRX *(Continued)*

SRX Series Feature	vSRX Notes
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
SRC application	Not supported
Junos Space Virtual Director	Only supported with VMware

2

CHAPTER

Installing vSRX in VMware

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" on page 45.

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 11 on page 36 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.

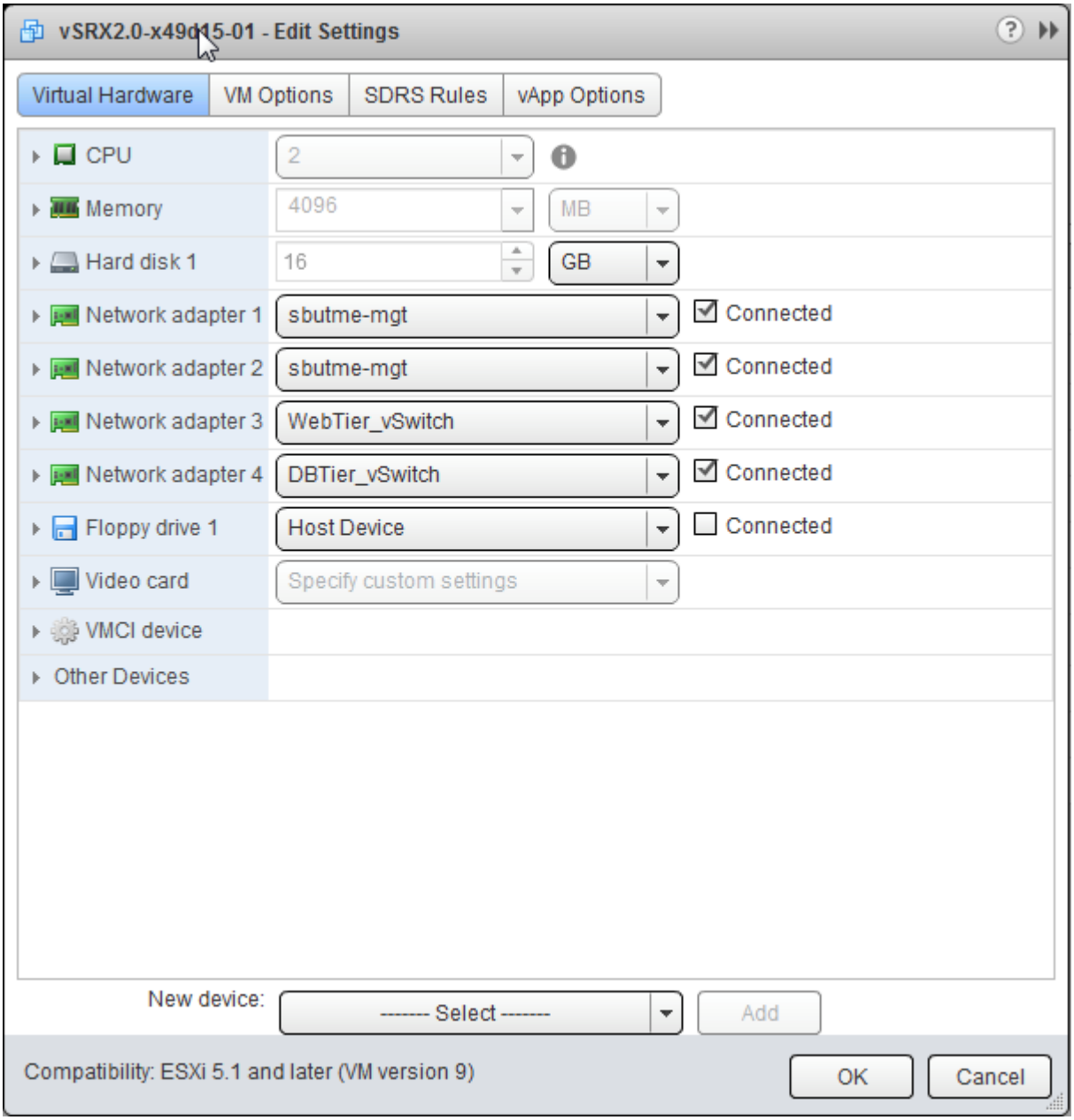
Table 11: Disk Formats for Virtual Disk Storage

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" on page 49.

In Figure 4 on page 37, network adapter 2 uses the management network for the uplink to the external network.

Figure 4: 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.u3ESWGB1aSQeXrSeg6zoCWZw0D6M6vnmWb8DAWsprNXyKZeW6M3kErFFTFtAuNpG
jDjfwX4t."; ## 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.

3

CHAPTER

vSRX VM Management

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" on page 10.

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" on page 10 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.

4

CHAPTER

Configuring and Managing vSRX

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

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

Configuring vSRX Using the J-Web Interface

IN THIS SECTION

Accessing 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:

- Configure an IP address on fxp0.
- Configure a default route if the fxp0 IP address is on a different subnet than the host server.
- Enable Web management through the fxp0 interface.

```
system {
    services {
        web-management {
            http {
                interface fxp0.0;
```

```
                    }
                }
            }
        }
```

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 12 on page 60.

       - Instance name and user account information

         Table 12: 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 12: 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 13 on page 61 lists the system time options.

Table 13: 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.

Table 13: System Time Options *(Continued)*

Field	Description
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.

Applying 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.

Adding 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.

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

Software Receive Side Scaling

IN THIS SECTION

Overview

Contemporary NICs support multiple receive and transmit descriptor queues (multi-queue). On reception, a NIC can send different packets to different queues to distribute processing among CPUs. The NIC distributes packets by applying a filter to each packet that assigns it to one of a small number of logical flows. Packets for each flow are steered to a separate receive queue, which in turn can be processed by separate CPUs. This mechanism is generally known as Receive-side Scaling (RSS). The goal of RSS technique is to increase performance uniformly. RSS is enabled when latency is a concern or whenever receive interrupt processing forms a bottleneck. Spreading load between CPUs decreases queue length. For low latency networking, the optimal setting is to allocate as many queues as there are CPUs in the system (or the NIC maximum, if lower). The most efficient high-rate configuration is likely

the one with the smallest number of receive queues where no receive queue overflows due to a saturated CPU. You can improve bridging throughput with Receive Side Scaling.

