

vSRX Deployment Guide for AWS

Published
2020-12-28

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

vSRX Deployment Guide for AWS

Copyright © 2020 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | vii

1

Overview

vSRX Overview | 2

Understand vSRX with AWS | 5

Requirements for vSRX on AWS | 12

Junos OS Features Supported on vSRX | 16

2

Installing vSRX in AWS

Configure an Amazon Virtual Private Cloud for vSRX | 31

Step 1: Create an Amazon VPC and Internet Gateway | 32

Step 2: Add Subnets for vSRX | 35

Step 3: Attach an interface to a Subnet | 36

Step 4: Add Route Tables for vSRX | 40

Step 5: Add Security Groups for vSRX | 43

Launch a vSRX Instance on an Amazon Virtual Private Cloud | 45

Step 1: Create an SSH Key Pair | 46

Step 2: Launch a vSRX Instance | 48

Step 3: View the AWS System Logs | 52

Step 4: Add Network Interfaces for vSRX | 52

Step 5: Allocate Elastic IP Addresses | 54

Step 6: Add the vSRX Private Interfaces to the Route Tables | 54

Step 7: Reboot the vSRX Instance | 55

Step 8: Log in to a vSRX Instance | 56

Using Cloud-Init to Automate the Initialization of vSRX Instances in AWS | 57

Upgrade Junos OS Software on a vSRX Instance | 59

Upgrade the Junos OS for vSRX Software Release | 60

Replace the vSRX Instance on AWS | 60

Remove a vSRX Instance on AWS | 61

3

Configuring and Managing vSRX

vSRX Configuration and Management Tools | 63

Configure vSRX Using the CLI | 64

Understand vSRX on AWS Preconfiguration and Factory Defaults | 64

Add a Basic vSRX Configuration | 65

Add DNS Servers | 68

Add vSRX Feature Licenses | 68

Configure vSRX Using the J-Web Interface | 69

Access the J-Web Interface and Configure vSRX | 69

Apply the Configuration Settings for vSRX | 71

Add vSRX Feature Licenses | 72

Managing Security Policies for Virtual Machines Using Junos Space Security Director | 72

AWS Elastic Load Balancing and Elastic Network Adapter | 73

Overview of AWS Elastic Load Balancing | 74

Overview of Application Load Balancer | 75

Deployment of AWS Application Load Balancer | 77

Invoking Cloud Formation Template (CFT) Stack Creation for vSRX Behind AWS Application Load Balancer Deployment | 81

Overview of AWS Elastic Network Adapter (ENA) for vSRX Instances | 90

Software Receive Side Scaling | 91

Overview | 91

Understanding Software Receive Side Scaling Configuration | 92

Multi-Core Scaling Support on AWS with SWRSS and ENA | 93

GTP Traffic with TEID Distribution and SWRSS | 94

- Overview GTP Traffic Distribution with TEID Distribution and SWRSS | 94
- Enabling GTP-U TEID Distribution with SWRSS for Asymmetric Fat Tunnels | 96

Centralized Monitoring and Troubleshooting using AWS Features | 98

- Understanding Centralized Monitoring Using Cloudwatch | 99
- Integration of vSRX with AWS Monitoring and Troubleshooting Features | 103
 - Enable Monitoring of vSRX Instances with AWS CloudWatch Metric | 103
 - Collect, Store, and View vSRX Logs to AWS CloudWatch | 104
 - Enable and Configure Security Hub on vSRX | 105
 - Grant Permission for vSRX to access AWS CloudWatch and Security Hub | 106

Deploying vSRX 3.0 for Securing Data using AWS KMS | 108

- Integrate AWS KMS with vSRX 3.0 | 109
- AWS Cloud Formation Templates | 111

4

vSRX in AWS Use Cases

Example: Configuring NAT for vSRX | 119

- Before You Begin | 119
- Overview | 119
- Configuration | 120
- Configuring NAT | 120

Example: Configure VPN on vSRX Between Amazon VPCs | 121

- Before You Begin | 122
- Overview | 122
- vSRX1 VPN Configuration | 122
- Verification | 126

Example: Configure Juniper Sky ATP for vSRX | 127

- Before You Begin | 127
- Overview | 128
- Juniper Sky ATP Configuration | 128

5

Monitoring and Troubleshooting**Monitoring | 131****Backup and Recovery | 132****Finding the Software Serial Number for vSRX | 133**

About This Guide

Use this guide to install the vSRX Virtual Firewall in a virtual private cloud (VPC) in the Amazon Web Services (AWS) cloud. 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

[vSRX Overview | 2](#)

[Understand vSRX with AWS | 5](#)

[Requirements for vSRX on AWS | 12](#)

[Junos OS Features Supported on vSRX | 16](#)

vSRX Overview

SUMMARY

In this topic you learn about vSRX architecture and its benefits.

IN THIS SECTION

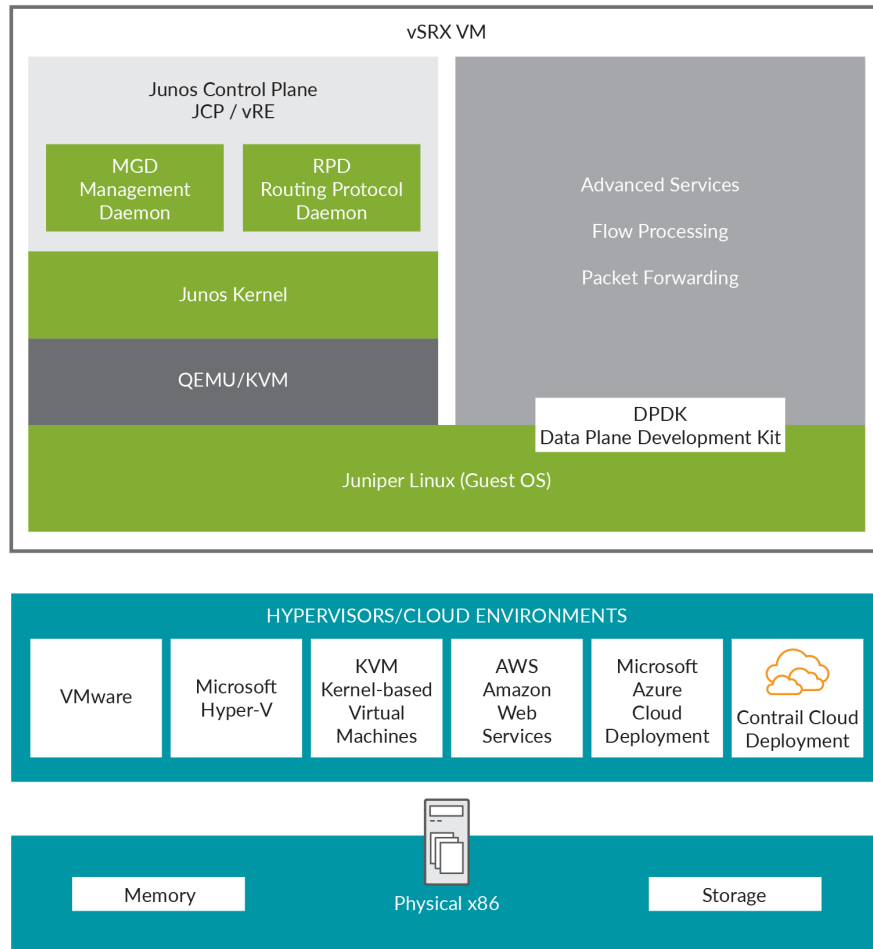
- [Benefits | 5](#)

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 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.

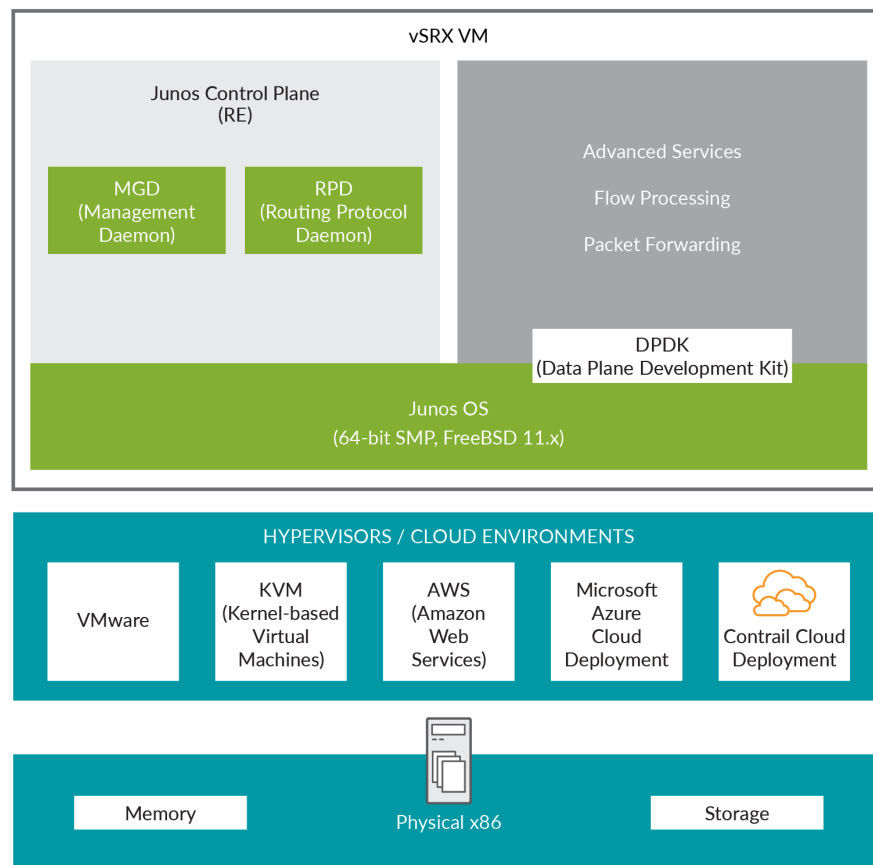
In vSRX 3.0 architecture, FreeBSD 11.x is used as the guest OS and the Routing Engine and Packet Forwarding Engine runs on FreeBSD 11.x as single virtual machine for improved performance and scalability. vSRX 3.0 uses DPDK to process the data packets in the data plane. A direct Junos upgrade from vSRX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

Figure 2: vSRX 3.0 Architecture



Benefits

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

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

- *Stateful firewall* protection at the tenant edge
- Faster deployment of virtual firewalls into new sites
- Ability to run on top of various hypervisors and public cloud infrastructures
- Full routing, *VPN*, core security, and networking capabilities
- Application security features (including IPS and App-Secure)
- Content security features (including Anti Virus, Web Filtering, Anti Spam, and Content Filtering)
- Centralized management with Junos Space Security Director and local management with J-Web Interface
- Juniper Networks Sky Advanced Threat Prevention (Sky ATP) integration

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, multi-core vSRX supports scaling vCPUs and virtual RAM (vRAM). Additional vCPUs are applied to the data plane to increase performance.

Understand vSRX with AWS

IN THIS SECTION

- [vSRX with AWS | 6](#)
- [AWS Glossary | 8](#)

This section presents an overview of vSRX on Amazon Web Services (AWS).

vSRX with AWS

AWS provides on-demand services in the cloud. Services range from Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), to Application and Database as a Service. AWS is a highly flexible, scalable, and reliable cloud platform. In AWS, you can host servers and services on the cloud as a pay-as-you-go (PAYG) or bring-your-own-license (BYOL) service.

NOTE: vSRX PAYG images do not require any Juniper Networks licenses.

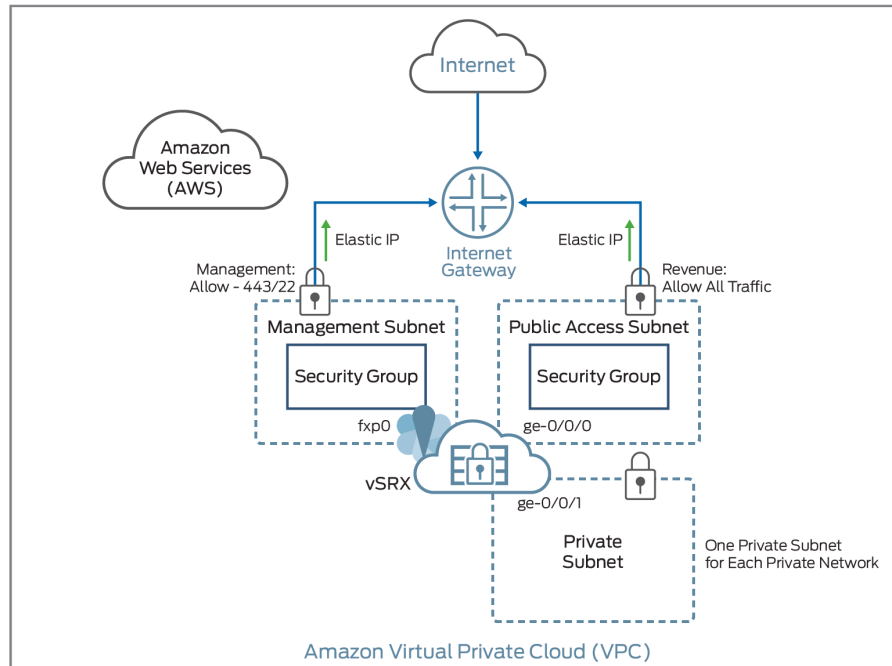
vSRX can be deployed in a virtual private cloud (VPC) in the Amazon Web Services (AWS) cloud. You can launch vSRX as an Amazon Elastic Compute Cloud (EC2) instance in an Amazon VPC dedicated to a specific user account. The vSRX Amazon Machine Image (AMI) uses hardware virtual machine (HVM) virtualization.

[Figure 3 on page 7](#) shows an example of deploying a vSRX instance to provide security for applications running in a private subnet of an Amazon VPC.

In the Amazon VPC, public subnets have access to the Internet gateway, but private subnets do not. vSRX requires two public subnets and one or more private subnets for each individual instance group. The public subnets consist of one for the management interface (fxp0) and one for a revenue (data)

interface. The private subnets, connected to the other vSRX interfaces, ensure that all traffic between applications on the private subnets and the Internet must pass through the vSRX instance.

Figure 3: vSRX in AWS Deployment



AWS Marketplace also enables you to discover and subscribe to software that supports regulated workloads through AWS Marketplace for AWS GovCloud (US).

Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, vSRX supports two bundles for PAYG that are available as 1-hour or 1-year subscriptions.

- vSRX Next Generation Firewall—Includes standard (STD) features of core security, including core firewall, IPsec VPN, NAT, CoS, and routing services, as well as advanced Layer 4 through 7 security services such as AppSecure features of AppID, AppFW, AppQoS, and AppTrack, IPS and rich routing capabilities.
- vSRX Premium-Next Generation Firewall with Anti-Virus Protection—Includes the features in the vSRX Next- Generation Firewall package, including the UTM antivirus feature.

You deploy vSRX in an Amazon Virtual Private Cloud (Amazon VPC) as an application instance in the Amazon Elastic Compute Cloud (Amazon EC2). Each Amazon EC2 instance is deployed, accessed, and configured over the Internet using the AWS Management Console, and the number of instances can be scaled up or down as needed.

NOTE: In the current release, each vSRX instance uses two vCPUs and 4 GB of memory, even if the instance type selected on AWS provides more resources.

vSRX uses hardware assisted virtual machines (HVM) for high performance (enhanced networking), and supports the following deployments on AWS cloud environments:

- As a firewall between other Amazon EC2 instances on your Amazon VPC and the Internet
- As a VPN endpoint between your corporate network and your Amazon VPC
- As a firewall between Amazon EC2 instances on different subnets

There are default limits for AWS services for an AWS account. For more information on AWS service limits, see https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html and <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html>.

AWS Glossary

This section defines some common terms used in an AWS configuration. [Table 1 on page 8](#) defines common terms used for Amazon Virtual Private Cloud (Amazon VPC) and [Table 2 on page 10](#) defines common terms for Amazon Elastic Compute Cloud (Amazon EC2) services.

Table 1: Amazon VPC Related Terminology

Term	Description
Internet gateways	Amazon VPC components that allow communications between your instances in the Amazon VPC and the Internet.

Table 1: Amazon VPC Related Terminology (Continued)

Term	Description
IP addressing	<p>AWS includes three types of IP address:</p> <ul style="list-style-type: none"> • Public IP address—Addresses obtained from a public subnet that is publicly routable from the Internet. Public IP addresses are mapped to primary private IP addresses through AWS NAT. • Private IP address—IP addresses in the Amazon VPC Classless Interdomain Routing (CIDR) range, as specified in RFC 1918, that are not publicly routable. • Elastic IP address—A static IP address designed for dynamic cloud computing. When an Elastic IP address is associated with a public IP network interface, the public IP address associated is released until the Elastic IP address is disassociated from the network interface. <p>Each network interface can be associated with multiple private IP addresses. Public subnets can have multiple private IP addresses, public addresses, and Elastic IP addresses associated with the private IP address of the network interface. Instances in private and public subnets can have multiple private IP addresses. One Elastic IP address can be associated with each private IP address for instances in public subnets.</p> <p>You can assign static private IP addresses in the subnet. The first five IP addresses and the last IP address in the subnet are reserved for Amazon VPC networking and routing. The first IP address is the gateway for the subnet.</p>
Network ACL	<p>AWS stateless virtual firewall operating at the subnet level.</p>
Route tables	<p>A set of routing rules used to determine where the network traffic is directed. Each subnet needs to be associated with a route table. Subnets not explicitly associated with a route table are associated with the main route table.</p> <p>Custom route tables can be created other than the default table.</p>

Table 1: Amazon VPC Related Terminology (Continued)

Term	Description
Subnet	<p>A virtual addressing space in the Amazon VPC CIDR block. The IP addresses for the Amazon EC2 instances are allocated from the subnet pool of IP addresses.</p> <p>You can create two types of subnets in the Amazon VPC:</p> <ul style="list-style-type: none"> • Public subnets–Subnets that have traffic connections to the Internet gateway. • Private subnets–Subnets that do not have connections to the Internet gateway <p>NOTE: With vSRX Network Address Translation (NAT) , you can launch all customer instances in private subnets and connect vSRX interfaces to the Internet. This protects customer instances from being directly exposed to Internet traffic.</p>
VPC	Virtual private cloud.

Table 2: Amazon EC2 Related Terminology

Term	Description
Amazon Elastic Block Store (EBS)	Persistent block storage that can be attached to an Amazon EC2 instance. Block storage volumes can be formatted and mounted on an instance. Amazon EBS optimized instances provide dedicated throughput between Amazon EC2 and Amazon EBS.
Amazon Elastic Compute Cloud (EC2)	Amazon Web service that enables launch and management of elastic virtual servers or computers that run on the Amazon infrastructure.
Amazon Machine Image (AMI)	Amazon image format that contains the information, such as the template for root volume, launch permissions, and block device mapping, that is required to launch an Amazon EC2 instance.
Elastic IP	A static IP designed for dynamic cloud computing. The public IP is mapped to the private subnet IP using NAT.

Table 2: Amazon EC2 Related Terminology (Continued)

Term	Description
Enhanced networking	Provides high packet per second performance, low latency, higher I/O performance, and lower CPU utilization compared to traditional implementations. vSRX leverages this networking with hardware virtualized machine (HVM) Amazon Machine Images (AMIs).
Instance	A virtual machine or server on Amazon EC2 that uses XEN or, XEN-HVM hypervisor types. Amazon EC2 provides a selection of instances optimized for different use cases.
Key pairs	<p>Public key cryptography used by AWS to encrypt and decrypt login information. Create these key pairs using AWS-EC2 or import your own key pairs.</p> <p>NOTE: AWS does not accept DSA. Limit the public key access permissions to 400.</p> <p>For more information on key rotation, see https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html.</p>
Network interfaces	<p>Virtual network interfaces that you can attach to an instance in the Amazon VPC. An Elastic Network Interface (ENI) can have a primary private IP address, multiple secondary IP addresses, one Elastic IP address per private IP address, one public IP address, one or more security groups, one MAC address, and a source/destination check flag.</p> <p>For vSRX instances, disable the source/destination check for all interfaces.</p> <p>NOTE: ENIs use the IP addresses within the subnet range. So, the ENI IP addresses are not exhausted.</p>
Network MTU	<p>All Amazon instance types support an MTU of 1500. Some instance types support jumbo frames (9001 MTU).</p> <p>NOTE: Use C3, C4, C5, CC2, M3, M4, or T2 AWS instance types for vSRX instances with jumbo frames.</p>
Placement Groups	Instances launched in a common cluster placement group. Instances within the cluster have networks with high bandwidth and low latency.

Table 2: Amazon EC2 Related Terminology (Continued)

Term	Description
Security groups	<p>An AWS-provided virtual firewall that controls the traffic for one or more instances. Security groups can be associated with an instance only at launch time.</p> <p>NOTE: Because vSRX manages your firewall settings, we recommend that you ensure there is no contradiction between rule sets on AWS security groups and rule sets in your vSRX configuration.</p>

Release History Table

Release	Description
15.1X49-D70	Starting in Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, vSRX supports two bundles for PAYG that are available as 1-hour or 1-year subscriptions.

RELATED DOCUMENTATION

[AWS Tutorials](#)

[Getting Started with AWS](#)

Requirements for vSRX on AWS

IN THIS SECTION

- [Minimum System Requirements for AWS | 13](#)
- [Interface Mapping for vSRX on AWS | 13](#)
- [vSRX Default Settings on AWS | 15](#)
- [Best Practices for Improving vSRX Performance | 15](#)

This section presents an overview of requirements for deploying a vSRX instance on Amazon Web Services (AWS).

Minimum System Requirements for AWS

Table 3 on page 13 lists the minimum system requirements for vSRX instances to be deployed on AWS.

Table 3: Minimum System Requirements for vSRX

Component	Specification and Details
Hypervisor support	XEN-HVM
Memory	4 GB
Disk space	16 GB
vCPUs	2
vNICs	3
vNIC type	SR-IOV

Interface Mapping for vSRX on AWS

vSRX on AWS supports up to a maximum of eight network interfaces, but the actual maximum number of interfaces that can be attached to a vSRX instance is dictated by the AWS instance type in which it is launched. For AWS instances that allow more than eight interfaces, vSRX will support up to a maximum of eight interfaces only.

For more information on maximum network interfaces by instance type, see <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html>.

Table 4 on page 14 shows a mapping between vSRX interface names and their corresponding AWS interface names for up to eight network interfaces. The first network interface is used for the out-of-band management (fxp0) for vSRX.

Table 4: vSRX and AWS Interface Names

Interface Number	vSRX Interface	AWS Interface
1	fxp0	eth0
2	ge-0/0/0	eth1
3	ge-0/0/1	eth2
4	ge-0/0/2	eth3
5	ge-0/0/3	eth4
6	ge-0/0/4	eth5
7	ge-0/0/5	eth6
8	ge-0/0/6	eth7

We recommend putting revenue interfaces in routing instances as a best practice to avoid asymmetric routing. Since fxp0 is part of the default (inet.0) routing table, there might be two default routes needed in the same routing instance: one for the fxp0 interface for external management access, and the other for the revenue interfaces for traffic access, resulting in asymmetric routing. Putting the revenue interfaces in a separate routing instance avoids this situation of two default routes in a single routing instance.

NOTE: Ensure that interfaces belonging to the same security zone are in the same routing instance. See [KB Article - Interface must be in the same routing instance as the other interfaces in the zone.](#)

vSRX Default Settings on AWS

vSRX requires the following basic configuration settings:

- Interfaces must be assigned IP addresses.
- Interfaces must be bound to zones.
- Policies must be configured between zones to permit or deny traffic.
- The ENA driver-related component must be ready for vSRX.

[Table 5 on page 15](#) lists the factory-default settings for security policies on the vSRX.

Table 5: Factory-Default Settings for Security Policies

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



CAUTION: Do not use the **load factory-default** command on a vSRX AWS instance. The factory-default configuration removes the AWS preconfiguration. If you must revert to factory default, ensure that you manually reconfigure AWS preconfiguration statements before you commit the configuration; otherwise, you will lose access to the vSRX instance. See "[Configure vSRX Using the CLI](#)" on page 64 for AWS preconfiguration details.

Best Practices for Improving vSRX Performance

Review the following deployment practices to improve vSRX performance:

- Disable the source/destination check for all vSRX interfaces.
- Limit public key access permissions to 400 for key pairs.
- Ensure that there are no contradictions between AWS security groups and your vSRX configuration.
- Use the c5n instance types on AWS for best throughput on the vSRX.

- Ensure traffic flows through multiple interfaces of the vSRX for optimal usage of the vCPUs.
- Use vSRX NAT to protect your Amazon EC2 instances from direct Internet traffic.

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 | 16](#)
- [SRX Series Features Not Supported on vSRX | 21](#)

SRX Series Features Supported on vSRX

vSRX inherits most of the branch SRX Series features with the following considerations shown in [Table 6 on page 16](#).

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 6: vSRX Feature Considerations

Feature	Description
IDP	<p>The IDP feature is subscription based and must be purchased. After purchase, you can activate the IDP feature with the license key.</p> <p>For SRX Series IDP configuration details, see:</p> <p>Understanding Intrusion Detection and Prevention for SRX Series</p>

Table 6: vSRX Feature Considerations (Continued)

Feature	Description	
IPSec VPNs	<p>Starting in Junos OS Release 19.3R1, vSRX supports the following authentication algorithms and encryption algorithms:</p> <ul style="list-style-type: none"> • Authentication algorithm: hmac-sha1-96 and HMAC-SHA-256-128 authentication • Encryption algorithm: aes-128-cbc <p>Starting in Junos OS Release 20.3R1, vSRX supports 10,000 IPsec VPN tunnels.</p> <p>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.</p> <p>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.</p> <p>You can configure the number of vCPUs allocated to Junos Routing Engine using the set security forwarding-options resource-manager cpu re <value>.</p> <p>NOTE: 64 G memory is required to support 10000 tunnels in PMI mode.</p> <p>[See show security ipsec security-associations, show security ike tunnel-map, and show security ipsec tunnel-distribution.]</p>	
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 6: 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	<p>Starting in Junos OS Release 20.1R1, you can configure logical systems and tenant systems on vSRX and vSRX 3.0 instances.</p> <p>With Junos OS, you can partition a single security device into multiple logical devices that can perform independent tasks.</p> <p>Each logical system has its own discrete administrative domain, logical interfaces, routing instances, security firewall and other security features.</p> <p>See Logical Systems Overview.</p>	

Table 6: vSRX Feature Considerations (*Continued*)

Feature	Description
PowerMode IPsec	<p data-bbox="496 369 1398 554">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.</p> <p data-bbox="496 590 971 619">Supported Features in PowerMode IPsec</p> <ul data-bbox="496 653 922 1276" style="list-style-type: none"> <li data-bbox="496 653 748 682">• IPsec functionality <li data-bbox="496 720 716 749">• Traffic selectors <li data-bbox="496 787 862 816">• Secure tunnel interface (st0) <li data-bbox="496 854 922 884">• All control plane IKE functionality <li data-bbox="496 921 883 951">• Auto VPN with traffic selector <li data-bbox="496 989 906 1018">• Auto VPN with routing protocol <li data-bbox="496 1056 586 1085">• IPv6 <li data-bbox="496 1123 808 1152">• Stateful Layer 4 firewall <li data-bbox="496 1190 727 1220">• High-Availability <li data-bbox="496 1257 610 1287">• NAT-T <p data-bbox="496 1320 1029 1350">Non-Supported Features in PowerMode IPsec</p> <ul data-bbox="496 1383 894 1797" style="list-style-type: none"> <li data-bbox="496 1383 586 1413">• NAT <li data-bbox="496 1451 691 1480">• IPsec in IPsec <li data-bbox="496 1518 748 1547">• GTP/SCTP firewall <li data-bbox="496 1585 894 1614">• Application firewall/AppSecure <li data-bbox="496 1652 586 1682">• QoS <li data-bbox="496 1719 699 1749">• Nested tunnel <li data-bbox="496 1787 610 1816">• Screen

Table 6: vSRX Feature Considerations (*Continued*)

Feature	Description
	<ul style="list-style-type: none"> • Multicast • Host traffic
Tenant Systems	<p>Starting in Junos OS Release 20.1R1, you can configure tenant systems on vSRX and vSRX 3.0 instances.</p> <p>A tenant system provides logical partitioning of the SRX device into multiple domains similar to logical systems and provides high scalability.</p> <p>See Tenant Systems Overview.</p>
Transparent mode	<p>The known behaviors for transparent mode support on vSRX are:</p> <ul style="list-style-type: none"> • The default MAC learning table size is restricted to 16,383 entries. <p>For information about configuring transparent mode for vSRX, see Layer 2 Bridging and Transparent Mode Overview.</p>

Table 6: vSRX Feature Considerations (Continued)

Feature	Description
UTM	<ul style="list-style-type: none"> • 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. <p>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.</p> <p>[See show security monitoring and show security forward-options resource-manager settings.]</p>

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 7 on page 22](#) 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 7: 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 7: 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 7: SRX Series Features Not Supported on vSRX (Continued)

SRX Series Feature	vSRX Notes
Ethernet link fault management	
Physical interface (encapsulations) <ul style="list-style-type: none"> • ethernet-ccc • ethernet-tcc • extended-vlan-ccc • extended-vlan-tcc 	Not supported
Interface family <ul style="list-style-type: none"> • 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 7: SRX Series Features Not Supported on vSRX (Continued)

SRX Series Feature	vSRX Notes
IEEE 802.1X port-based authentication control with multisuppllicant 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 7: 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 7: 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	
Circuit 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 7: 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 <i>gr</i> , <i>ip</i> , and <i>st0</i> . Packet capture is not supported on redundant Ethernet interfaces (<i>reth</i>).
Routing	
BGP extensions for IPv6	Not supported
BGP Flowspec	Not supported
BGP route reflector	Not supported
C RTP	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 7: 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 AWS

[Configure an Amazon Virtual Private Cloud for vSRX](#) | 31

[Launch a vSRX Instance on an Amazon Virtual Private Cloud](#) | 45

[Using Cloud-Init to Automate the Initialization of vSRX Instances in AWS](#) | 57

[Upgrade Junos OS Software on a vSRX Instance](#) | 59

[Remove a vSRX Instance on AWS](#) | 61

Configure an Amazon Virtual Private Cloud for vSRX

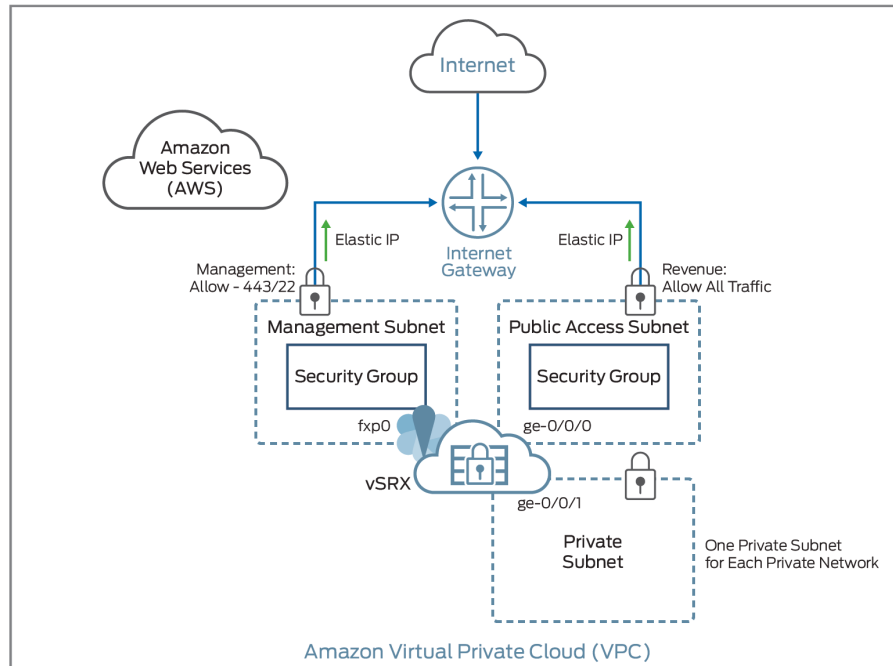
IN THIS SECTION

- Step 1: Create an Amazon VPC and Internet Gateway | 32
- Step 2: Add Subnets for vSRX | 35
- Step 3: Attach an interface to a Subnet | 36
- Step 4: Add Route Tables for vSRX | 40
- Step 5: Add Security Groups for vSRX | 43

Before you begin, you need an Amazon Web Services (AWS) account and an Identity and Access Management (IAM) role, with all required permissions to access, create, modify, and delete Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (S3), and Amazon Virtual Private Cloud (Amazon VPC) objects. You should also create access keys and corresponding secret access keys, X.509 certificates, and account identifiers. For better understanding of AWS terminologies and their use in vSRX AWS deployments, see "[Understand vSRX with AWS](#)" on page 5.

Figure 4 on page 32 shows an example of how you can deploy vSRX to provide security for applications running in a private subnet of an Amazon VPC.

Figure 4: Example of vSRX in AWS Deployment

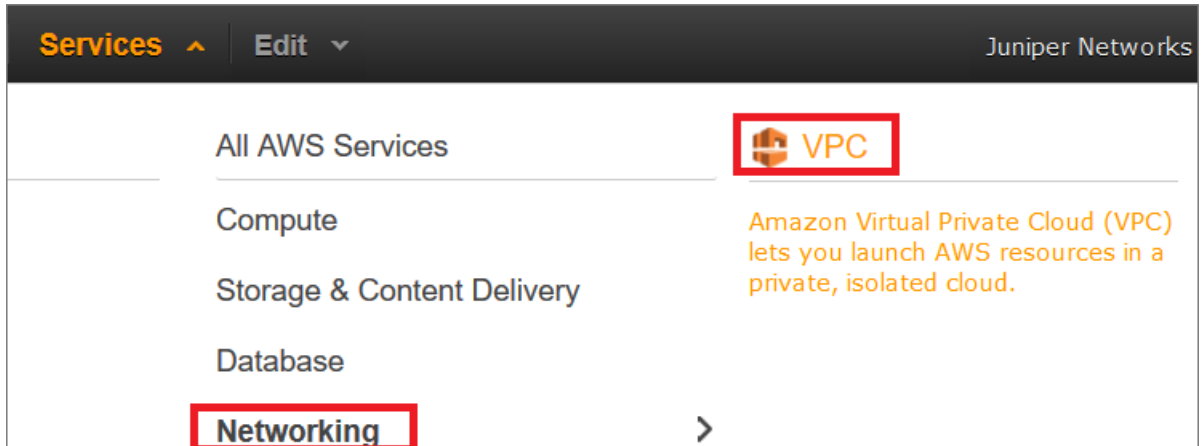


The following procedures outline how to create and prepare an Amazon VPC for vSRX. The procedures describe how to set up an Amazon VPC with its associated Internet gateway, subnets, route table, and security groups.

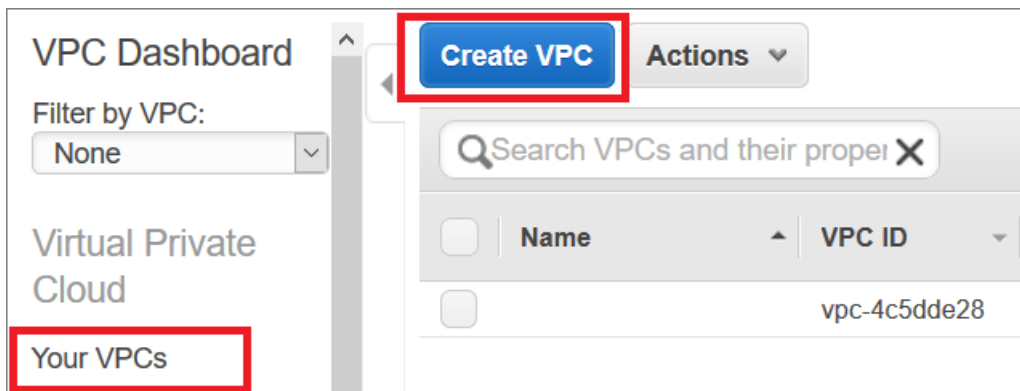
Step 1: Create an Amazon VPC and Internet Gateway

Use the following procedure to create an Amazon VPC and an Internet gateway. If you have already have a VPC and an Internet gateway, go to "[Step 2: Add Subnets for vSRX](#)" on page 35.

1. Log in to the AWS Management Console and select **Services > Networking > VPC**.



2. In the VPC Dashboard, select **Your VPCs** in the left pane, and click **Create VPC**.



3. Specify a VPC name and a range of private IP addresses in Classless Interdomain Routing (CIDR) format. Leave Default as the Tenancy.

Create VPC ✕

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. Use the Classless Inter-Domain Routing (CIDR) block format to specify your VPC's contiguous IP address range, for example, 10.0.0.0/16. You cannot create a VPC larger than /16.