As per flow thread affinity architecture each flow thread (FLT) polls for packet from dedicated receiving queue of NIC and process the packets until run to completion. Therefore, flow threads are bound to NIC receiving (RX) and transmitting (TX) queues for packet processing to avoid any disagreement. Hence, NIC must have same number of RX and TX queues as number of vSRX data plane CPU to support multi core vSRX flavors. Software RSS (SWRSS) removes this limitation of NIC HW queues to run vSRX multi-core flavors by implementing software-based packet spraying across various FLT thread.

Software RSS offloads the handling of individual flows to one of the multiple kernel, so the flow thread that takes the packets from the NIC can process more packets. Similar to RSS, network throughput improvement when using SWRSS has a linear correlation with CPU utilization.

In SWRSS, each NIC port is initialized with equal or lesser number of hardware RX/TX queues as that of I/O threads. I/O threads are determined based on total data-path CPU and minimum of NIC queues among all the NIC interface in vSRX. For example, if I/O thread is computed as 4, then number of HW queue per NIC port can be less or equal to 4 queues.

If NICs do not have sufficient number of queues as FLT threads in vSRX instances supported, then Software RSS (SWRSS) is enabled by flowd data-path. SWRSS implements software model of packet distribution across FLTs after obtaining the packets from NIC receiving queues. By removing NIC HW queue limitation, SWRSS helps to scale vCPUs by supporting various vSRX instance types.

During the I/O operation the packets are fetched from receiving queues of NIC ports and packet classification is performed. Followed by distribution of packets to FLT threads virtual queues. These virtual queues are implemented over DPDK ring queue. In the transmission path, SWRSS fetches the packets from virtual transmitting queues of FLT threads and pushes these packets to NIC transmitting queues for transmit.

Number of SWRSS I/O threads are selected based on total CPU and number of NIC queues found in vSRX instances. Mix mode of operation with HWRSS and and SWRSS is not supported.

Understanding Software Receive Side Scaling Configuration

This topic provide you details on types of Software Receive Side Scaling (SWRSS) and its configuration.

SWRSS supports two modes of operation and it gets enabled based on number of data-path CPU needed. These modes are Shared IO mode and dedicated IO mode. These modes are enabled based on number of data-path CPUs needed. vSRX and vSRX3.0 supports dedicated I/O mode only.

In dedicated I/O mode flowd process creates dedicated I/O threads for I/O operation. Based on number of required I/O threads for vSRX, I/O thread is associated to a dedicated NIC port. NIC ports receiving and transmitting queue is then bonded to each I/O thread in round robin method for uniform

distribution and to avoid I/O thread locks. Each dedicated I/O thread pulls the packets in burst mode from NIC receiving queue and distributes to FLT threads and vice versa for TX path for packet transmit.

SWRSS is enabled based on the number of vCPUs. If NIC does not have sufficient number of queues as flow thread (FLT) in vSRX with different vCPUs, then Software RSS (SWRSS) is enabled by flowd process.

SWRSS is not enabled in the following scenarios:

- When the NIC has sufficient number of hardware RX or TX queues for required PFE data-path CPU.
- When the vSRX (based on number of vCPUs) and NIC result the smaller number of FLT CPUs as that obtained in nearest hardware RSS (HWRSS) mode. In such scenario, vSRX will be enabled with HWRSS mode which results more FLT CPU than SWRSS mode, providing better packet processing throughput.
- SWRSS is not recommended for vSRX with certain type of NIC that supports lesser number of NIC queues than needed to run dedicated IO thread. In such cases, SWRSS is enabled but extra CPUs are attached to FLT CPU, until I/O CPUs are completely utilized.

If SWRSS is not enabled use the **set security forwarding-options receive-side-scaling software-rss mode enable** command to enable SWRSS. When you run this command SWRSS will be enabled by force regardless of the NIC RSS or the number of vCPUs. If you do not enable SWRSS using the CLI then enabling of SWRSS automatically is decided based on the default ratio of FLT: IO (4:1).

To configure the number of required IO threads, use the **set security forwarding-options receive-side-scaling software-rss io-thread-number *<1-8>*** command. To view the actual number of vCPUs assigned to IO flow threads use the **show security forwarding-options resource-manager** command.

You can decide enabling of SWRSS automatically or by force based on the architecture and conception of IO thread and worker thread. Enabling SWRSS impacts the performance, so we recommend that the number of IO thread should be changed only if required and until the performance impact bottleneck point is reached.

GTP Traffic with TEID Distribution and SWRSS

IN THIS SECTION

Overview GTP Traffic Distribution with TEID Distribution and SWRSS

IN THIS SECTION

The topic provides an overview of asymmetric fat tunnel solution for GTP traffic with TEID distribution and SWRSS.

With TEID-based hash distributions feature, the GTP packets would be distributed to the flow thread according to the hash value calculated by TEID. The algorithm of hash calculation is same as GTP distribution in flow module, which ensures the GTP packets would not be reinjected again in the flow process.

There is a 4-byte field inside GTP payload called tunnel endpoint identifier (TEID), which is used to identify different connections in the same GTP tunnel.

A fat GTP tunnel carries data from different users. IPsec tunnels on the security gateway could be a fat tunnel due to the fat GTP tunnel. vSRX can create one GTP session with a high-bandwidth of GTP traffic. However, the throughput is limited to one core processor's performance.

If you use TEID-based hash distribution for creating GTP-U sessions, then you can:

- Enable vSRX and vSRX 3.0 instances to process asymmetric fat tunnels for parallel encryption on multiple cores for one tunnel.
- You can split a fat GTP session to multiple sessions and distribute them to different cores. This helps to increase the bandwidth for fat GTP tunnel.

The TEID based hash distribution creates GTP-U sessions to multiple cores. The clear text traffic acts as a fat GTP tunnel. This helps a fat GTP session to split into multiple slim GTP sessions and handle them on multiple cores simultaneously.

GTP Traffic Performance with TEID Distribution and SWRSS

vSRX instances support Software Receive Side Scaling (SWRSS) feature. SWRSS is a technique in the networking stack to increase parallelism and improve performance for multi-processor systems. If NICs do not have sufficient number of queues as flow thread (FLT), based on vSRX type, then Software RSS (SWRSS) is enabled by flowd process.