Name tag i

CIDR block i

Tenancy i

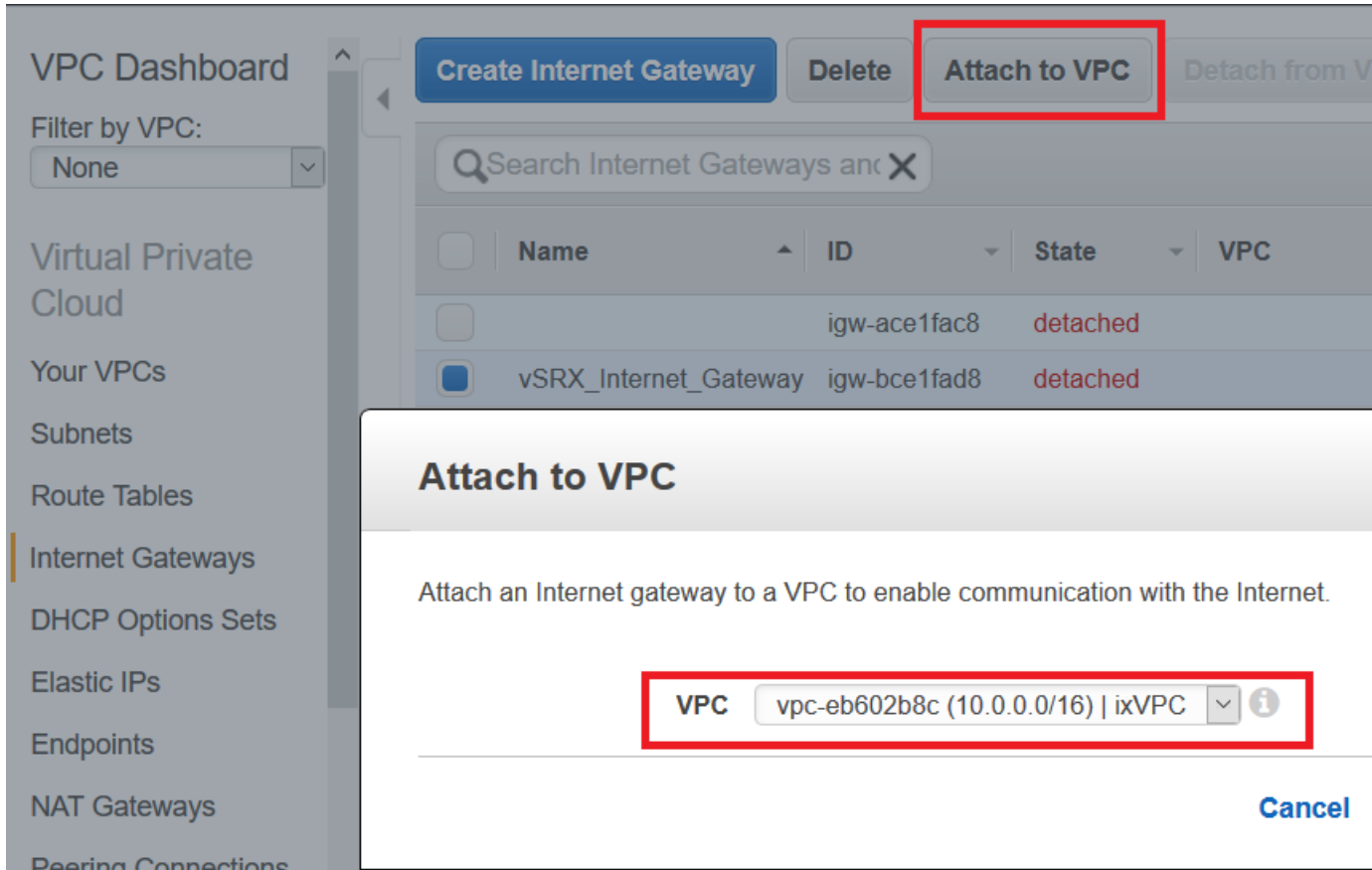
Cancel
Yes, Create

4. Click **Yes, Create**.

5. Select **Internet Gateways** in the left pane, and click **Create Internet Gateway**.



6. Specify a gateway name and click **Yes, Create**.
7. Select the gateway you just created and click **Attach to VPC**.
8. Select the new Amazon VPC, and click **Yes, Attach**.



Step 2: Add Subnets for vSRX

In the Amazon VPC, public subnets have access to the Internet gateway, but private subnets do not. vSRX requires two public subnets and one or more private subnets for each individual instance group. The public subnets consist of one for the management interface (fxp0) and one for a revenue (data) interface. The private subnets, connected to the other vSRX interfaces, ensure that all traffic between applications on the private subnets and the Internet must pass through the vSRX instance.

To create each vSRX subnet:

1. In the VPC Dashboard, select **Subnets** in the left pane, and click **Create Subnet**.
2. Specify a subnet name, select the Amazon VPC and availability zone, and specify the range of subnet IP addresses in CIDR format.

TIP: As a naming convention best practice for subnets, we recommend including **private** or **public** in the name to make it easier to know which subnet is public or private.

NOTE: All subnets for a vSRX instance must be in the same availability zone. Do not use **No Preference** for the availability zone.

3. Click **Yes, Create**.

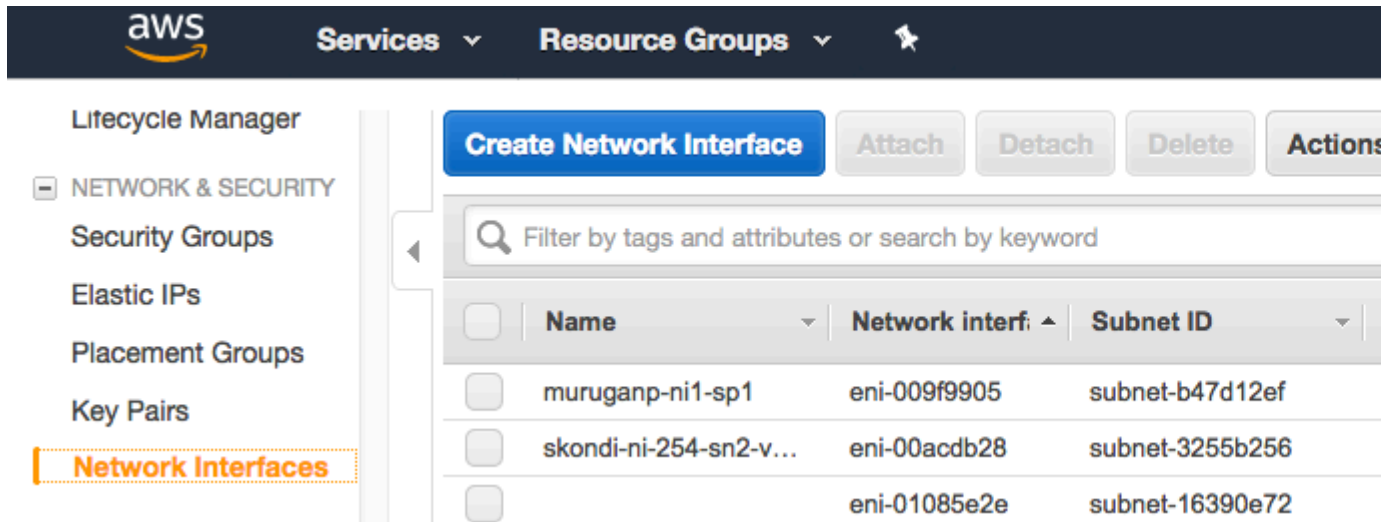
The screenshot shows the AWS Management Console interface. On the left, the 'VPC Dashboard' sidebar is visible, with 'Subnets' highlighted in a red box. The main content area shows the 'Create Subnet' dialog box. The dialog box has a title bar with 'Create Subnet' and a 'Subnet Actions' dropdown. Below the title bar, there is a text instruction: 'Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that the CIDR block must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the only subnet in your VPC.' Below this text are four input fields: 'Name tag' with the value 'iXmgm_subnet_254', 'VPC' with a dropdown menu showing 'vpc-eb602b8c (10.0.0.0/16) | ixVPC', 'Availability Zone' with a dropdown menu showing 'us-east-1a', and 'CIDR block' with the value '10.0.254.0/24'. Each input field has an information icon (i) to its right. At the bottom right of the dialog box, there is a 'Cancel' button.

Repeat these steps for each subnet you want to create and attach to the vSRX instance.

Step 3: Attach an interface to a Subnet

To attach an interface to a subnet:

1. Create a network interface from the Amazon EC2 home page.
Click the Network Interface option on the EC2 home page and the **Create Network Interface** page opens.



The screenshot shows the AWS Management Console interface for Network Interfaces. The top navigation bar includes the AWS logo, 'Services', and 'Resource Groups'. The left sidebar lists various services, with 'Network Interfaces' highlighted in orange. The main content area features a blue 'Create Network Interface' button, along with 'Attach', 'Detach', 'Delete', and 'Actions' buttons. Below these is a search bar and a table of existing network interfaces.

<input type="checkbox"/>	Name	Network interf.	Subnet ID
<input type="checkbox"/>	muruganp-ni1-sp1	eni-009f9905	subnet-b47d12ef
<input type="checkbox"/>	skondi-ni-254-sn2-v...	eni-00acdb28	subnet-3255b256
<input type="checkbox"/>		eni-01085e2e	subnet-16390e72

2. Click the **Create Network Interface** option, fill in the required information in the fields, and then click **Create**.

[Network interfaces](#) > Create Network Interface

Create Network Interface

Description ⓘ

Subnet* ↕ ⓘ ⓘ

IPv4 Private IP Auto-assign ⓘ
 Custom

Security groups* ⓘ

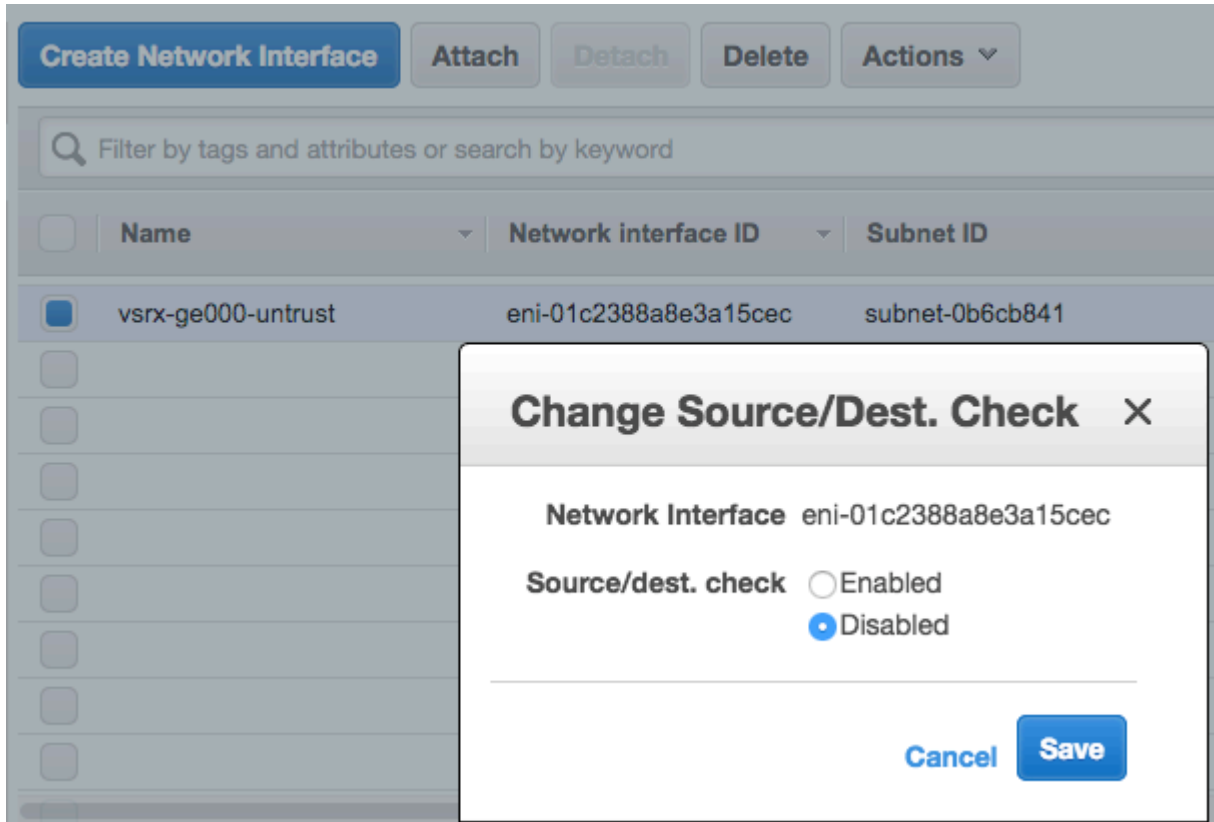
🔍 Filter by attributes or search by keyword

<input type="checkbox"/>	Group ID	Group name	Description
<input checked="" type="checkbox"/>	sg-082cb06f	ha1-to-ha2	ha1-to-ha2
<input type="checkbox"/>	sg-0cc75b6b	ha2-local	ha2-local
<input type="checkbox"/>	sg-70eb7b17	default	default VPC security
<input type="checkbox"/>	sg-cefd61a9	ha2-jnpr	ha2-jnpr

* Required

- Find and select your newly created interface.

If this interface is the revenue interface, then select **Change Source/Dest.Check** from the **Action** menu, choose **Disabled**, and click **Save**. If this interface is your fxp0 interface then skip this disabling step.



The screenshot displays the AWS Management Console interface for network interfaces. At the top, there are buttons for 'Create Network Interface', 'Attach', 'Detach', 'Delete', and 'Actions'. Below these is a search bar labeled 'Filter by tags and attributes or search by keyword'. A table lists network interfaces with columns for 'Name', 'Network interface ID', and 'Subnet ID'. The first row is selected, showing 'vsrx-ge000-untrust', 'eni-01c2388a8e3a15cec', and 'subnet-0b6cb841'. A modal dialog titled 'Change Source/Dest. Check' is open over the first row. The dialog shows the 'Network Interface' as 'eni-01c2388a8e3a15cec' and the 'Source/dest. check' option set to 'Disabled' (indicated by a selected radio button). The dialog also includes 'Cancel' and 'Save' buttons.

Name	Network interface ID	Subnet ID
vsrx-ge000-untrust	eni-01c2388a8e3a15cec	subnet-0b6cb841

Change Source/Dest. Check

Network Interface eni-01c2388a8e3a15cec

Source/dest. check Enabled
 Disabled

Cancel Save

4. Click **Attach** from the menu on top of the screen, choose the **Instance ID** of your vSRX instance, and click **Attach**.

The screenshot shows the AWS Management Console interface for network interfaces. At the top, there are buttons for 'Create Network Interface', 'Attach', 'Detach', 'Delete', and 'Actions'. Below these is a search bar. A table lists several network interfaces with columns for Name, Network interface ID, Subnet ID, and VPC ID. The interface 'eni-01c2388a8e3a15cec' is selected. A modal dialog titled 'Attach Network Interface' is open, showing the selected network interface ID and a dropdown menu for the Instance ID, which is currently set to 'i-041bb7c0f08ddf96b (stopped)'. The dialog includes 'Cancel' and 'Attach' buttons.

Name	Network interface ID	Subnet ID	VPC ID
	eni-00247dc5f6da36918	subnet-f99cad5	vpc-f2a
	eni-007f6d74fc4828e43	subnet-07f9f349680a6c64a	vpc-fba
	eni-0151c8c0b4725ba4a	subnet-0c5f9c32	vpc-70
<input checked="" type="checkbox"/> vsrx-ge000-untrust	eni-01c2388a8e3a15cec	subnet-0b6cb841	vpc-f0

5. vSRX does not support interface hot plug-in. So, when you are done adding the interfaces, reboot the vSRX instances on which the interfaces were added, to apply the changes to take effect.

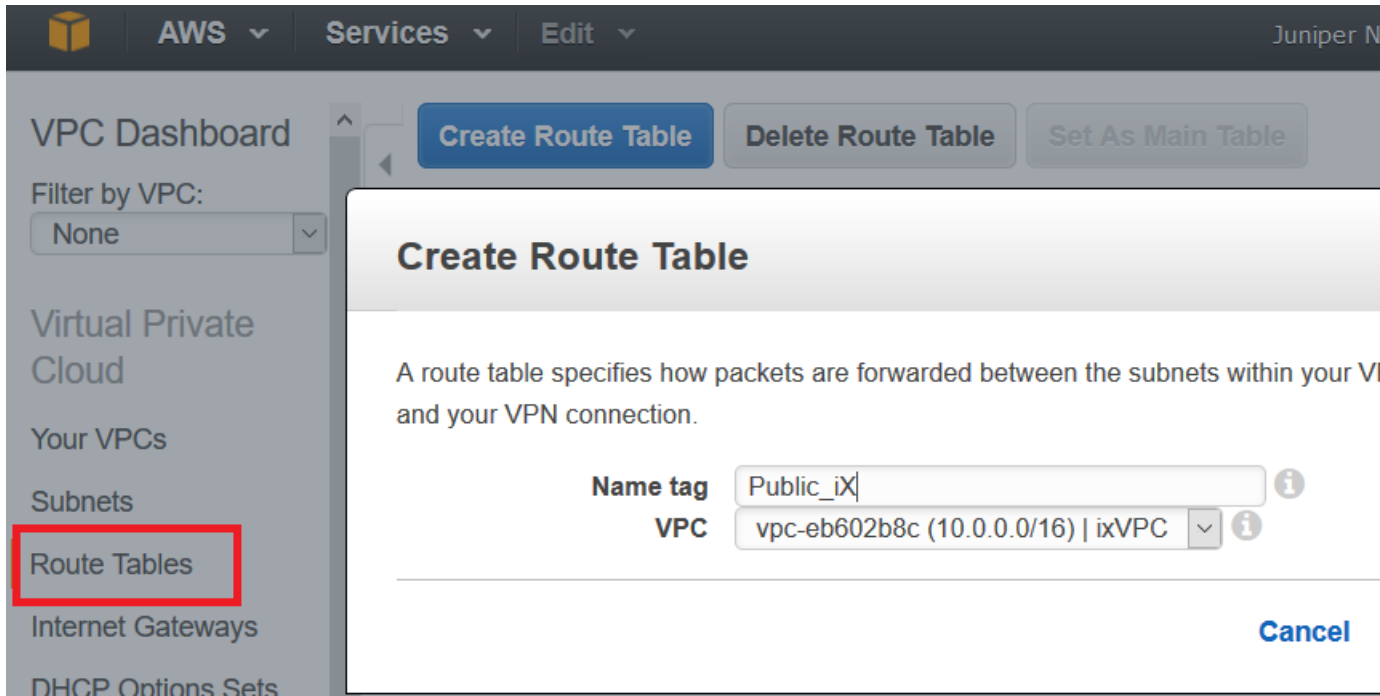
Step 4: Add Route Tables for vSRX

A main route table is created for each Amazon VPC by default. We recommend that you create a custom route table for the public subnets and a separate route table for each private subnet. All subnets that are not associated with a custom route table are associated with the main route table.

To create the route tables:

1. In the VPC Dashboard, select **Route Tables** in the left pane, and click **Create Route Table**.
2. Specify a route table name, select the VPC, and click **Yes, Create**.

TIP: As a naming convention best practice for route tables, we recommend including **private** or **public** in the name to make it easier to know which route table is public or private.



3. Repeat steps 1 and 2 to create all the route tables.
4. Select the route table you created for the public subnets and do the following:
 - a. Select the **Routes** tab below the list of route tables.
 - b. Click **Edit** and click **Add another route**.
 - c. Enter **0.0.0.0/0** as the destination, select your VPC internet gateway as the target, and click **Save**.

Public_iX rtb-02a60c64 0 Subnets No vpc-eb602b8c (10...

rtb-02a60c64 | Public_iX

Summary Routes Subnet Associations Route Propagation Tags

Cancel Save

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="igw-bce1fad8"/>		No	<input type="button" value="✕"/>

- d. Select the **Subnet Associations** tab, and click **Edit**.
- e. Select the check boxes for the public subnets, and click **Save**.

Public_iX rtb-02a60c64 0 Subnets No vpc-eb602b8c (10...

rtb-02a60c64 | Public_iX

Summary Routes Subnet Associations Route Propagation Tags

Cancel Save

Associate	Subnet	CIDR	Current Route Table
<input checked="" type="checkbox"/>	subnet-6cbd6025 (10.0.10.0/24) iXpublic_subnet_10	10.0.10.0/24	Main
<input type="checkbox"/>	subnet-19bd6050 (10.0.20.0/24) ixPrivate_subnet_20	10.0.20.0/24	Main
<input checked="" type="checkbox"/>	subnet-b7825ffe (10.0.254.0/24) iXmgm_subnet_254	10.0.254.0/24	Main

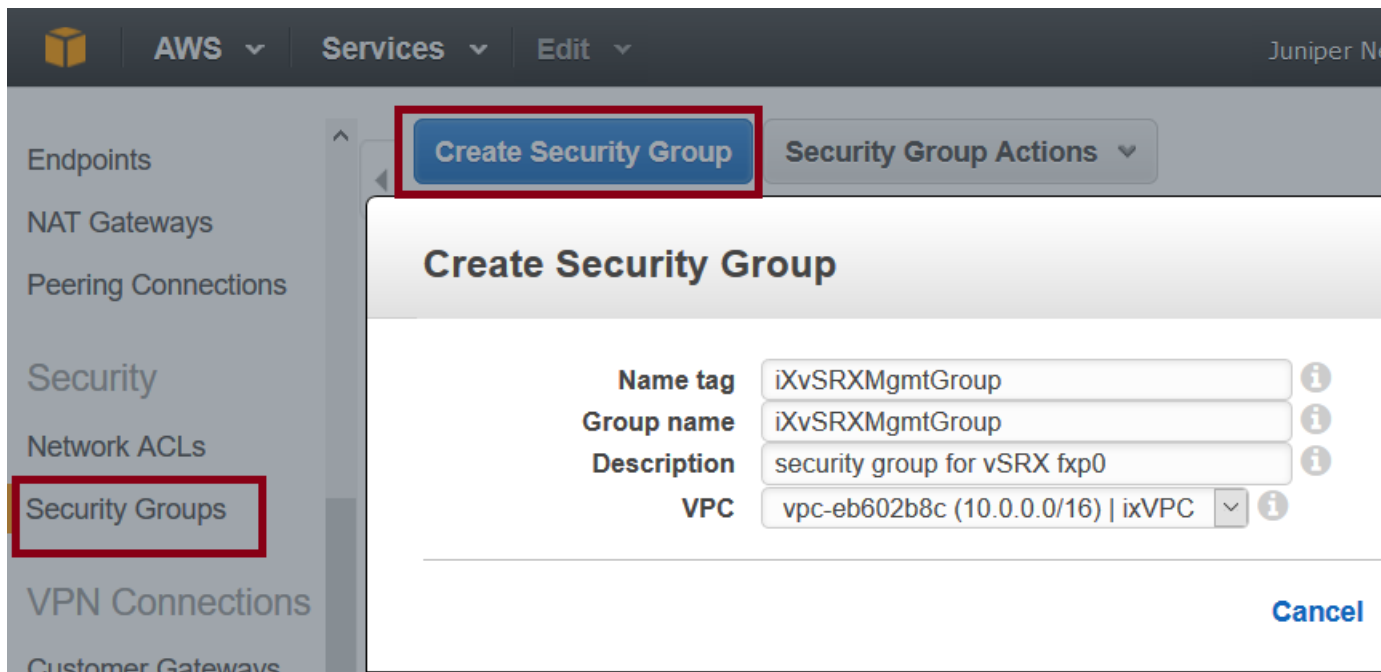
5. Select each route table you created for a private subnet and do the following:
 - a. Select the **Subnet Associations** tab, and click **Edit**.
 - b. Select the check box for one private subnet, and click **Save**.

Step 5: Add Security Groups for vSRX

A default security group is created for each Amazon VPC. We recommend that you create a separate security group for the vSRX management interface (fxp0) and another security group for all other vSRX interfaces. The security groups are assigned when a vSRX instance is launched in the Amazon EC2 Dashboard, where you can also add and manage security groups.

To create the security groups:

1. In the VPC Dashboard, select **Security Groups** in the left pane, and click **Create Security Group**.
2. For the vSRX management interface, specify a security group name in the Name Tag field, edit the Group Name field (optional), enter a description of the group, and select the VPC.
3. Click **Yes, Create**.



4. Repeat Steps "1" on page 43 through "3" on page 43 to create a security group for the vSRX revenue interfaces.
5. Select the security group you created for the management interface and do the following:
 - a. Select the **Inbound Rules** tab below the list of security groups.
 - b. Click **Edit** and click **Add another rule** to create the following inbound rules:

Type	Protocol	Port	Source
Custom TCP rule	Default	20-21	Enter CIDR address format for each rule (0.0.0.0/0 allows any source).
SSH (22)	Default	Default	
HTTP (80)	Default	Default	
HTTPS (443)	Default	Default	

- c. Click **Save**.

- d. Select the **Outbound Rules** tab to view the default rule that allows all outbound traffic. Use the default rule unless you need to restrict the outbound traffic.
6. Select the security group you created for all other vSRX interfaces and do the following:

NOTE: The inbound and outbound rules should allow all traffic to avoid conflicts with the security settings on vSRX.

- a. Select the **Inbound Rules** tab below the list of security groups.
- b. Click **Edit** and create the following inbound rule:

Type	Protocol	Port	Source
All Traffic	All	All	<ul style="list-style-type: none"> • For webservers, enter 0.0.0.0/0 • For VPNs, enter a range of IPv4 addresses in the form of a Classless Inter-Domain Routing (CIDR) block (for example, 10.0.0.0/16).

- c. Click **Save**.
- d. Keep the default rule in the **Outbound Rules** tab. The default rule allows all outbound traffic.

RELATED DOCUMENTATION

[Day One: Amazon Web Services with vSRX Cookbook](#)

[IAM Roles for Amazon EC2](#)

Launch a vSRX Instance on an Amazon Virtual Private Cloud

IN THIS SECTION

- [Step 1: Create an SSH Key Pair | 46](#)
- [Step 2: Launch a vSRX Instance | 48](#)
- [Step 3: View the AWS System Logs | 52](#)
- [Step 4: Add Network Interfaces for vSRX | 52](#)
- [Step 5: Allocate Elastic IP Addresses | 54](#)
- [Step 6: Add the vSRX Private Interfaces to the Route Tables | 54](#)
- [Step 7: Reboot the vSRX Instance | 55](#)
- [Step 8: Log in to a vSRX Instance | 56](#)

The following procedures describe how to launch and configure a vSRX instance in the Amazon Virtual Private Cloud (Amazon VPC):

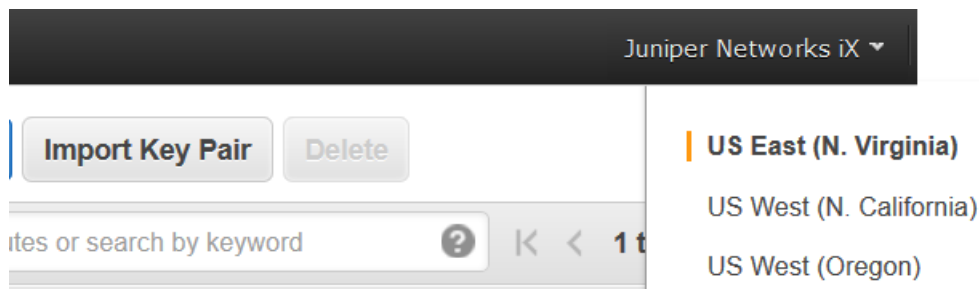
Step 1: Create an SSH Key Pair

An SSH key pair is required to remotely access a vSRX instance on AWS. You can create a new key pair in the Amazon EC2 Dashboard or import a key pair created by another tool.

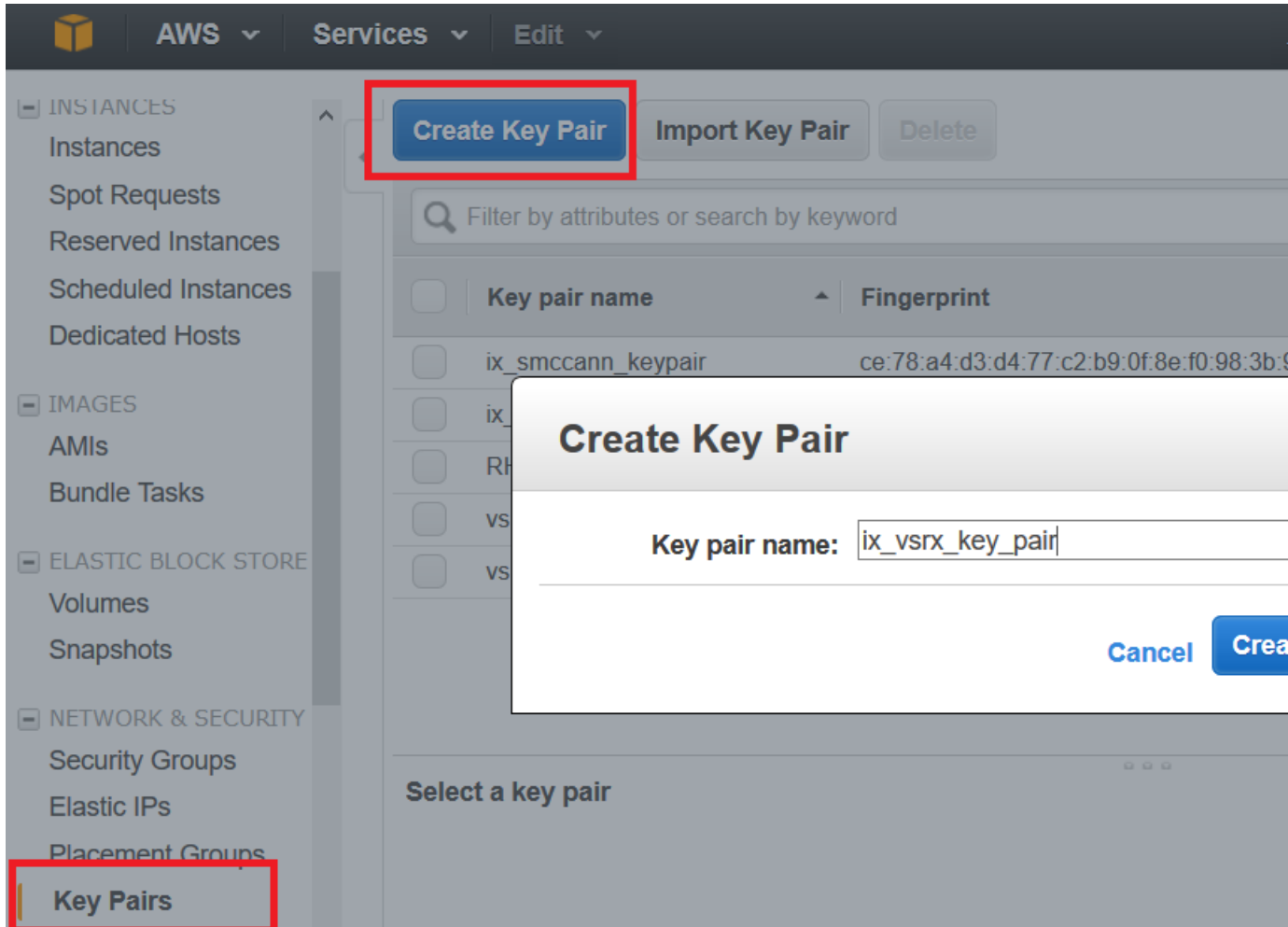
To create an SSH key pair:

1. Log in to the AWS Management Console and select **Services > Compute > EC2**.
2. In the Amazon EC2 Dashboard, select **Key Pairs** in the left pane. Verify that the region name shown in the toolbar is the same as the region where you created the Amazon Virtual Private Cloud (Amazon VPC).

Figure 5: Verify Region



3. Click **Create Key Pair**, specify a key pair name, and click **Create**.



4. The private key file (.pem) is automatically downloaded to your computer. Move the downloaded private key file to a secure location.
5. To use an SSH client on a Mac or Linux computer to connect to the vSRX instance, use the following command to set the permissions of the private key file so that only you can read it:

```
host# chmod 400 <key-pair-name>.pem
```

6. To access the vSRX instance from a shell prompt, use the `ssh -i <full path to your keyfile.pem>/<ssh-key-pair-name>.pem ec2-user@<public-ip-of-vsrx>` command. If the key file is in your current directory, then you can use the file name instead of the full path as `ssh -i <keyfile.pem>/<ssh-key-pair-name>.pem ec2-user@<public-ip-of-vsrx>`.

NOTE: Alternately, use **Import Key Pair** to import a different key pair you generated with a third-party tool.

For more information on key rotation, see <https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html>.

Step 2: Launch a vSRX Instance

The AWS instance types supported for vSRX are listed in [Table 8 on page 48](#).

vSRX does not support M and C3 instances types. If you have spun your vSRX using any of these instances types, then you must change the instance type to either C4 or C5 instances type.

Table 8: Supported AWS Instance Types for vSRX

Instance Type	vSRX Type	vCPUs	Memory (GB)
c4.xlarge	VSRX-4CPU-7G memory	4	7.5
c4.2xlarge	VSRX-8CPU-15G memory	8	15
c4.4xlarge	VSRX-16CPU-30G memory	16	30
c4.8xlarge	VSRX-36CPU-60G memory	36	60
c5.large	VSRX-2CPU-3G memory	2	4
c5.2xlarge	VSRX-8CPU-15G memory	8	16
c5.4xlarge	VSRX-16CPU-31G memory	16	32
c5n.2xlarge	VSRX-8CPU-20G memory	8	21
c5n.4xlarge	VSRX-16CPU-41G memory	16	42
c5n.9xlarge	VSRX-36CPU-93G memory	36	96

BEST PRACTICE: Instance Type Selection—Based on the changes that you require for your network, you might find that your instance is overutilized, (such as the instance type is too small) or underutilized, (such as the instance type is too large). If this is the case, you can change the size of your instance. For example, if your instance is too small for its workload, you can change it to another instance type that is appropriate for the workload. You might also want to migrate from a previous generation instance type to a current generation instance type to take advantage of some features; for example, support for IPv6. Consider change of instances for better performance and throughputs.

Starting with Junos OS Release 18.4R1, c5.large vSRX instances are supported. These are cost effective and provide better performance and throughput.

To launch a vSRX instance in the Amazon VPC:

1. In the Amazon EC2 Dashboard, select **Instances** in the left pane.
2. Click **Launch Instance**, search for the vSRX on AWS Marketplace, and click **Select** next to the vSRX AMI.
3. Select a supported instance type. See [Table 8 on page 48](#) for details.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tagging

Step 2: Choose an Instance Type

<input type="radio"/>	Compute optimized	c4.large	2	3.75
<input checked="" type="radio"/>	Compute optimized	c4.xlarge	4	7.5
<input type="radio"/>	Compute optimized	c4.2xlarge	8	15
<input type="radio"/>	Compute optimized	c4.4xlarge	16	30
<input type="radio"/>	Compute optimized	c4.8xlarge	36	60

Cancel Previous

4. Click **Next: Configure Instance Details**, and specify the fields in [Table 9 on page 50](#). Expand **Advanced Details** to see all settings.

Table 9: AWS Instance Details

Field	Setting
Network	Select the Amazon VPC configured for vSRX.
Subnet	Select the public subnet for the vSRX management interface (fxp0).
Auto-assign Public IP	Select Disable (you will assign an Elastic IP address later).
Placement group	Use the default.
Shutdown behavior	Select Stop (the default).
<ul style="list-style-type: none"> • Enable terminal protection • Monitoring 	Use your IT policy.
Network Interfaces	Use the default or assign a public IP address for the Primary IP field.
User data	<p>Starting in Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX for AWS image to help simplify configuring new vSRX instances operating on AWS according to a specified user-data file.</p> <p>In the User data section on the Configure Instance Details page, select As File and attach the user-data file. The selected file is used for the initial launch of the instance. During the initial boot-up sequence, the vSRX instance processes the cloud-init request. See "Using Cloud-Init to Automate the Initialization of vSRX Instances in AWS" on page 57 for information about how to create the user-data file.</p> <p>NOTE: The Junos OS configuration that is passed as user data is only imported at initial launch. If the instance is stopped and restarted, the user-data file is not imported again.</p>

5. Click **Next: Add Storage**, and use the default settings or change the Volume Type and IOPS as needed.
6. Click **Next: Tag Instance**, and specify a name for the vSRX instance.
7. Click **Next: Configure Security Group**, select **Select an existing security group**, and select the security group created for the vSRX management interface (fxp0).
8. Click **Review and Launch**, review the settings for the vSRX instance, and click **Launch**.