With Software Receive Side Scaling (SWRSS) support on vSRX and vSRX 3.0, you can assign more vCPUs to the vSRX regardless of the limitation of RSS queue of underlying interfaces.

Based on SWRSS you can improve the GTP traffic performance using Tunnel endpoint identifier (TEID) distribution and asymmetric fat tunnel solution by:

- Assigning specific number of vCPUs for input output flow usage—With SWRSS enabled, you can assign more vCPUs for input/output (IO) threads when the IO threads are less. Or you can assign less vCPUs for IO threads if the flow process is consuming more vCPU. Use the **set security forwarding-options receive-side-scaling software-rss io-thread-number <io-thread-number>**.

- Distributing the packets to flow threads according to the TEID inside the packet, which would avoid reinjecting the packets in flow process—This feature is enabled when both SWRSS is enabled and when you configure the **set security forwarding-process application-services enable-gtpu-distribution** command.

  With this feature, the GTP packets would be distributed to the flow thread according to the hash value calculated by TEID. The algorithm of hash calculation is same as GTP distribution in flow module, which ensures the GTP packets would not be reinjected again in flow process.

- Utilizing fragment matching and forwarding mechanism in input/output thread when GTPU distribution is enabled—This mechanism ensures that all the fragments of the same packet would be distributed to one flow thread according to the TEID.

  SWRSS uses IP pair hash to distribute packets to flow threads. For GTP traffic with GTPU distribution enabled, TEID distribution is used to distribute packets to the flow threads. For fragmented packets, TEID cannot be retrieved from non-first fragments. This will require fragment matching and forwarding logic to ensure all fragments are forwarded to the flow thread based on TEID.

Enabling GTP-U TEID Distribution with SWRSS for Asymmetric Fat Tunnels

The following configuration helps you enable PMI and GTP-U traffic distribution with SWRSS enabled.

Before you begin, understand:

- SWRSS concepts and configurations.
- How to establish PMI and GTP-U

With Software Recieve Side Scaling (SWRSS) enabled, you can assign more vCPUs for input/output (IO) threads when the IO threads are less. Or you can assign less vCPUs for IO threads if the flow process is consuming more vCPU. You can configure the number of IO threads required. With SWRSS is enabled and IO threads configured, reboot the vSRX for configuration to take effect. After IO threads are configured, distribute the GTP traffic to the configured IO threads according to TEID-based hash

distribution for splitting a fat GTP session to multiple slim GTP sessions and process them on multiple cores in parallel.

NOTE: When PMI mode is enabled with TEID distribution and SWRSS support, performance of PMI is improved. If you want to enable PMI mode then run the **set securtiy flow power-mode-ipsec** command.

The following steps provide you details on how to enable SWRSS, configure IO threads, enable PMI mode for GTP sessions with TEID distribution for obtaining asymmetric fat tunnels:

1. SWRSS is enabled by default when NICs do not have sufficient number of queues as flow thread (FLT) based on vSRX type, then Software RSS (SWRSS) is enabled by flowd process. But, when SWRSS is not enabled use the following CLIs to enable. When you run this command SWRSS will be be enabled by force regardless of the NIC RSS or number of vCPUs.

   Enable SWRSS.

   ```
   [edit]
   user@host# set security forwarding-options receive-side-scaling software-rss mode enable
   ```

2. Configure the number of IO threads required. In this configuration we are configuring eight IO threads. The assigned number of vCPUs would be assigned for IO threads, and the rest vCPUs would be assigned for flow thread.

   ```
   [edit]
   user@host# set security forwarding-options receive-side-scaling software-rss io-thread-number 8
   ```

3.

   ```
   [edit security]
   user@host# set flow power-mode-ipsec
   ```

4. Configure GTP-U session distribution.

   ```
   [edit security]
   user@host# set forwarding-process application-services enable-gtpu-distribution
   ```

5. From the configuration mode, confirm your configuration by entering the **show** command.

```
[edit security]
user@host# show
forwarding-options {
    receive-side-scaling {
        software-rss {
             mode enable;
        io-thread-number 8;
flow {
    power-mode-ipsec;
}
forwarding-process {
    application-services {
        enable-gtpu-distribution;
    }
}
```

From the operational mode run the following command to view the actual number of vCPUs assigned to IO/flow threads.

```
show security forward-options resource-manager settings
```

```
------------------------------------------------------------------------------
Owner         Type                    Current settings      Next settings
SWRSS-IO      CPU core number         2                     2
SWRSS         SWRSS mode              Enable                Enable
```

6. Commit the configuration.

```
[edit security]
user@host# commit
```

7. Reboot the vSRX for the configuration to take effect. After rebooting the whole device, PFE would check the IO-thread value according to the NIC RSS queue and its memory.

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.

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**.