Step 7: Review Instance Launch

sg-874ddefd iXvSRXMgmtGroup security group for vSRX

All selected security groups inbound rules

Security Group ID	Type ⓘ	Protocol ⓘ	Port Range ⓘ
sg-	HTTP	TCP	80
sg-	SSH	TCP	22
sg-	Custom TCP Rule	TCP	20 - 21
sg-	HTTPS	TCP	443

▼ Instance Details

Number of instances	1	Purchasing option	On demand
Network	vpc-e		
Subnet	subnet-b78		
EBS-optimized	Yes		
Monitoring	No		
Termination protection	No		
Shutdown behavior	Stop		
IAM role	None		
Tenancy	default		
Host ID			
Affinity	Off		
Kernel ID	Use default		
RAM disk ID	Use default		
User data			
Assign Public IP	Use subnet setting (Disable)		

Network interfaces

Device	Network Interface	Subnet	Primary IP	Seconda
eth0	New network interface	subnet-b78	Auto-assign	

► Storage

► Tags

9. Select the SSH key pair you created, select the acknowledgment check box, and click **Launch Instance**.
10. Click **View Instances** to display the Instances list in the Amazon EC2 Dashboard. It might take several minutes to launch a vSRX instance.

Step 3: View the AWS System Logs

To debug launch time errors, you can view the AWS system logs, as follows:

1. In the Amazon EC2 Dashboard, select **Instances**.
2. Select the vSRX instance, and select **Actions > Instance Settings > Get System Logs**.

Step 4: Add Network Interfaces for vSRX

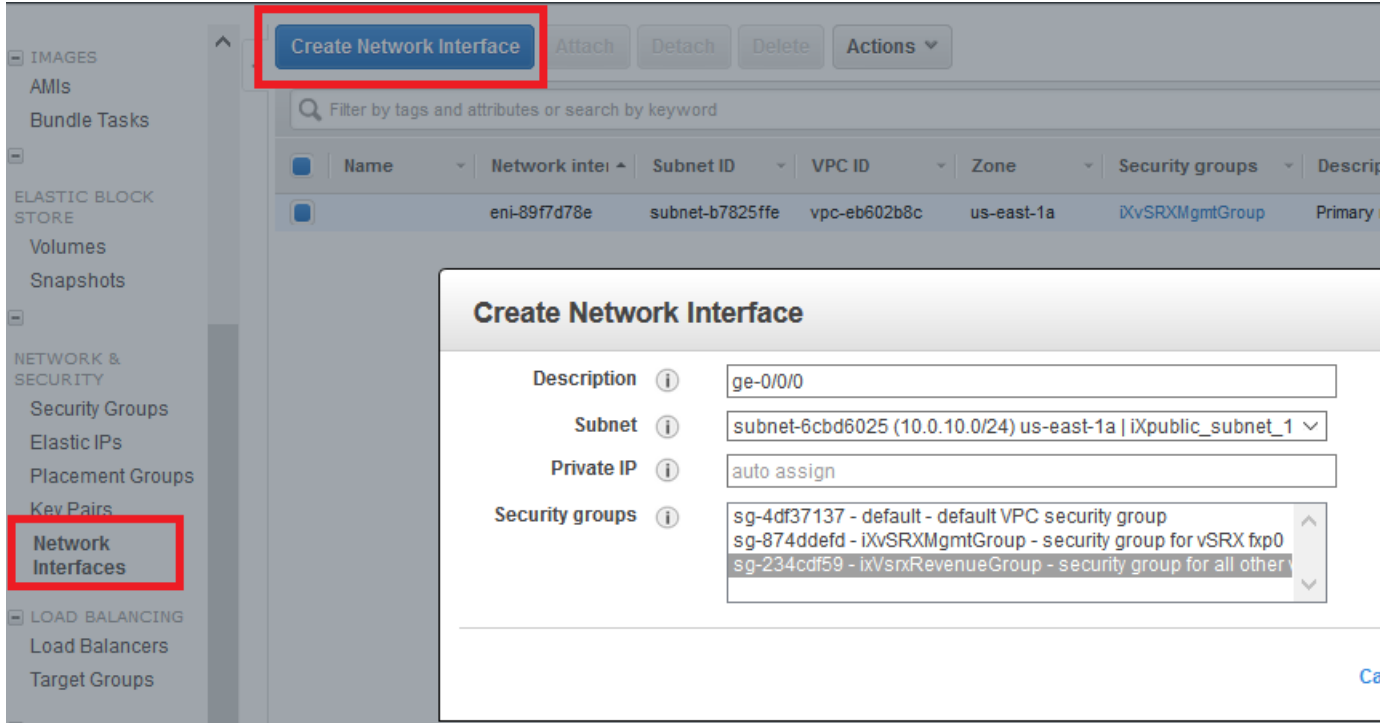
AWS supports up to eight interfaces for an instance, depending on the AWS instance type selected. Use the following procedure for each of the revenue interfaces you want to add to vSRX (up to seven). The first revenue interface is ge-0/0/0, the second is ge-0/0/1, and so on (see ["Requirements for vSRX on AWS" on page 12](#)).

To add a vSRX revenue interface:

1. In the Amazon EC2 Dashboard, select **Network Interfaces** in the left pane, and click **Create Network Interface**.
2. Specify the interface settings as shown in [Table 10 on page 52](#), and click **Yes, Create**.

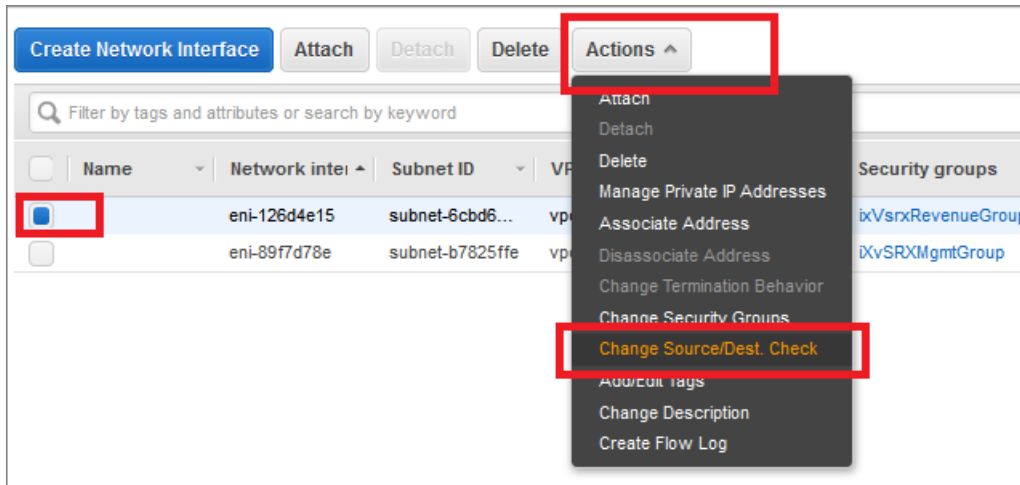
Table 10: Network Interface Settings

Field	Setting
Description	Enter an interface description for each of the revenue interfaces.
Subnet	Select the public subnet created for the first revenue interface (ge-0/0/0) or the private subnet created for all the other revenue interfaces.
Private IP	Enter an IP address from the selected subnet or allow the address to be assigned automatically.
Security Groups	Select the security group created for the vSRX revenue interfaces.



3. Select the new interface, select **Actions > Change Source/Dest. Check**, select **Disabled**, and click **Save**.

Figure 6: Disable Source/Dest. Check



4. Select the new interface, select **Attach**, select the vSRX instance, and click **Attach**.
5. Click the pencil icon in the new interface Name column and give the interface a name (for example, ix-fxp0.0).

NOTE: For a private revenue interface (ge-0/0/1 through ge-0/0/7), make a note of the network name you created or the network interface ID. You will add the name or interface ID later to the route table created for the private subnet.

Step 5: Allocate Elastic IP Addresses

For public interfaces, AWS does a NAT translation of the public IP address to a private IP address. The public IP address is called an *Elastic IP address*. We recommend that you assign an Elastic IP address to the public vSRX interfaces (fxp0 and ge-0/0/0). Note that when a vSRX instance is restarted, the Elastic IPs are retained, but public subnet IPs are released.

To create and allocate Elastic IPs:

1. In the Amazon EC2 Dashboard, select **Elastic IPs** in the left pane, click **Allocate New Address**, and click **Yes, Allocate**. (If your account supports EC2-Classic, you must first select **EC2-VPC** from the Network platform list.)
2. Select the new Elastic IP address, and select **Actions > Associate Address**.
3. Specify the settings in [Table 11 on page 54](#), and click **Allocate**.

Table 11: Elastic IP Settings

Field	Setting
Network Interface	Select the vSRX management interface (fxp0) or the first revenue interface (ge-0/0/0).
Private IP Address	Enter the private IP address to be associated with the Elastic IP address.

Step 6: Add the vSRX Private Interfaces to the Route Tables

For each private revenue interface you created for vSRX, you must add the interface ID to the route table you created for the associated private subnet.

To add a private interface ID to a route table:

1. In the VPC Dashboard, select **Route Tables** in the left pane.
2. Select the route table you created for the private subnet.
3. Select the **Routes** tab below the list of route tables.
4. Click **Edit** and click **Add another route**.
5. Specify the settings in [Table 12 on page 55](#), and click **Save**.

Table 12: Private Route Settings

Field	Setting
Destination	Enter 0.0.0.0/0 for Internet traffic.
Target	Type the network name or the network interface ID for the associated private subnet. The network interface must be in the private subnet shown in the Subnet Associations tab. NOTE: Do not select the Internet gateway (igw-nnnnnnnn).

Repeat this procedure for each private network interface. You must reboot the vSRX instance to complete this configuration.

Step 7: Reboot the vSRX Instance

To incorporate the interface changes and complete the Amazon EC2 configuration, you must reboot the vSRX instance. Interfaces attached while the vSRX instance is running do not take effect until the instance is rebooted.

NOTE: Always use AWS to reboot the vSRX instance. Do not use the vSRX CLI to reboot.

To reboot a vSRX instance:

1. In the Amazon EC2 Dashboard, select **Instances** in the left pane.
2. Select the vSRX instance, and select **Actions > Instance State > Reboot**.

It might take several minutes to reboot a vSRX instance.

Step 8: Log in to a vSRX Instance

In AWS deployments, vSRX instances provide the following capabilities by default to enhance security:

- Allows you to login only through SSH.
- cloud-init is used to setup SSH key login.
- SSH password login is disabled for root account.

vSRX instances launched on Amazon's AWS cloud infrastructure uses the cloud-init services provided by Amazon to copy the SSH public-key associated with your account that is used to launch the instance. You will then be able to login to the instance using the corresponding private-key.

NOTE: Root login using SSH password is be disabled by default.

Use an SSH client to log in to a vSRX instance for the first time. To log in, specify the location where you saved the SSH key pair **.pem** file for the user account, and the Elastic IP address assigned to the vSRX management interface (fxp0).

NOTE: Starting in Junos OS Release 17.4R1, the default user name has changed from **root@** to **ec2-user@**.

```
ssh -i <path>/<ssh-key-pair-name>.pem ec2-user@<fxpo-elastic-IP-address>
```

NOTE: Root login using a Junos OS password is disabled by default. You can configure other users after the initial Junos OS setup phase.

If you do not have the key pair filename and Elastic IP address, use these steps to view the key pair name and Elastic IP for a vSRX instance:

1. In the Amazon EC2 Dashboard, select **Instances**.
2. Select the vSRX instance, and select **eth0** in the Description tab to view the Elastic IP address for the fxp0 management interface.
3. Click **Connect** above the list of instances to view the SSH key pair filename.

To configure the basic settings for the vSRX instance, see ["Configure vSRX Using the CLI" on page 64](#).

NOTE: vSRX pay-as-you-go images do not require any separate licenses.

Release History Table

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, the cloud-init package (version 0.7x) comes pre-installed in the vSRX for AWS image to help simplify configuring new vSRX instances operating on AWS according to a specified user-data file.

RELATED DOCUMENTATION

| [Day One: Amazon Web Services with vSRX Cookbook](#)

Using Cloud-Init to Automate the Initialization of vSRX Instances in AWS

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

Cloud-init is an open source application for automating the initialization of a cloud instance at boot-up. Cloud-init is designed to support multiple different cloud environments, such as Amazon EC2, so that the same virtual machine (VM) image can be directly used in multiple cloud instances without any modification. Cloud-init support in a VM instance runs at boot time (first-time boot) and initializes the VM instance according to the specified user-data file.

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

When you create a vSRX instance, you can use **cloud-init** services on AWS to pass a valid Junos OS configuration file as user data to initialize new vSRX instances. The user-data file uses the standard Junos OS syntax to define all the configuration details for your vSRX instance. The default Junos OS

configuration is replaced during the vSRX instance launch with a validated Junos OS configuration that you supply in the form of a user-data file.

NOTE: The user-data file cannot exceed 16 KB. If your user-data file exceeds this limit, you must compress the file using gzip and use the compressed file. For example, the gzip junos.conf command results in the junos.conf.gz file.

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



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

To initiate the automatic setup of a vSRX instance from AWS:

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

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

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

2. Copy the Junos OS configuration file to an accessible location from where it can be retrieved to launch the vSRX instance.
3. To specify the user-data file for configuring the vSRX instance, select **As File** in the User data section on the Configure Instance Details page and attach the file (as described in ["Launch a vSRX Instance on an Amazon Virtual Private Cloud" on page 45](#)). The selected configuration file is used for the initial launch of the vSRX instance. During the initial boot-up sequence, the vSRX instance processes the cloud-init request.

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

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

Release History Table

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

RELATED DOCUMENTATION

[Cloud-Init Documentation](#)

[cloud-init](#)

[Launching an Instance](#)

Upgrade Junos OS Software on a vSRX Instance

IN THIS SECTION

- [Upgrade the Junos OS for vSRX Software Release](#) | 60
- [Replace the vSRX Instance on AWS](#) | 60

This section outlines how to upgrade Junos OS software on your vSRX instance to a newer release. Depending upon your preference, you can replace the vSRX software in one of two ways:

Upgrade the Junos OS for vSRX Software Release

You can directly upgrade the Junos OS for vSRX software using the CLI. Upgrading or downgrading Junos OS can take several hours, depending on the size and configuration of the network. You download the desired Junos OS Release for vSRX .tgz file from the [Juniper Networks website](#).

You also can upgrade using J-Web (see [J-Web](#)) or the Junos Space Network Management Platform (see [Junos Space](#)).

For the procedure on upgrading a specific Junos OS for vSRX software release, see the *Migration, Upgrade, and Downgrade Instructions* topic in the release-specific *vSRX Release Notes* available on the [vSRX TechLibrary](#).

Replace the vSRX Instance on AWS

To replace a vSRX instance on AWS with a different software release:

1. Log in to the vSRX instance using SSH and start the CLI.

NOTE: Starting in Junos OS Release 17.4R1, the default user name has changed from **root@** to **ec2-user@**.

```
ec2-user@% cli
ec2-user@>
```

2. Enter configuration mode.

```
ec2-user@> configure
[edit]
ec2-user@#
```

3. Copy the existing Junos OS configuration from the vSRX. The contents of the current level of the statement hierarchy (and below) are saved, along with the statement hierarchy containing it.

NOTE: By default, the configuration is saved to a file in your home directory.

- See [Saving a Configuration File](#) for additional background information on saving a Junos OS configuration file.
- See [file copy](#) for information on how to copy files from one location to another location on the local device or to a location on a remote device that is reachable by the local device.

```
ec2-user@#save <filename>
[edit]
ec2-user@#
```

4. Remove the vSRX instance on AWS as described in ["Remove a vSRX Instance on AWS" on page 61](#).
5. Once the vSRX instance on AWS has been successfully removed, define the specifics of a vSRX instance prior to launching it. See ["Configure an Amazon Virtual Private Cloud for vSRX" on page 31](#).
6. Launch the vSRX image using the desired software version available from AWS Marketplace as described in ["Launch a vSRX Instance on an Amazon Virtual Private Cloud" on page 45](#)
7. Load the previously copied Junos OS configuration file onto your new (upgraded) vSRX instance as described in [Loading a Configuration File](#).

Remove a vSRX Instance on AWS

To remove a vSRX instance on AWS:

1. Log in to the AWS Management Console and select **Services > Compute > EC2 > Instances**.
2. Select the vSRX instance and select **Actions > Instance State > Terminate** to remove the instance.
3. In the dialog box, expand the section and select **Release associated Elastic IP**.
4. Click **Yes, Terminate**.

NOTE: See [Deleting Your VPC](#) to remove any unused VPCs from AWS.

3

CHAPTER

Configuring and Managing vSRX

[vSRX Configuration and Management Tools | 63](#)

[Configure vSRX Using the CLI | 64](#)

[Configure vSRX Using the J-Web Interface | 69](#)

[Managing Security Policies for Virtual Machines Using Junos Space Security Director | 72](#)

[AWS Elastic Load Balancing and Elastic Network Adapter | 73](#)

[Software Receive Side Scaling | 91](#)

[Multi-Core Scaling Support on AWS with SWRSS and ENA | 93](#)

[GTP Traffic with TEID Distribution and SWRSS | 94](#)

[Centralized Monitoring and Troubleshooting using AWS Features | 98](#)

[Deploying vSRX 3.0 for Securing Data using AWS KMS | 108](#)

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 | 63](#)
- [Understanding the J-Web Interface | 63](#)
- [Understanding Junos Space Security Director | 63](#)

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

IN THIS SECTION

- [Understand vSRX on AWS Preconfiguration and Factory Defaults | 64](#)
- [Add a Basic vSRX Configuration | 65](#)
- [Add DNS Servers | 68](#)
- [Add vSRX Feature Licenses | 68](#)

Understand vSRX on AWS Preconfiguration and Factory Defaults

vSRX on AWS deploys with the following preconfiguration defaults:

- SSH access with the RSA key pair configured during the installation
- No password access allowed for SSH access
- The management (fxp0) interface is preconfigured with the AWS Elastic IP and default route

Starting in Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, the following example summarizes the preconfiguration statements added to a factory-default configuration for vSRX on AWS instances:

```
set groups aws-default system root-authentication ssh-rsa "ssh-rsa XXXRSA-KEYXXXXX"
set groups aws-default system services ssh no-passwords
set groups aws-default system services netconf ssh
set groups aws-default system services web-management https system-generated-certificate
set groups aws-default interfaces fxp0 unit 0 family inet address aws-ip-address
set groups aws-default routing-options static route 0.0.0.0/0 next-hop aws-ip-address
set apply-groups aws-default
```

For Junos OS Release 15.1X49-D70 and earlier, the following example summarizes the preconfiguration statements added to a factory-default configuration for vSRX on AWS instances:

```
set system root-authentication ssh-rsa "ssh-rsa XXXRSA-KEYXXXXX"
set system services ssh no-passwords
set interfaces fxp0 unit 0 family inet address aws-ip-address
set routing-options static route 0.0.0.0/0 next-hop aws-ip-address
```



CAUTION: Do not use the **load factory-default** command on a vSRX AWS instance. The factory default configuration removes the AWS preconfiguration. If you must revert to factory default, ensure that you manually reconfigure AWS preconfiguration statements before you commit the configuration; otherwise, you will lose access to the vSRX instance.

Add a Basic vSRX Configuration

You can either create a new configuration on vSRX or copy an existing configuration from another SRX or vSRX and load it onto your vSRX on AWS. Use the following steps to copy and load an existing configuration:

1. [Saving a Configuration File](#)
2. [Loading a Configuration File](#)

To configure a vSRX instance using the CLI:

1. Log in to the vSRX instance using SSH and start the CLI.

NOTE: Starting in Junos OS Release 17.4R1, the default user name has changed from **root@** to **ec2-user@**.

```
ec2-user@% cli
ec2-user@>
```

2. Enter configuration mode.

```
ec2-user@> configure
[edit]
ec2-user@#
```

3. Set the authentication method to log into the vSRX. You can specify a password by entering a cleartext password or an encrypted password. If you require a more robust level of authentication security, we recommend that you select an SSH public key string (DSA, ECDSA, or RSA).

```
ec2-user@# set system root-authentication ssh-rsa <public-key>
```

or

```
ec2-user@# set system root-authentication plain-text-password
New password: password
Retype new password: password
```

4. Optionally, enable passwords for SSH if you want to create password access for additional users.

```
ec2-user@# delete services ssh no-passwords
```

5. Configure the hostname.

```
ec2-user@# set system host-name host-name
```

- For each vSRX revenue interface, assign the IP address defined on AWS. For example:

```
ec2-user@# set interfaces ge-0/0/0 unit 0 family inet address 10.0.10.197/24
```

For multiple private addresses, enter a **set** command for each address. Do not assign the Elastic IP address.

- Specify a security zone for the public interface.

```
ec2-user@# set security zones security-zone untrust interfaces ge-0/0/0.0
```

- Specify a security zone for the private interface.

```
ec2-user@# set security security-zone trust interfaces ge-0/0/1.0
```

- Configure routing to add a separate virtual router and routing option for the public and private interfaces.

NOTE: We recommend putting the revenue (data) interfaces in routing instances as a best practice to avoid asymmetric traffic/routing, because fxp0 is part of the default (inet.0) table by default. With fxp0 as part of the default routing table, there might be two default routes needed: one for the fxp0 interface for external management access, and the other for the revenue interfaces for traffic access. Putting the revenue interfaces in a separate routing instance avoids this situation of two default routes in a single routing instance.

```
set routing-instances aws instance-type virtual-router
set routing-instances aws interface ge-0/0/0.0
set routing-instances aws interface ge-0/0/1.0
set routing-instances aws interface st0.1
set routing-instances aws routing-options static route 0.0.0.0/0 next-hop 10.0.0.1
set routing-instances aws routing-options static route 10.20.20.0/24 next-hop st0.1
```

- Verify the configuration.

```
ec2-user@# commit check
configuration check succeeds
```

11. Commit the configuration to activate it on the device.

```
ec2-user@# commit
commit complete
```

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

For an example of how to configure vSRX to NAT all hosts behind the vSRX instance in the Amazon Virtual Private Cloud (Amazon VPC) to the IP address of the vSRX egress interface on the untrust zone, see ["Example: Configuring NAT for vSRX" on page 119](#). This configuration allows hosts behind vSRX in a cloud network to access the Internet.

For an example of how to configure IPsec VPN between two instances of vSRX on AWS on different Amazon VPCs, see ["Example: Configure VPN on vSRX Between Amazon VPCs" on page 121](#).

Add DNS Servers

vSRX does not include any DNS servers in the default configuration. You might need DNS configured to deploy Layer 7 services, such as IPS, to pull down signature updates, for example. You can use your own external DNS server or use an AWS DNS server. If you enable DNS on your Amazon VPC, queries to the Amazon DNS server (169.254.169.253) or the reserved IP address at the base of the VPC network range plus two should succeed. See [AWS - Using DNS with Your Amazon VPC](#) for complete details.

Add vSRX Feature Licenses

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

See [Managing Licenses for vSRX](#) for details.

RELATED DOCUMENTATION

[CLI User Guide](#)

[AWS - Using DNS with Your VPC](#)

Configure vSRX Using the J-Web Interface

IN THIS SECTION

- Access the J-Web Interface and Configure vSRX | 69
- Apply the Configuration Settings for vSRX | 71
- Add vSRX Feature Licenses | 72

Access the J-Web Interface and Configure vSRX

To configure vSRX using the *J-Web* Interface:

1. Enter the AWS Elastic IP address of the eth0 interface in the browser Address box.
2. Specify the username and password.
3. Click **Log In**, and select the **Configuration Wizards** tab from the left navigation panel. The J-Web Setup Wizard page opens.
4. Click **Setup**.

You can use the Setup wizard to configure a device or edit an existing configuration.

- Select **Edit Existing Configuration** if you have already configured the wizard using the factory mode.
- Select **Create New Configuration** to configure a device using the wizard.

The following configuration options are available in the guided setup:

- Basic

Select **basic** to configure the device name and user account information as shown in [Table 13 on page 70](#).

- Device name and user account information

Table 13: Device Name and User Account Information

Field	Description
Device name	Type the name of the device. For example: vSRX .
Root password	Create a default root user password.
Verify password	Verify the default root user password.
Operator	<p>Add an optional administrative account in addition to the root account.</p> <p>User role options include:</p> <ul style="list-style-type: none"> • Superuser: This user has full system administration rights and can add, modify, and delete settings and users. • Operator: This user can perform system operations such as a system reset but cannot change the configuration or add or modify users. • Read only: This user can only access the system and view the configuration. • Disabled: This user cannot access the system.

- Select either **Time Server** or **Manual**. [Table 14 on page 70](#) lists the system time options.

Table 14: System Time Options

Field	Description
Time Server	
Host Name	Type the hostname of the time server. For example: ntp.example.com .

Table 14: System Time Options (*Continued*)

Field	Description
IP	Type the IP address of the time server in the IP address entry field. For example: 192.168.1.254 .
NOTE: You can enter either the hostname or the IP address.	
Manual	
Date	Click the current date in the calendar.
Time	Set the hour, minute, and seconds. Choose AM or PM .
Time Zone (mandatory)	
Time Zone	Select the time zone from the list. For example: GMT Greenwich Mean Time GMT.

- Expert
 - a. Select **Expert** to configure the basic options as well as the following advanced options:
 - Four or more internal zones
 - Internal zone services
 - Application of security policies between internal zones
 - b. Click **Need Help** for detailed configuration information.

You see a success message after the basic configuration is complete.

Apply the Configuration Settings for vSRX

To apply the configuration settings for vSRX:

1. Review and ensure that the configuration settings are correct, and click **Next**. The Commit Configuration page appears.
2. Click **Apply Settings** to apply the configuration changes to vSRX.
3. Check the connectivity to vSRX, because you might lose connectivity if you have changed the management zone IP. Click the URL for reconnection instructions on how to reconnect to the device.
4. Click **Done** to complete the setup.

After successful completion of the setup, you are redirected to the J-Web interface.



CAUTION: After you complete the initial setup, you can relaunch the J-Web Setup wizard by clicking **Configuration>Setup**. You can either edit an existing configuration or create a new configuration. If you create a new configuration, the current configuration in vSRX will be deleted.

Add vSRX Feature Licenses

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

See [Managing Licenses for vSRX](#) for details.

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](#)

AWS Elastic Load Balancing and Elastic Network Adapter

IN THIS SECTION

- [Overview of AWS Elastic Load Balancing | 74](#)
- [Overview of Application Load Balancer | 75](#)
- [Deployment of AWS Application Load Balancer | 77](#)
- [Invoking Cloud Formation Template \(CFT\) Stack Creation for vSRX Behind AWS Application Load Balancer Deployment | 81](#)
- [Overview of AWS Elastic Network Adapter \(ENA\) for vSRX Instances | 90](#)

This section provides an overview of the AWS ELB and ENA features and also describes how these features are deployed on vSRX instances.

Overview of AWS Elastic Load Balancing

IN THIS SECTION

- Benefits of AWS Elastic Load Balancing | 74
- AWS Elastic Load Balancing Components | 75

This section provides information about AWS ELB.

Elastic Load Balancing (ELB) is a load-balancing service for Amazon Web Services (AWS) deployments.

ELB distributes incoming application or network traffic across ntra availability zones, such as Amazon EC2 instances, containers, and IP addresses. ELB scales your load balancer as traffic to your application changes over time, and can scale to the vast majority of workloads automatically.

AWS ELB using application load balancers enables automation by using certain AWS services:

- **Amazon Simple Notification Service**—For more information, see <https://docs.aws.amazon.com/sns/latest/dg/welcome.html>.
- **AWS Lambda**—For more information, see <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>.
- **AWS Auto Scale Group**—For more information, see <https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html>.

Benefits of AWS Elastic Load Balancing

- Ensures elastic load balancing for intra available zone by automatically distributing the incoming traffic.
- Provides flexibility to virtualize your application targets by allowing you to host more applications on the same instance and to centrally manage Transport Layer Security (TLS) settings and offload CPU-intensive workloads from your applications.
- Provides robust security features such as integrated certificate management, user authentication, and SSL/TLS decryption.

- Supports auto-scaling a sufficient number of applications to meet varying levels of application load without requiring manual intervention.
- Enables you to monitor your applications and their performance in real time with Amazon CloudWatch metrics, logging, and request tracing.
- Offers load balancing across AWS and on-premises resources using the same load balancer.

AWS Elastic Load Balancing Components

AWS Elastic Load Balancing (ELB) components include:

- **Load balancers**—A load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple availability zones (AZs), thereby increasing the availability of your application. You add one or more listeners to your load balancer.
- **Listeners or vSRX instances**—A listener is a process for checking connection requests, using the protocol and port that you configure. vSRX instances as listeners check for connection requests from clients, using the protocol and port that you configure, and forward requests to one or more target groups, based on the rules that you define. Each rule specifies a target group, condition, and priority. When the condition is met, the traffic is forwarded to the target group. You must define a default rule for each vSRX instance, and you can add rules that specify different target groups based on the content of the request (also known as content-based routing).
- **Target groups or vSRX application workloads**—Each vSRX application as target group is used to route requests to one or more registered targets. When you create each vSRX instance as a listener rule, you specify a vSRX application and conditions. When a rule condition is met, traffic is forwarded to the corresponding vSRX application. You can create different vSRX applications for different types of requests. For example, create one vSRX application for general requests and other vSRX applications for requests to the microservices for your application.

AWS ELB supports three types of load balancers: application load balancers, network load balancers, and classic load balancers. You can select a load balancer based on your application needs. For more information about the types of AWS ELB load balancers, see [AWS Elastic Load Balancing](#).

Overview of Application Load Balancer

Starting in Junos OS Release 18.4R1, vSRX instances support AWS Elastic Load Balancing (ELB) using the application load balancer to provide scalable security to the Internet-facing traffic using native AWS services. An application load balancer automatically distributes incoming application traffic and scales resources to meet traffic demands.

You can also configure health checks to monitor the health of the registered targets so that the load balancer can send requests only to the healthy targets.

The key features of an application load balancer are:

- Layer-7 load balancing
- HTTPS support
- High availability
- Security features
- Containerized application support
- HTTP/2 support
- WebSockets support
- Native IPv6 support
- Sticky sessions
- Health checks with operational monitoring, logging, request tracing
- Web Application Firewall (WAF)

When the application load balancer receives a request, it evaluates the rules of the vSRX instance in order of priority to determine which rule to apply, and then selects a target from the vSRX application for the rule action. You can configure a vSRX instance rule to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.

You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application. ELB scales your load balancer as traffic to your application changes over time. ELB can scale majority of workloads automatically.

The application load balancer launch sequence and current screen can be viewed using the vSRX instance properties. When running vSRX as an AWS instance, logging in to the instance through SSH starts a session on Junos OS. Standard Junos OS CLI can be used to monitor health and statistics of the vSRX instance. If the `#load_balancer=true` tag is sent in user data, then boot-up messages mention that the vSRX interfaces are configured for ELB and auto-scaling support. Interfaces `eth0` and `eth1` are then swapped.

If an unsupported Junos OS configuration is sent to the vSRX instance in user data, then the vSRX instance reverts to its factory-default configuration. If the `#load_balancer=true` tag is missing, then interfaces are not swapped.

Deployment of AWS Application Load Balancer

IN THIS SECTION

- [vSRX Behind AWS ELB Application Load Balancer Deployment | 77](#)
- [Sandwich Deployment of AWS ELB Application Load Balancer | 79](#)

AWS ELB application load balancer can be deployed in two ways:

- vSRX behind AWS ELB application load balancer
- ELB sandwich

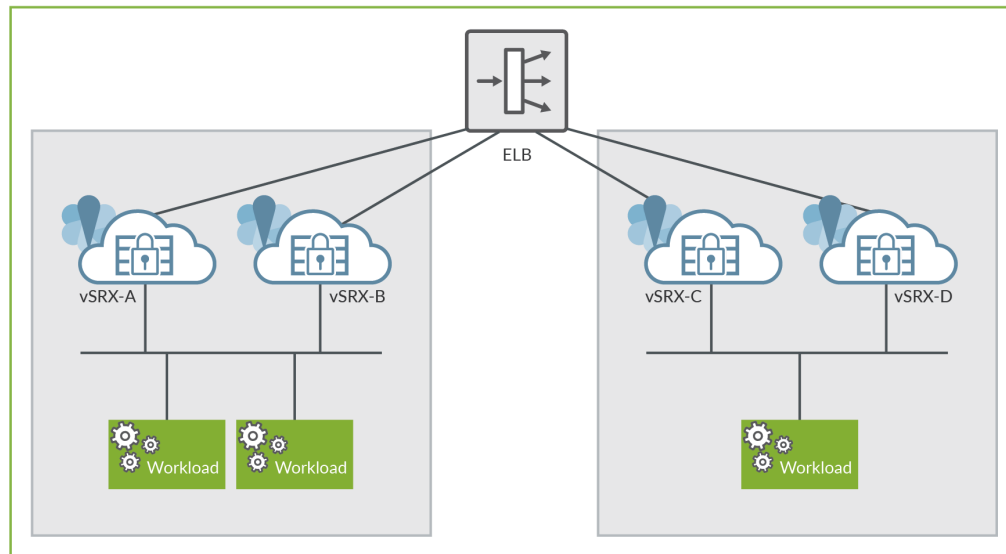
vSRX Behind AWS ELB Application Load Balancer Deployment

In this type of deployment, the vSRX instances are attached to the application load balancer, in one or more availability zones (AZs), and the application workloads are behind the vSRX instances. The application load balancer sends traffic only to the primary interface of the instance. For a vSRX instance, the primary interface is the management interface fxp0.

To enable ELB in this deployment, you have to swap the management and the first revenue interface.

Figure 7 on page 78 illustrates the vSRX behind AWS ELB application load balancer deployment.

Figure 7: vSRX Behind AWS ELB Application Load Balancer Deployment



Enabling AWS ELB with vSRX Behind AWS ELB Application Load Balancer Deployment

The following are the prerequisites for enabling AWS ELB with the vSRX behind AWS ELB application load balancer type of deployment:

- All incoming and outgoing traffic to ELB are monitored from the ge-0/0/0 interface associated with the vSRX instance.
- The vSRX instance at launch has two interfaces in which the subnets containing the interfaces are connected to the internet gateway (IGW). The two interface limit is set by the AWS auto scaling group deployment. You need to define at least one interface in the same subnet as the AWS ELB. The additional interfaces can be attached by the lambda function.
- Source or destination check is disabled on the eth1 interface of the vSRX instance.

For deploying an AWS ELB application load balancer using the vSRX behind AWS ELB application load balancer method:

The vSRX instance contains:

- Cloud initialization (cloud-init) user data with ELB tag as #load_balancer=true.

- The user data configuration with #junos-config tag, fxp0 (dhcp), ge-0/0/0 (dhcp) (must be DHCP any security group that it needs to define)
- Cloud-Watch triggers an Simple Notification Service (SNS), which in turn triggers a Lambda function that creates and attaches an Elastic Network Interface (ENI) with Elastic IP address (EIP) to the vSRX instance. Multiple new ENIs (maximum of 8) can be attached to this instance.
- The vSRX Instance must be rebooted. A reboot must be performed for all subsequent times the vSRX instance launches with swapped interfaces.

NOTE: Chassis cluster is not supported if you try to swap the ENI between instances and IP monitoring.

NOTE: You can also launch the vSRX instance in an Auto Scaling Group (ASG). This launch can be automated using a cloud formation template (CFT).

Sandwich Deployment of AWS ELB Application Load Balancer

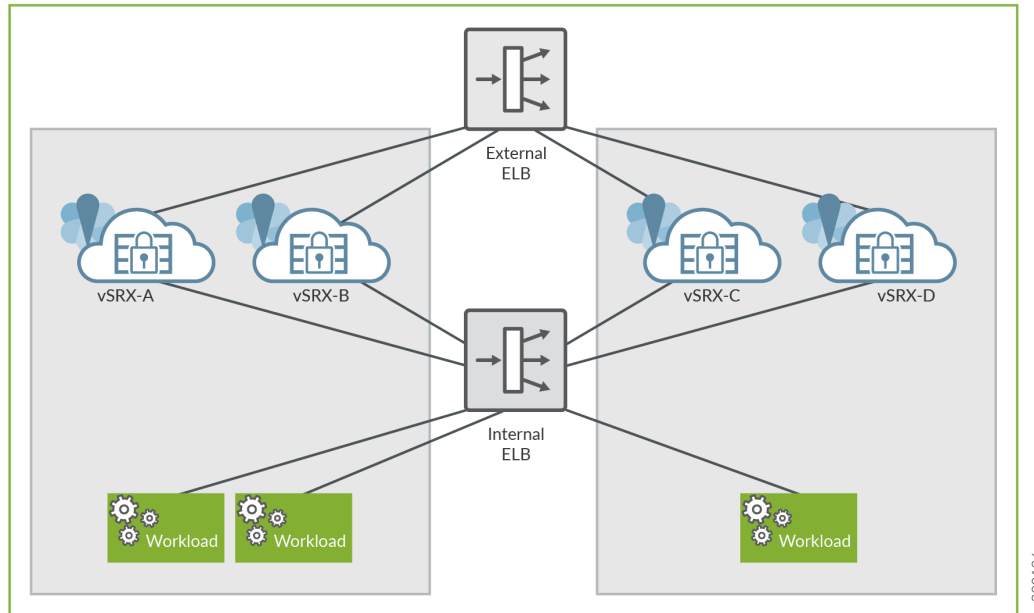
In this deployment model, you can scale both, security and applications. vSRX instances and the applications are in different ASGs and each of these ASGs is attached to a different application load balancer. This type of ELB deployment is elegant and simplified way to manually scale vSRX deployments to address planned or projected traffic increases while also delivering multi-AZ high availability. The deployment ensures inbound high availability and scaling for AWS deployments.