5

CHAPTER

Configuring vSRX Chassis Clusters

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
```

NOTE: Use the same cluster ID number for each node in the cluster.

NOTE: The vSRX interface naming and mapping to vNICs changes when you enable chassis clustering.

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 14 on page 78 explains the contents of the HA Cluster Settings tab.

Table 15 on page 80 explains how to edit the Node Settings tab.

Table 16 on page 81 explains how to add or edit the HA Cluster Interfaces table.

Table 17 on page 82 explains how to add or edit the HA Cluster Redundancy Groups table.

Table 14: 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 14: 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.

Table 14: Chassis Cluster Configuration Page *(Continued)*

Field	Function
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 15: 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.

Table 15: Edit Node Setting Configuration Details *(Continued)*

Field	Function	Action
Add	Adds the interface.	Click **Add**.
Delete	Deletes the interface.	Click **Delete**.

Table 16: Add HA Cluster Interface Configuration Details

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

Table 16: Add HA Cluster Interface Configuration Details *(Continued)*

Field	Function	Action
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 17: 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.

Table 17: Add Redundancy Groups Configuration Details *(Continued)*

Field	Function	Action
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**.
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.

Table 17: Add Redundancy Groups Configuration Details *(Continued)*

Field	Function	Action
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 18 on page 86.

Table 18: 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:

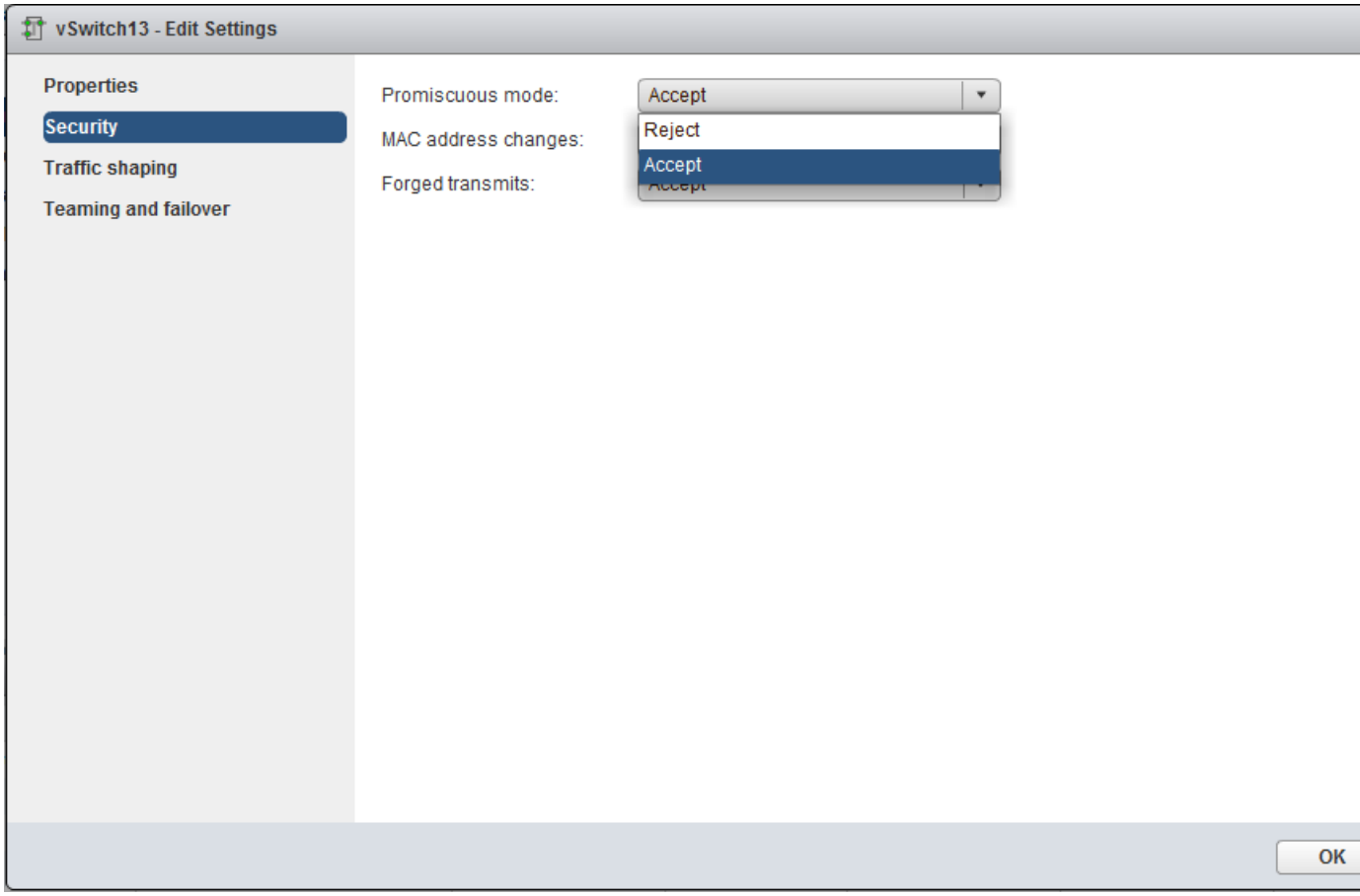
1. Choose **Configuration > Networking**.