Because the load balancer scales dynamically, its virtual IP address (VIP) is a fully qualified domain name (FQDN). This FQDN resolves to multiple IP addresses according to the availability zone. To enable this resolution, the vSRX instance should be able to send and receive traffic from the FQDN (or the multiple addresses that it resolves to).

You configure this FQDN by using the **set security zones security-zone ELB-TRAFFIC address-book address ELB dns-name FQDN_OF_ELB** command.

Figure 8 on page 80 illustrates the AWS ELB application load balancer sandwich deployment for vSRX.

Figure 8: Sandwich Deployment of AWS ELB Application Load Balancer



Enabling Sandwich Deployment of AWS Application Load Balancer for vSRX

For AWS ELB application load balancer sandwich deployment for vSRX:

- vSRX receives the `#load_balancer=true` tag in cloud-init user data.
- In Junos OS, the initial boot process scans the mounted disk for the presence of the flag file in the `setup_vsrx` file. If the file is present, it indicates that the two interfaces with DHCP in two different virtual references must be configured. This scan and configuration update is performed in the default configuration and on top of the user data if the flag file is present.

NOTE: If user data is present, then the boot time after the second or the third mgd process commit increases.

- You must reboot the vSRX instance. Perform reboot for all the subsequent times the vSRX instance is launched with swapped interfaces.

NOTE: Chassis cluster support for swapping the Elastic Network Interfaces (ENIs) between instances and IP monitoring does not work.

NOTE: You can also launch vSRX instance in an ASG and automate the deployment using a cloud formation template (CFT).

Invoking Cloud Formation Template (CFT) Stack Creation for vSRX Behind AWS Application Load Balancer Deployment

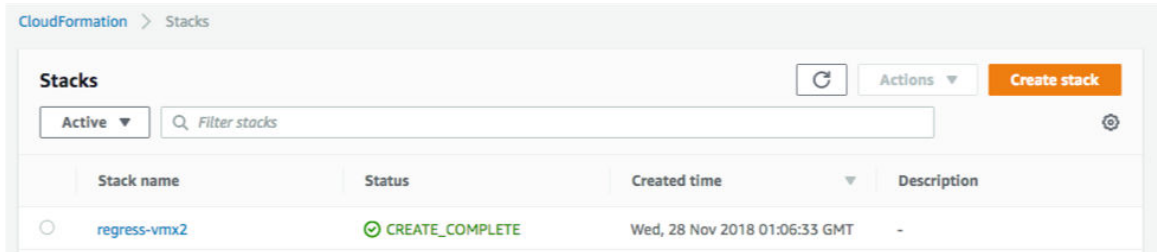
This topic provide details on how to invoke cloud formation template (CFT) stack creation for the non-sandwich deployment (with vSRX Behind AWS Application Load Balancer) which contains only one load balancer.

Before you invoke the CFT stack creation, ensure you have the following already available within AWS environment:

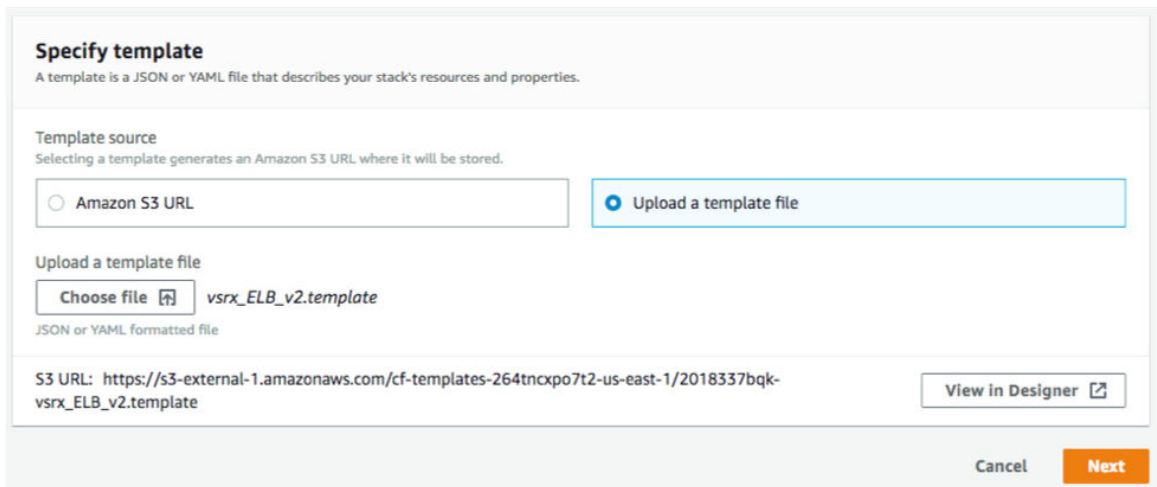
- VPC created and ready to use.
- A management subnet
- An external subnet (subnet for vSRX interface receiving traffic from the ELB).
- An internal subnet (subnet for vSRX interface sending traffic to the workload).
- An AMI ID of the vSRX instance that you want to launch.
- User data (the vSRX configuration that has to be committed before the traffic is forwarded to the workload. This is a base 64 encoded data not more than 4096 characters in length; you may use up to three user data fields if a single field data exceeds 4096 characters).
- EC2 key file.
- Get the lambda function file add_eni.zip from Juniper vSRX GitHub repository and upload it to your instances S3 bucket. Use this information in the **Lambda S3 Location** field of the template.
- Your AWS account should have permissions to create Lambda functions on various resources in your region.

Follow the following steps to invoke CFT stack creation for AWS ELB with vSRX behind AWS ELB application load balancer deployment.

1. Log into your AWS account and make sure the region on the top right is the one you want to use. Go to AWS console home page and under **All Services** look for **Management & Governance** section and click **CloudFormation** option.
2. Click the **Create Stack** button on the top right side of the CloudFormation page.



3. On the new page, select **Upload a template file radio** button, then click **Choose file** button, and then select your template file and click **Next**.



4. The next page that opens is a form created from the template. Some fields might already have a default value, that you might change if you want to. Enter a **Stack Name**, select the **VPC ID**, **InstanceType**, **MgtSubnetID**, **ExternalSubnetID**, **InternalSubnetID**, **ImageID**. Paste the Base64 encoded user data (which is the vSRX configuration to be committed and is provided in a separate text file). If your Base64 encoded vSRX configuration exceeds 4096 bytes, you may use **UserData2** and **UserData3** fields as needed.
5. Set **MinASGInstances** as 1 and **MaxASGInstances** as 3
6. Select your Amazon EC2 Key Pair file and click **Next**.

Create stack

Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

vSRX configuration

VPCID

VPC ID

InstanceType

Select the instance type you want to use

MgtSubnetID

Subnet ID used for the mgmt interface

ExternalSubnetID

Subnet ID used for the revenue interface

InternalSubnetID

Subnet ID used for the internal interface

ImageID

UserData

User Data configuration. Maximum 4096 bytes. For better readability, encode it with base64. Final userdata will be the combination of UserData, UserData2 and UserData3

UserData2

User Data configuration (Optional). Maximum 4096 bytes.

UserData3

User Data configuration (Optional). Maximum 4096 bytes.

Auto scaling group configuration

MinASGInstances

Minimum number of vSRX in the auto scaling group

MaxASGInstances

Maximum number of vSRX in the auto scaling group

Other parameters

EC2KeyPair

Amazon EC2 Key Pair

Cancel

Previous

Next

- 7. Skip the next page with **Configure stack** options and **Advanced** option and click **Next**.

Configure stack options

Tags
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more.](#)

Key Value Remove

Add tag

Permissions
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more.](#)

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name Sample-role-name Remove

Advanced options

- ▶ **Stack policy**
Defines the resources that you want to protect from unintentional updates during a stack update.
- ▶ **Rollback configuration**
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more.](#)
- ▶ **Notification options**
You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more.](#)
- ▶ **Stack creation options**

Cancel Previous Next

- 8. On the next page, you will be able to review and edit your stack creation details. Once you are done reviewing, click **Create stack** button on the bottom right of the page.

Stack creation options

Rollback on failure
Enabled

Timeout
-

Termination protection
Disabled

▶ Quick-create link

Cancel
Previous
Create change set
Create stack

9. On the next page, wait for the stack creation to be completed. If there are any errors in the stack creation, then the errors are displayed on this page. You have to rectify the errors and recreate the stack using the above steps.
10. Once the stack is created successfully , click **Services>EC2** and then click **Auto Scaling Groups** on the left-hand side menu.

On the right-hand side of the page, you should see an auto-scaling group (ASG) with the stack name that you created.

When you select the ASG you created then that ASG details are displayed at the bottom of the page.

Click the **Scaling Policies** tab to create a scaling policy for this ASG, to maintain a certain number of vSRXs in the ASG and to cater to various requests, as per your requirements. Refer to 'Scaling policy example' under the 'Sample Data' in this topic below.

Auto Scaling Group monitors the state of the vSRX instances. It will automatically re spawn a new instance if any vSRX instance failure is detected. You can find more information in the **Activity History** tab of the ASG and in the Cloudwatch logs.

Auto Scaling Group: vSRX3-ELB-CFT-01-vSRXASG-1BLW7XFQ4XVJL

Details
Activity History
Scaling Policies
Instances
Monitoring
Notifications
Tags
Scheduled Actions
Lifecycle Hooks

Add policy

vSRX3_ELB_SP01

Policy type: Target Tracking scaling

Execute policy when: As required to maintain Application Load Balancer Request Count Per Target at 5000

Take the action: Add or remove instances as required

Instances need: 15 seconds to warm up after scaling

Disable scale-in: No

- Click **Services**>**EC2** and then **Load Balancers** on the left-hand side menu. On the right-hand side of the page, you should see a load balancer (LB) with the stack name that you created. You can select this load balancer and view the load balancer details at the bottom of the page.

The **instances** tab above will show the vSRX instances being load-balanced by this LB. This LB will be assigned a DNS name as show above. Any HTTP traffic sent to that host will be forwarded by the vSRX to the web server workload being protected by the vSRX. The number of vSRX instances can vary between **MinASGInstances** and **MaxASGInstances** used during setup, depending upon the scaling criteria.

Load balancer: **sichao-v3-External-162OH8ELJZL71**

Description Instances Health check Listeners Monitoring Tags Migration

Basic Configuration

Name	sichao-v3-External-162OH8ELJZL71	Creation time	September 14, 2018 at 3:19:04 PM UTC-7
* DNS name	sichao-v3-External-162OH8ELJZL71-668480999.us-east-1.elb.amazonaws.com (A Record)	Hosted zone	Z35SXDOTRQ7X7K
Type	Classic (Migrate Now)	Status	0 of 0 instances in service
Scheme	internet-facing	VPC	vpc-f00c6b8b
Availability Zones	subnet-eb6bfa1 - us-east-1b		

12. For Scaling a Policy:

- As mentioned in Step "11" on page 86, click on **Add policy** on the **Scaling Policies** tab of your Auto Scaling Group (ASG) and name the policy.
- Select a **Metric type** from the drop down list, for example: for Average CPU Utilization, enter a **Target Value** as 75. Add 30 seconds warm-up time the vSRX instances need and leave **Disable scale-in** unchecked.
- Click **Create** to add this policy to the ASG. The ASG executes the policy as required to maintain average CPU utilization at 75.

Sample Configuration of AWS Elastic Load Balancer with vSRX instance for HTTP Traffic

- You need to have your DNS server IP and your Web Server IP (or if your web server is behind a load balancer, then use that load balancer's IP address below instead of the Web Server IP).
- After using your IP addresses in the below configuration, convert this configuration into Base 64 format (refer to: <https://www.base64encode.org/>) and then paste the converted configuration into the UserData field. By doing so, applies the below configuration to the existing default configuration on a vSRX launched in AWS, during the stack creation process.

```
#load_balancer=true
#junos-config
system {
  name-server {
```

```
<Your DNS Server IP>
}
  syslog {
    file messages {
      any any;
    }
  }
}
security {
  address-book {
    global {
      address webserv <Your Web Server IP>/32;
    }
  }
  nat {
    source {
      rule-set src-nat {
        from interface ge-0/0/0.0;
        to zone trust;
        rule rule1 {
          match {
            source-address 0.0.0.0/0;
            destination-port {
              80;
            }
          }
          then {
            source-nat {
              interface;
            }
          }
        }
      }
    }
  }
  destination {
    pool pool1 {
      address <Your Web Server IP>/32;
    }
  }
  rule-set dst-nat {
    from interface ge-0/0/0.0;
    rule rule1 {
      match {
        destination-address 0.0.0.0/0;
```



```
security-zone untrust {
    host-inbound-traffic {
        system-services {
            any-service;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                dhcp;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                dhcp;
            }
        }
    }
}
routing-instances {
    ELB_RI {
        instance-type virtual-router;
        interface ge-0/0/0.0;
        interface ge-0/0/1.0;
    }
}
```


Overview of AWS Elastic Network Adapter (ENA) for vSRX Instances

IN THIS SECTION

- [Benefits | 90](#)
- [Understanding AWS Elastic Network Adapter | 90](#)

Amazon Elastic Compute Cloud (EC2) provides the Elastic Network Adapter (ENA), the next-generation network interface and accompanying drivers that provide enhanced networking on EC2 vSRX instances.

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA).

Benefits

- Supports multiqueue device interfaces. ENA makes use of multiple transmit and receive queues to reduce internal overhead and to increase scalability. The presence of multiple queues simplifies and accelerates the process of mapping incoming and outgoing packets to a particular vCPU.
- The ENA driver supports industry-standard TCP/IP offload features such as checksum offload and TCP transmit segmentation offload (TSO).
- Supports receive-side scaling (RSS) network driver technology that enables the efficient distribution of network receive processing across multiple CPUs in multiprocessor systems, for multicore scaling. Some of the ENA devices support a working mode called low-latency queue (LLQ), which saves several microseconds.

Understanding AWS Elastic Network Adapter

Enhanced networking uses single-root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types. SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. Enhanced networking provides higher bandwidth, higher packet per second (pps) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking.

ENA is a custom network interface optimized to deliver high throughput and packet per second (pps) performance, and consistently low latencies on EC2 vSRX instances. Using ENA for vSRX C5.large instances (with 2 vCPUs and 4-GB memory), you can utilize up to 20 Gbps of network bandwidth. ENA-based enhanced networking is supported on vSRX instances.

The ENA driver exposes a lightweight management interface with a minimal set of memory-mapped registers and an extendable command set through an admin queue. The driver supports a wide range of ENA adapters, is link-speed independent (that is, the same driver is used for 10 Gbps, 25 Gbps, 40 Gbps, and so on), and negotiates and supports various features. The ENA enables high-speed and low-overhead Ethernet traffic processing by providing a dedicated Tx/Rx queue pair per CPU core.

The DPDK drivers for ENA are available at <https://github.com/amzn/amzn-drivers/tree/master/userspace/dpdk>.

NOTE: When AWS ELB application load balancers are used, the eth0 (first) and eth1 (second) interfaces are swapped for the vSRX instance. The AWS ENA detects and rebinds the interface with its corresponding kernel driver.

Software Receive Side Scaling

IN THIS SECTION

- [Overview | 91](#)
- [Understanding Software Receive Side Scaling Configuration | 92](#)

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.

Multi-Core Scaling Support on AWS with SWRSS and ENA

EC2 instance types are predefined by AWS. You cannot launch an instance with an arbitrary number of vCPUs. This scenario leads to a gap between the resource AWS provides and the resource that vSRX 3.0 can use.

As an example: For AWS C5.4xlarge without software RSS, vSRX 3.0 will be launched with 9 vCPUs. Whereas we have 16 vCPUs that can be used. So, the remaining 7 vCPUs offered by AWS are wasted. With Software RSS, the hardware RSS queue limitation is removed. With more software queue available, more vCPUs can be deployed as data vCPUs.

Starting in Junos OS Release 19.4R1, vSRX 3.0 instances with the Software Receive Side Scaling (SWRSS) feature can scale up the number of vCPUs on instances with ENA support in AWS. The ENA enabled instances allow for more RSS queues. With the SWRSS feature, the dynamic ratio between number of vCPUs and RSS queues allows for the scale up of vSRX with larger AWS EC2 instances.

Software RSS supports up to 32 vCPUs. Launching vSRX into EC2 instance with more than 32 vCPUs will not provide further benefits. To support multi-core scaling you need to ensure SWRSS is enabled on vSRX instances.

With this feature support the AWS instances type supported by vSRX are c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge, and c5.9xlarge. For more information, see [Amazon EC2 Instance Types](#).

GTP Traffic with TEID Distribution and SWRSS

IN THIS SECTION

- [Overview GTP Traffic Distribution with TEID Distribution and SWRSS | 94](#)
- [Enabling GTP-U TEID Distribution with SWRSS for Asymmetric Fat Tunnels | 96](#)

Overview GTP Traffic Distribution with TEID Distribution and SWRSS

IN THIS SECTION

- [GTP Traffic Performance with TEID Distribution and SWRSS | 95](#)

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 security 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 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.

Centralized Monitoring and Troubleshooting using AWS Features

IN THIS SECTION

- [Understanding Centralized Monitoring Using Cloudwatch | 99](#)
- [Integration of vSRX with AWS Monitoring and Troubleshooting Features | 103](#)

This topic provides you details on how you can perform monitoring and troubleshooting of your vSRX instances on the AWS console by integrating vSRX with CloudWatch, IAM, and Security Hub.

Understanding Centralized Monitoring Using Cloudwatch

IN THIS SECTION

- Benefits | 101
- CloudWatch Overview | 102
- Security Hub Overview | 102
- Identity and Access Management Console | 103

AWS provides a comprehensive view of various metrics, logs, security events from third-party services across AWS accounts. With the support of CloudWatch, vSRX can publish native metrics and logs to cloud, which you can use to monitor vSRX running status. Security Hub is the single place that aggregates, organizes and prioritizes security alerts.

The CloudWatch logs agent provides an automated way to send log data to CloudWatch Logs from Amazon EC2 instances. The agent pushes log data to CloudWatch Logs.

To integrate vSRX with AWS CloudWatch and Security Hub, AWS creates an agent (named cloudagent) which is running in vSRX. The agent will be able to:

- Collect device metrics and send metrics to AWS CloudWatch
- Collect system and security logs and sends the logs to AWS CloudWatchLog

Any event type (component or log level) that can be collected by the cloudagent under vSRX event log mode is supported for CloudWatch log collection. Events supported for CloudWatchLog are:

- System activities such as Interfaces status (up/down), configuration changes, user login logout and so on.
- Security events such as IDP, SkyATP, and security logs such as UTM logs, and Screen, SkyATP and so on.
- Collect security alerts and import those alerts to Security Hub in security finding format.