2. Click **Add Networking** to create a vSwitch for the control link.
   Choose the following attributes:
   - Connection Type
     - Virtual Machines
   - Network Access
     - Create a vSphere switch
     - No physical adapters
   - Port Group Properties
     - 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 5 on page 87.

Figure 5: 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 6 on page 88 for vSwitch properties and Figure 7 on page 89 for VM properties for the control vSwitch.

Figure 6: Control vSwitch Properties

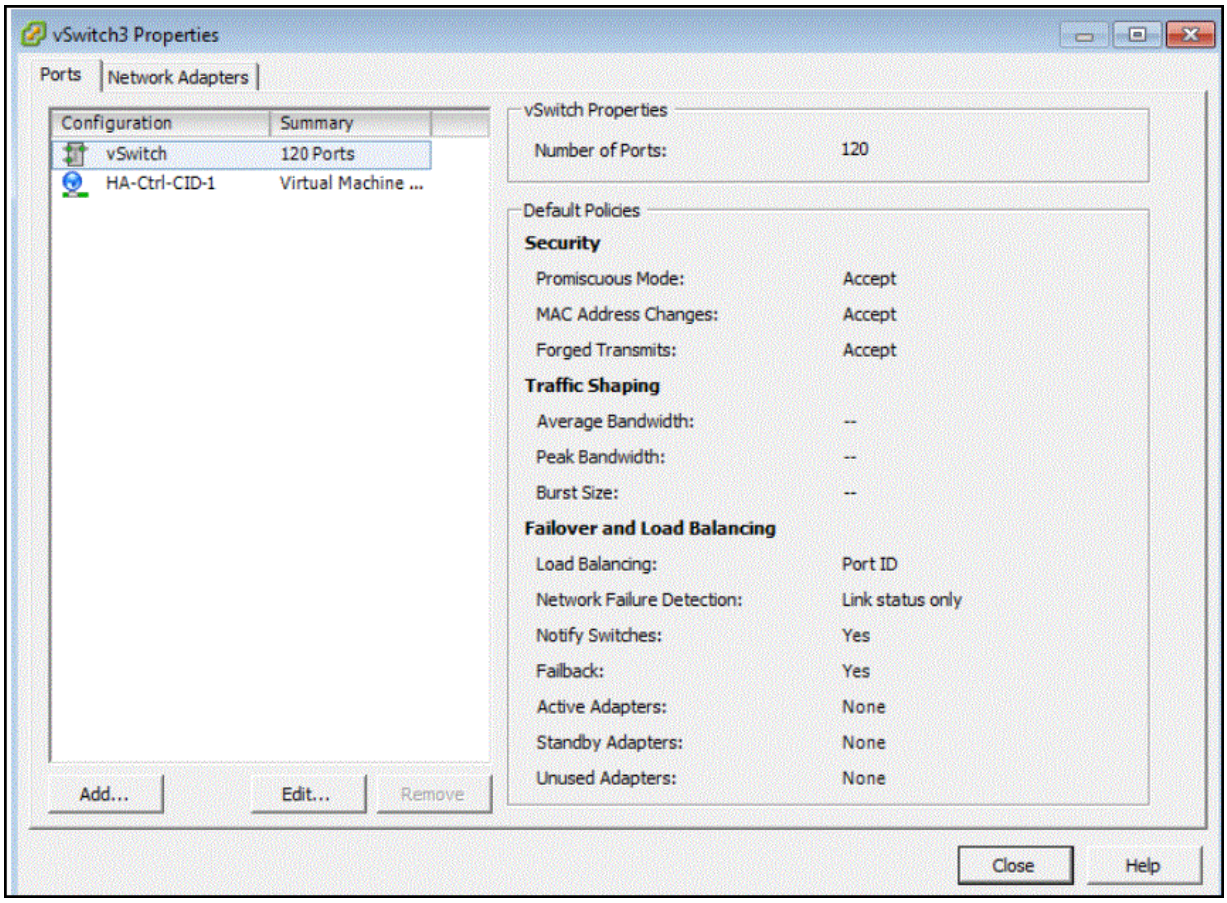
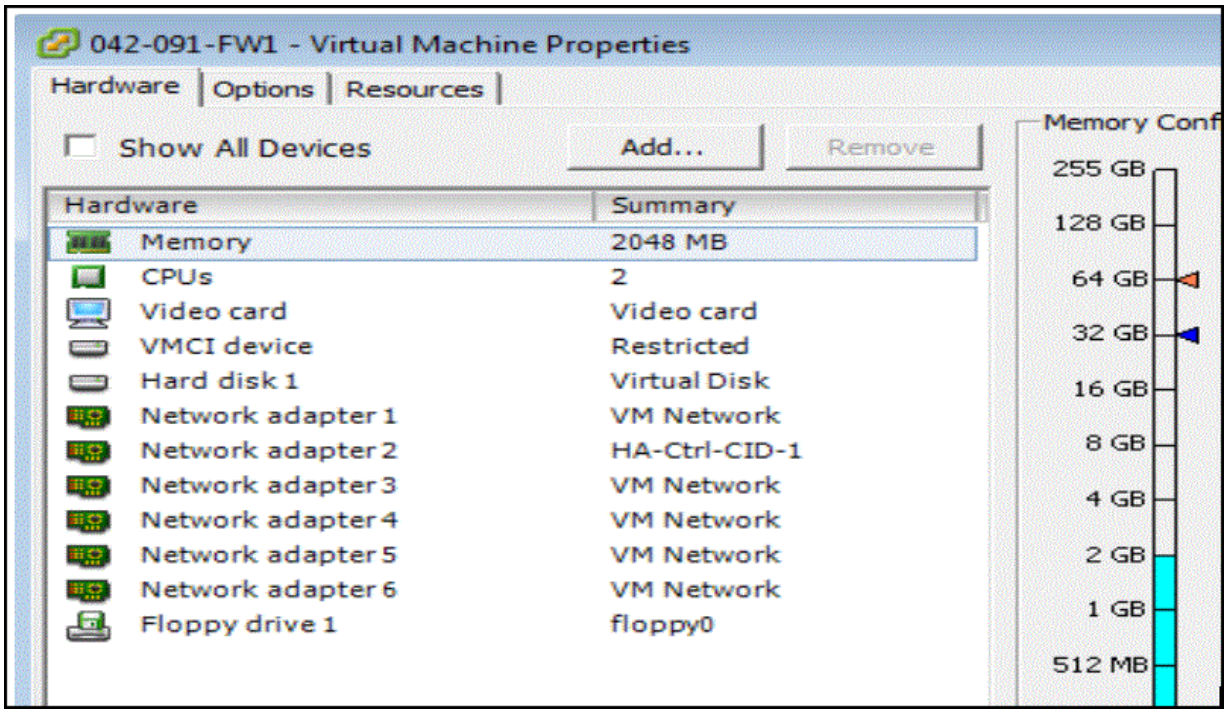
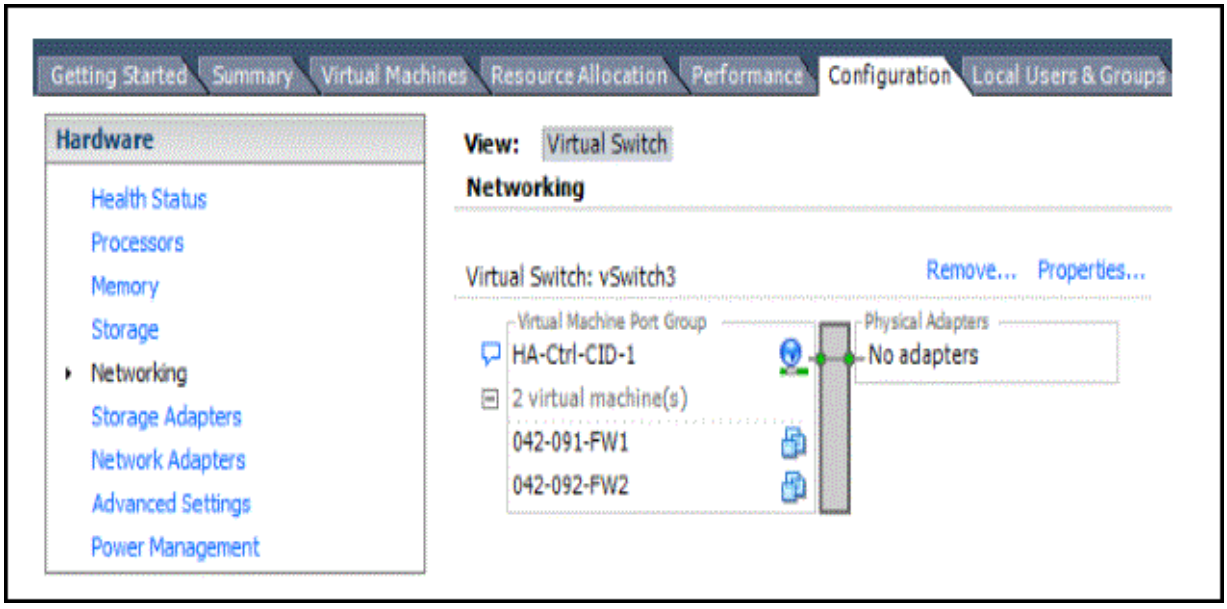


Figure 7: Virtual Machine Properties for the Control vSwitch



The control interface will be connected through the control vSwitch. See Figure 8 on page 89.

Figure 8: 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 9 on page 91 for vSwitch properties and Figure 10 on page 92 for VM properties for the fabric vSwitch.

Figure 9: Fabric vSwitch Properties

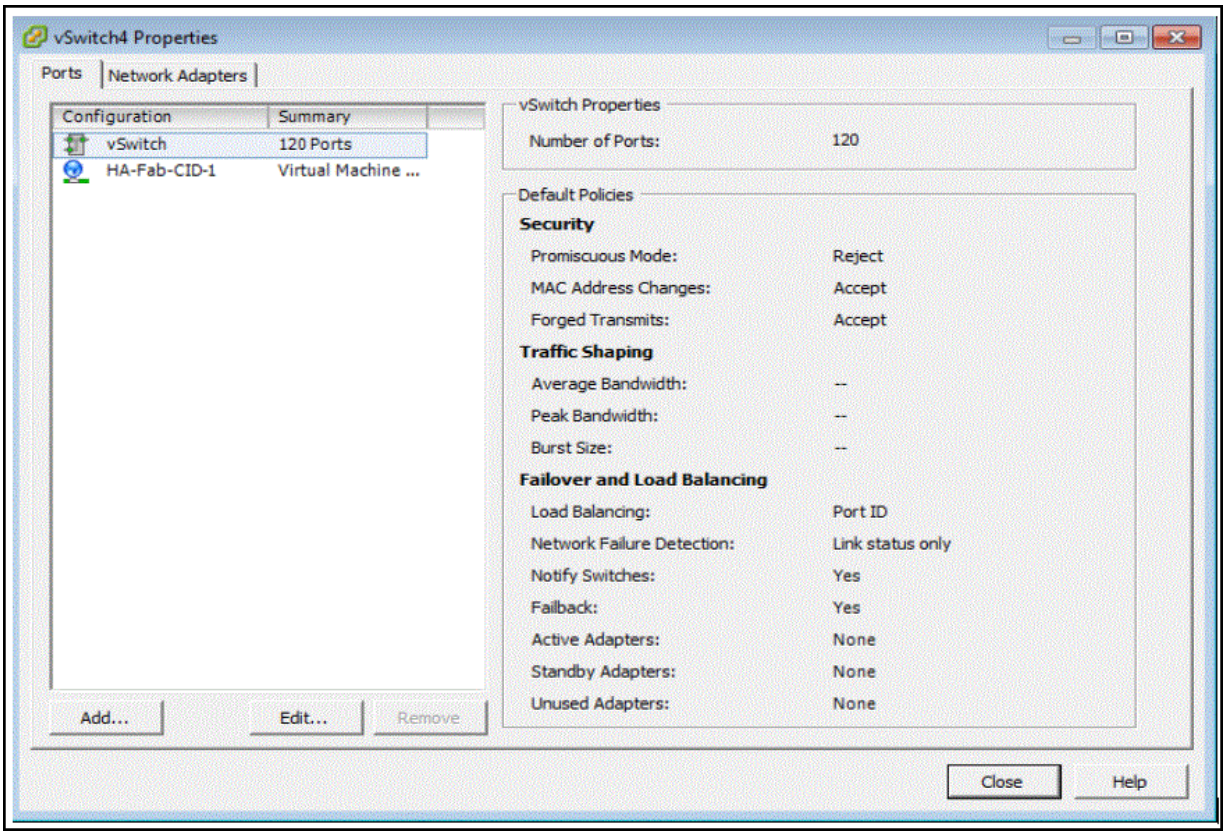
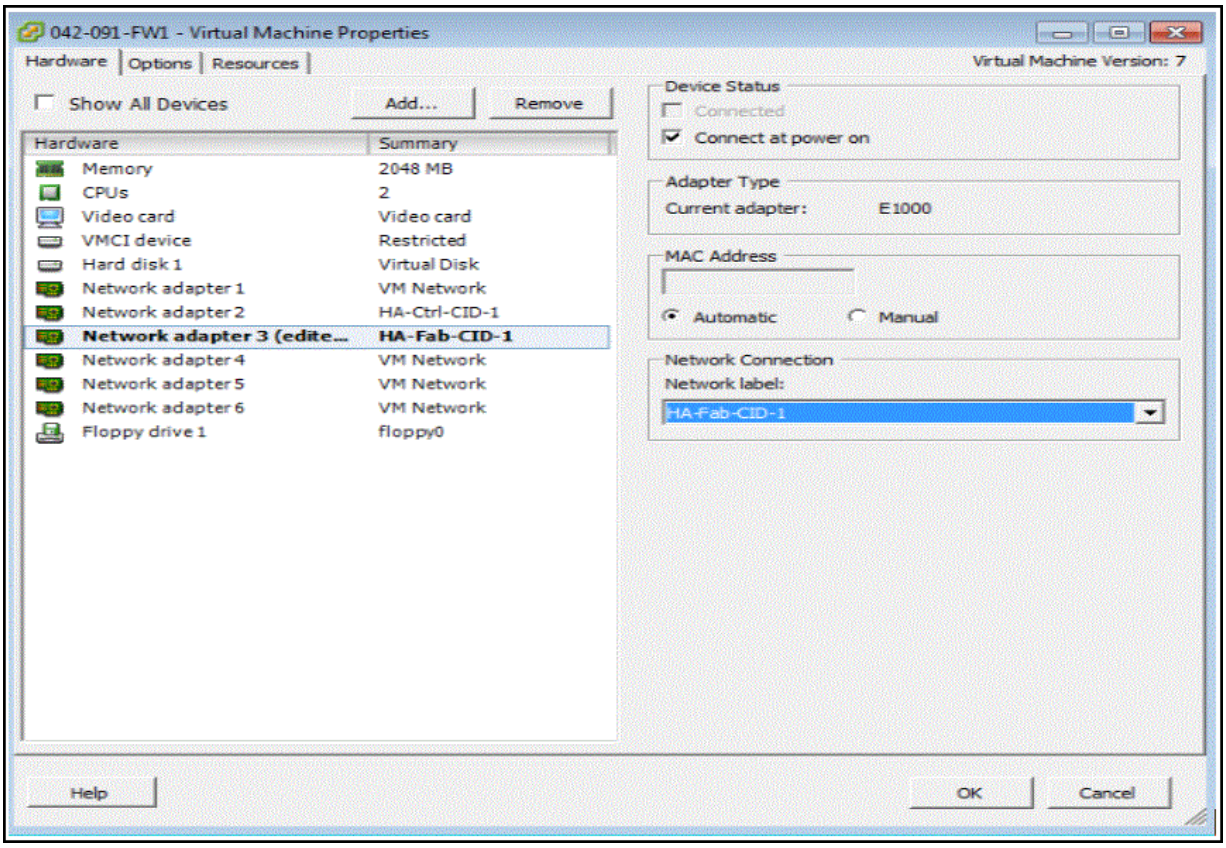
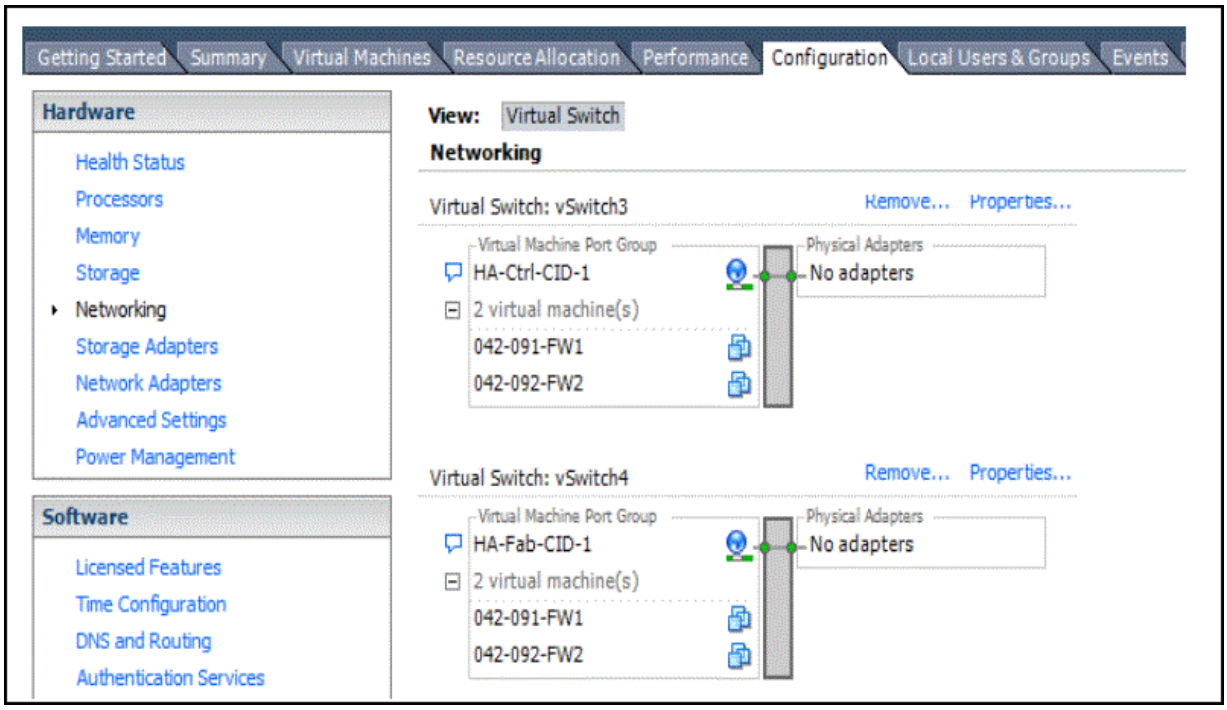


Figure 10: Virtual Machine Properties for the Fabric vSwitch



The fabric interface will be connected through the fabric vSwitch. See Figure 11 on page 93.

Figure 11: 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 12 on page 97 and Figure 13 on page 98.

Figure 12: dvPortGroup3 Settings

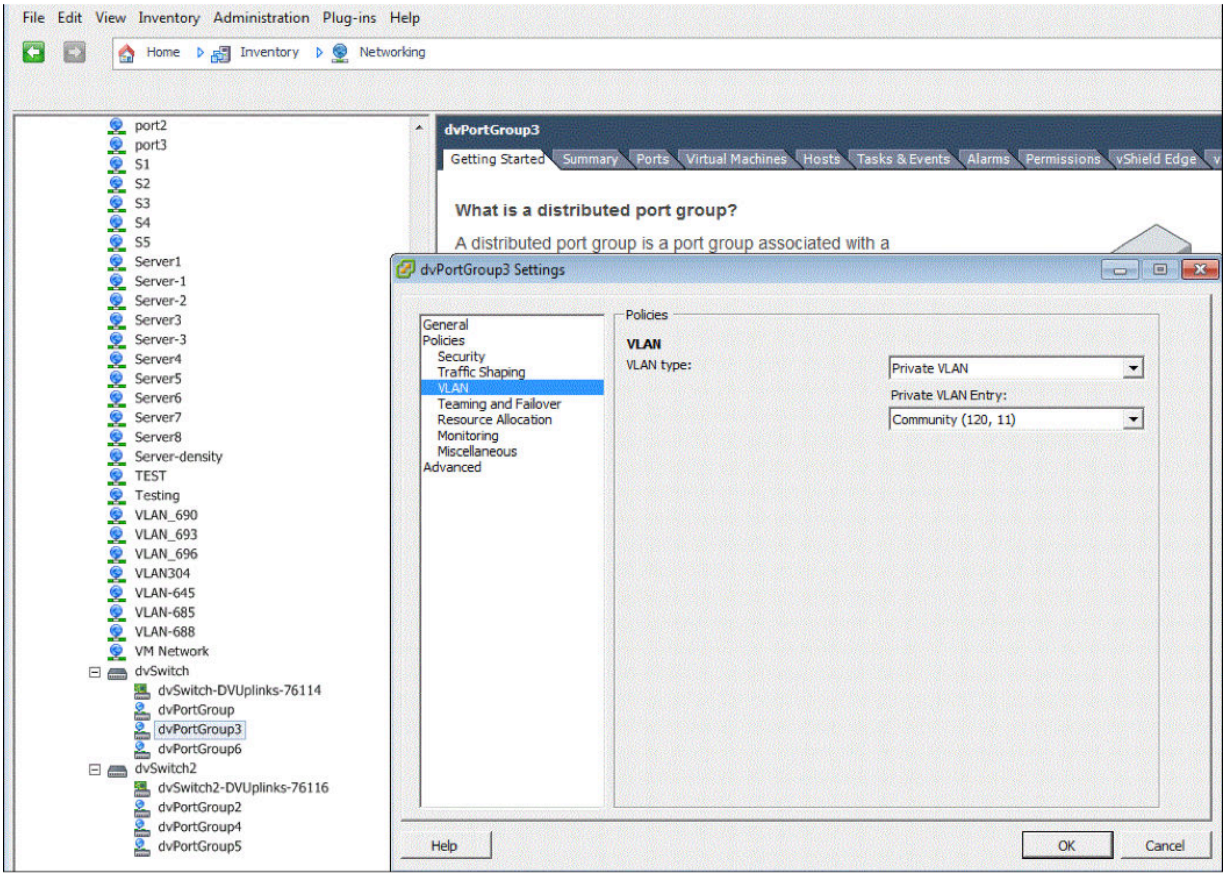
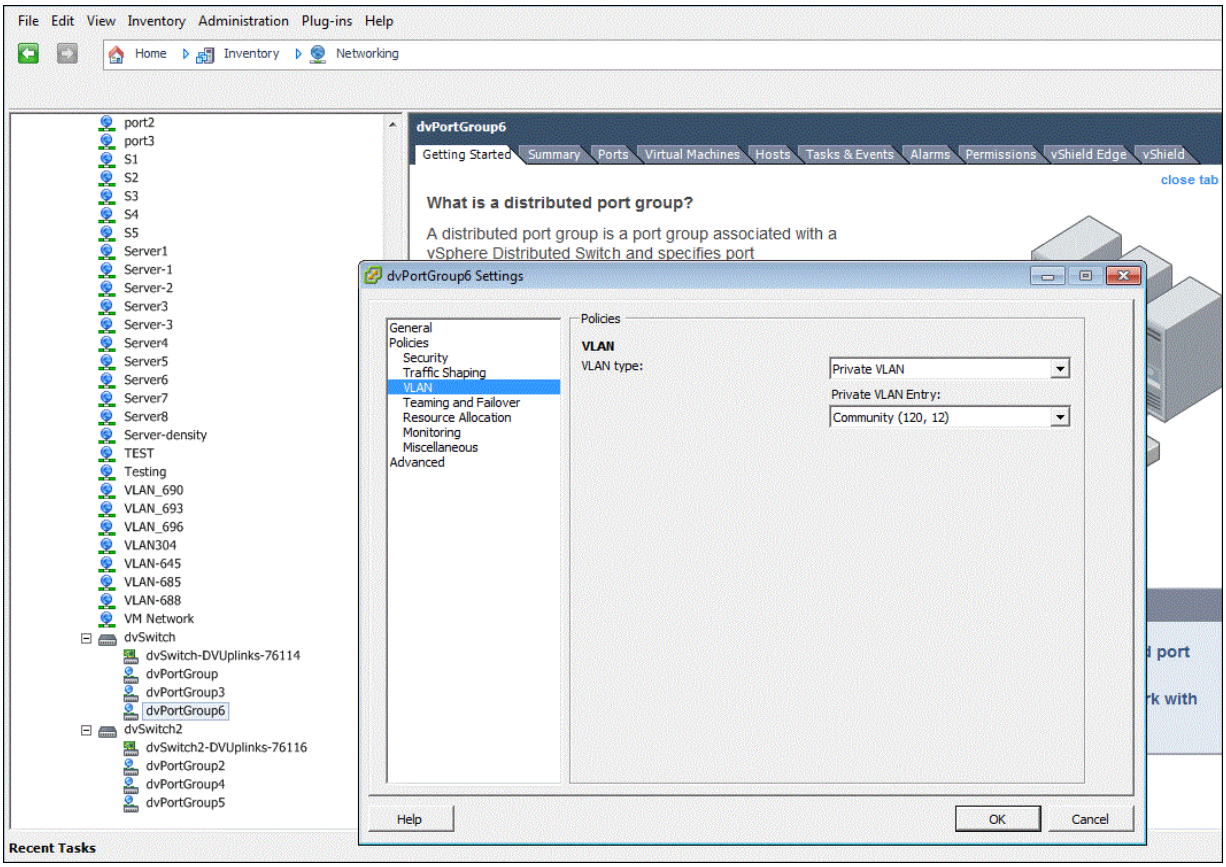


Figure 13: dvPortGroup6 Settings



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.

CHAPTER 6

Troubleshooting

Finding the Software Serial Number for vSRX

You need the software serial number to open a support case or to renew a vSRX license.

The serial number is a unique 14-digit number that Juniper Networks uses to identify your particular software installation. You can find the software serial number in the Software Serial Number Certificate attached to the e-mail that was sent when you ordered your Juniper Networks software or license. You can also use the show system license command to find the software serial number.

Use the **show system license** command to find the vSRX software serial number.

```
vsrx> show system license
```

```
License usage:
                                 Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  Virtual Appliance                     1            1           0    58 days

Licenses installed:
  License identifier: E420588955
  License version: 4
  Software Serial Number: 20150625
  Customer ID: vSRX-JuniperEval
  Features:
    Virtual Appliance - Virtual Appliance
      count-down, Original validity: 60 days

  License identifier: JUNOS657051
  License version: 4
  Software Serial Number: 9XXXXAXXXXXXX9
  Customer ID: MyCompany
  Features:
    Virtual Appliance - Virtual Appliance
      permanent
```

For more information, see Licenses for vSRX