To import security events to Security Hub, you need to configure CloudWatch log collection and import the security events based on the log messages.

[Table 15 on page 100](#) provides the list and details of the events that are imported.

Table 15: Events Imported to Security Hub

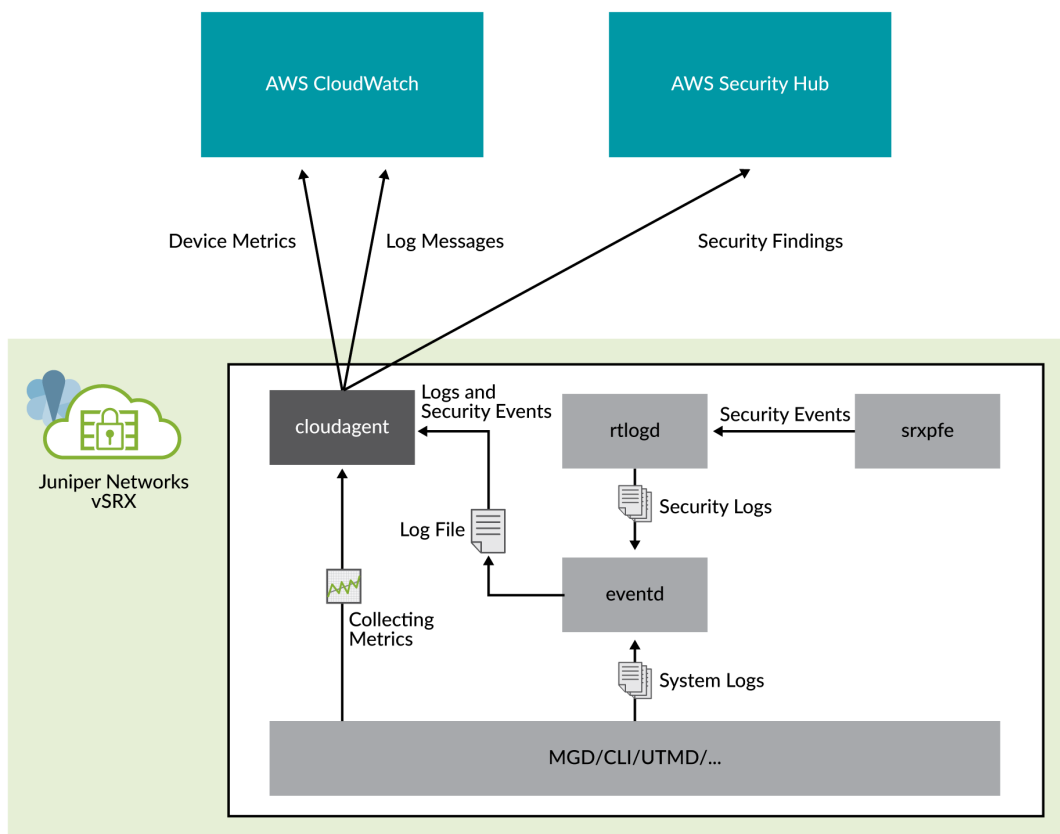
Metric	Unit	Description
ControlPlaneCPUUtil	Percent	Utilization of the CPU on which control plane tasks are running
DataPlaneCPUUtil	Percent	Utilization of each CPU on which data plane tasks are running
DiskUtil	Percent	Disk storage utilization
ControlPlaneMemoryUtil	Percent	Memory utilization of control plane tasks
DataPlaneMemoryUtil	Percent	Memory utilization of data plane task
FlowSessionInUse	Count	Monitors the number of flow session in use, including all those sessions are allocated in valid, invalid, pending and other states.
FlowSessionUtil	Percent	Flow session utilization
RunningProcesses	Count	Number of processes in running state.
Ge00XInputKBPS	Kilobits/Second	Interfaces input statistics on Kilobits per second. Each GE interface will be monitored separately.
Ge00XInputPPS	Count/Second	Interfaces input statistics on packets per second. Each GE interface will be monitored separately.
Ge00XOutputKBPS	Kilobits/Second	Interfaces output statistics on Kilobits per second. Each GE interface will be monitored separately.
Ge00XOutputPPS	Count/Second	Interfaces output statistics on packets per second. Each GE interface will be monitored separately.

Besides the agent running in vSRX, you must configure the AWS console to enable CloudWatch and Security Hub service for vSRX, including:

- Grant privileges for vSRX to post data to CloudWatch and Security Hub
- Create a role with corresponding permission in AWS Identity and Access Management (IAM) console
- Attach the role to vSRX instances in AWS EC2 console
- Configure CloudWatch dashboard to display metric items with chart widget

Figure 9 on page 101 shows how a cloudagent collects data from vSRX and posts to AWS services.

Figure 9: Integration of AWS Cloudwatch on vSRX 3.0



Benefits

- Observability of events and data on a single platform across applications and infrastructure
- Easiest way to collect metric in AWS and on-premises

- Improve operational performance and resource optimization
- Get operational visibility and insight
- Derive actionable insights from logs

CloudWatch Overview

Amazon CloudWatch is a monitoring and management service built for developers, system operators, site reliability engineers (SRE), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your vSRX 3.0 instances running smoothly.

CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, and visualizes it using automated dashboards so you can get a unified view of your AWS resources, applications, and services that run in AWS and on-premises. You can correlate your metrics and logs to better understand the health and performance of your resources. You can also create alarms based on metric value thresholds you specify, or that can watch for anomalous metric behavior based on machine learning algorithms. To take action quickly, you can set up automated actions to notify you if an alarm is triggered and automatically start auto scaling, for example, to help reduce mean-time-to-resolution. You can also dive deep and analyze your metrics, logs, and traces, to better understand how to improve application performance.

Security Hub Overview

AWS Security Hub gives you a comprehensive view of your high-priority security alerts and compliance status across AWS accounts. Security Hub is the single place that aggregates, organizes, and prioritizes security alerts. vSRX supports Security Hub with authentication to post security finding data to Security Hub.

Various security alerts from your vSRX instances are collected by Security Hub. With the integration of Security Hub, you now have a single place that aggregates, organizes, and prioritizes your security alerts, or findings, from your vSRX instances. Your findings are visually summarized on integrated dashboards with actionable graphs and tables. You can also continuously monitor your environment using automated compliance checks based on the AWS best practices and Juniper standards. Enable Security Hub using the management console and once enabled, Security Hub will begin aggregating and prioritizing the findings.

Identity and Access Management Console

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

IAM is a feature of your AWS account offered at no additional charge.

Integration of vSRX with AWS Monitoring and Troubleshooting Features

IN THIS SECTION

- [Enable Monitoring of vSRX Instances with AWS CloudWatch Metric | 103](#)
- [Collect, Store, and View vSRX Logs to AWS CloudWatch | 104](#)
- [Enable and Configure Security Hub on vSRX | 105](#)
- [Grant Permission for vSRX to access AWS CloudWatch and Security Hub | 106](#)

This topic provides details on how to integrate CloudWatch and Security Hub with vSRX 3.0 for centralized monitoring and troubleshooting on the AWS console.

Enable Monitoring of vSRX Instances with AWS CloudWatch Metric

This procedure provides us steps to enable monitoring of vSRX with AWS CloudWatch Metric.

Metric is data about the performance of the system. By enabling CloudWatch Metric monitoring, you can monitor some resources of vSRX instances.

1. Enable CloudWatch and Security Hub using the AWS console.
2. Configure CloudWatch metric in the Cloudwatch agent.

To enable CloudWatch metric monitoring, you need to configure metric namespace and collection interval on the instance by executing the **# set security cloud aws cloudwatch metric namespace <namespace> collect-interval <integer>** command.

A namespace is a container for CloudWatch metrics. Metric in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics. Different vSRX instances can use same CloudWatch metric namespace. Metric from different vSRX instances can be differentiated by dimensional data (instances id/name) in metric value.

Collection interval is the frequency at which the firewall publishes the metrics to CloudWatch. The value can be set between 1 minute and 60 minutes. The default value is 3 minutes.

Once the Cloudwatch metric monitoring is enabled, the cloudagent running on vSRX collects all the required metric and publishes the metric data on the Cloudwatch.

Once monitoring is enabled you can view CloudWatch Metric. CloudWatch metric can be graphed after cloudagent starts to collect and post metric data to the cloud. By selecting the metric namespaces created from vSRX on AWS CloudWatch console, administrator can check and display all metric data. Check AWS CloudWatch guide for how to filter and display on those collected metric.

3. View the Cloudwatch metric data. CloudWatch metrics can be graphed after cloudagent starts to collect and post metric data to cloud.
4. Configure CloudWatch dashboard to display metric items with chart widget.

Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view, even those resources that are spread across different regions. You can manually create a dashboard for the vSRX under monitoring.

Collect, Store, and View vSRX Logs to AWS CloudWatch

CloudWatch Logs are used to monitor, store, and access log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources. For a vSRX instance, cloudagent collects both system and security logs and then post these logs to CloudWatchLog. The log collection in cloudagent will cache logs in a time window and post them to CloudWatchLog in a batch.

This procedure provides you details on how to enable and configure CloudWatch Logs on vSRX

1. To enable log collection for CloudWatchLog, you need to configure a log group, collect interval and from which file to collect log messages on the device.

```
# set security cloud aws cloudwatch log group vsrx-group
# set security cloud aws cloudwatch log file mylog collect-interval 2
# set security cloud aws cloudwatch log file syslog collect-interval 1
```

A log stream is a sequence of log events that share the same source. Each separate source of logs into CloudWatch Logs makes up a separate log stream. For log collection, one vSRX will post logs as a dedicated stream which means vSRX will automatically create a log stream in the destination log group.

A log group is a group of log streams that share the same retention, monitoring, and access control settings. By defining the log groups on the vSRX instance, you can specify which streams are placed into which group.

Collection interval is the frequency at which the firewall publishes logs to CloudWatchLog. The value can be set between 1 minute and 60 minutes. The default value is 3 minutes.

Three vSRX log files can be collected in CloudWatch simultaneously per vSRX instance. Each log file will create a corresponding a log stream in Cloudwatch. The log stream will be named under log group with convention <vsrx_instance_id> <log_file_name>.

After you enable CloudWatch logging in the cloudagent on vSRX instances, you need to configure syslog message file.

2. Configure the syslog message file.

Any filters can be applied based on vSRX syslog filtering. It provides the capability to define which log messages will be sent to CloudWatchLogs. For example, the below configuration means system will log any error messages to the syslog file under the **/var/log** and cloudagent will collect the messages from **/var/log/syslog** and post the messages to CloudWatchLogs.

```
# set security cloud aws cloudwatch log file syslog collect-interval 1
# set system syslog file syslog any error
```

3. View and search vSRX logs on CloudWatchLog console. Log groups and stream will be created automatically after configured on vSRX instances.

Select the log group and stream to check and search those logs sent to CloudWatch from the vSRX instance.

Enable and Configure Security Hub on vSRX

To import security events to AWS Security Hub, you need to configure CloudWatch log collection and then import the security events based on the log messages.

For example:

```
# set security cloud aws cloudwatch log group vsrx-group
# set security cloud aws cloudwatch log file mylog security-hub-import
# set security cloud aws cloudwatch log file mylog collect-interval 1
# set system syslog file mylog any any
# set system syslog file mylog structured-data
```

In the above configuration you are configuring CloudWatch log collection on file mylog under **/var/log** directory and any security events in the log file will be imported from the vSRX to Security Hub in the AWS security finding format.

NOTE: The `security-hub-import` option is only supported on log files with structured-data format. Which means if a message is logged with plain text format, security events in log messages cannot be converted to AWS security finding and imported to Security Hub.

You can view the security findings posted from vSRX on the Security Hub console.

The screenshot displays the AWS Security Hub console. On the left, a table lists several findings. The first finding is titled 'Account Compromise title, to test security hub' with a severity of 'LOW'. The second finding is also titled 'Account Compromise title, to test security hub' with a severity of 'LOW'. The third finding is titled '1.22 Ensure IAM policies that allow full administrative privileges are not created' with a severity of 'LOW'. The fourth finding is titled '1.10 Ensure IAM password policy' with a severity of 'LOW'. On the right, a detailed view of a finding is shown, titled 'Account Compromise title, to test security hub'. The finding ID is '2019-03-05 10:08:16.179525'. The account ID is '016135515484'. The severity (original) is '23'. The severity (normalized) is '23'. The created at date is '2019-03-05T10:08:16Z'. The updated at date is '2019-03-05T10:08:16Z'. The severity label is 'LOW'. The company is 'Personal'. The resource type is 'Juniper-vSRX' and the resource ID is 'i-05a6a684ff509cb61'.

Severity	Company	Product	Title	Resource ID	Resource
LOW	Personal	Default	Account Compromise title, to test security hub	i-05a6a684ff509cb61	Juniper-vSRX
LOW	Personal	Default	Account Compromise title, to test security hub	phlich@juniper.net	Email Address
LOW	AWS	Security Hub	1.22 Ensure IAM policies that allow full administrative privileges are not created	AWS::Account: 016135515484	AwsAccount
LOW	AWS	Security	1.10 Ensure IAM password policy	AWS::Account: 016135515484	AwsAccount

Grant Permission for vSRX to access AWS CloudWatch and Security Hub

This section provides you details on how to enable access on vSRX instances to interact with AWS CloudWatch and Security Hub.

1. Create an IAM role using AWS IAM console.

Login to AWS IAM console, create IAM role and attach the role to vSRX instances to grant those permissions. You must create an IAM role before you can launch an instance with that role or attach it to an instance. For more information, see [IAM Roles for Amazon EC2](#).

2. To launch an instance with an IAM role or to attach or replace an IAM role for an existing instance, permissions have to be granted to pass the role to the instance. AWS has to grant permission to pass an IAM role to an instance. For more information, see [Granting an IAM User Permission to Pass an IAM Role to an Instance](#).

3. Configure an IAM role role on the AWS console and attach the role to vSRX instance. After you create and IAM role, the role can be viewed on IAM console and edited as necessary.

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation options: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area is titled 'Permissions' and shows 'Permissions policies (3 policies applied)'. A blue 'Attach policies' button is visible. Two policies are listed:

- vSRXSecurityHub** (Managed policy):

Service	Access level	Resource	Request conditions
SecurityHub	Full: Write	All resources	None
- vSRXCloudWatchPolicy** (Managed policy):

Service	Access level	Resource	Request conditions
CloudWatch	Full: Write	All resources	None

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation options: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area is titled 'Roles' and shows a 'Create role' button and a search bar. A table of roles is displayed:

Role name	Description
<input type="checkbox"/> AWSServiceRoleForSecurityHub	A service-linked role required for AWS Security Hub to access your resources.
<input type="checkbox"/> AWSServiceRoleForSupport	Enables resource access for AWS to provide billing, administrative and support services
<input type="checkbox"/> AWSServiceRoleForTrustedAdvisor	Access for the AWS Trusted Advisor Service to help reduce cost, increase performance, and improve availability
<input type="checkbox"/> CloudWatchAgentServerRole	Allows EC2 instances to call AWS services on your behalf.
<input checked="" type="checkbox"/> vSRXRole	Allows EC2 instances to call AWS services on your behalf.

4. Attach an IAM role to vSRX instances by selecting a IAM role and the vSRX instance ID under the **Attach/Replace IAM Role** tab on the AWS console as shown in [Figure 10 on page 108](#). With the

created role, you can enable CloudWatch and Security Hub access for vSRX instance by attaching the role.

Figure 10: Attach or Replace IAM Role to the vSRX Instances

The screenshot shows the AWS Management Console interface for attaching or replacing an IAM role on an EC2 instance. The breadcrumb navigation indicates the path: **Instances** > Attach/Replace IAM Role. The main heading is **Attach/Replace IAM Role**. Below the heading, there is a descriptive text: "Select an IAM role to attach to your instance. If you don't have any IAM roles, choose Create new IAM role to create a role. If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role." The instance ID is shown as **Instance ID** i-05a6a684ff509cb61 (philch-ecs) with an information icon. The **IAM role*** dropdown menu is set to **vSRXRole**. To the right of the dropdown is a refresh icon and a link that says **Create new IAM**. A search filter is overlaid on the dropdown menu, with the text **Filter by attributes**. The filter results show a list of profile names: **No Role**, **CloudWatchAgentServerRole**, and **vSRXRole** (which is highlighted in orange). A note on the left side of the page states *** Required**.

Deploying vSRX 3.0 for Securing Data using AWS KMS

IN THIS SECTION

- Integrate AWS KMS with vSRX 3.0 | 109

Integrate AWS KMS with vSRX 3.0

A wrapper library is available in Junos to enable VPN and other applications (such as mgd) to integrate and communicate AWS KMS with vSRX 3.0. This wrapper library provides interface to Key Management Service (KMS) using PKCS#11 APIs. Junos applications use this wrapper library with updated support for AWS cloud platform to communicate with KMS.

To enable and setup vSRX 3.0 to access KMS on AWS.

1. Launch vSRX 3.0 instance on AWS.
2. Setup KMS and DynamoDB for vSRX 3.0.

Before you can use vSRX to communicate with KMS service, you need to setup AWS environment/account by doing the following:

- a. Create a DynamoDB table.

DynamoDB service on AWS is used by KMS PKCS11 process to store and manage key information created by vSRX 3.0 applications. Hence a dynamo DB table needs to be created and the name of table created should be passed on to vSRX 3.0 by running the **request security hsm set dynamo-db <>** command. This CLI is used to specify the name of DynamoDB needed by vSRX 3.0 to support KMS. This is the first CLI to be executed to enable vSRX 3.0 access to KMS.

NOTE: If you are having problems deploying the template or creating DynamoDb table using AWS GUI, please contact your administrator and make sure your account has permissions. While creating DynamoDB, refer the guidelines at [Naming Rules and Data Types](#).

See "[Cloud Formation Template for DynamoDB](#)" on page 111 for information on AWS Cloud formation templates for DynamoDB.

- b. Create IAM role to enable access for vSRX 3.0 instance.

KMS service is available for EC2 instances such as vSRX on AWS. As mentioned above once DynamoDB table is created, for vSRX to use the service, access policies need to be enabled for the instance. vSRX will also use Cloud Watch to log any events, hence policies to enable this service for instance are also needed.

These policies are minimum required to enable vSRX instance to use KMS service. Once the role is created, you can then attach this IAM role to the instance from GUI or using AWS CLI.

See ["Cloud Formation Template to Create IAM Role" on page 111](#) for information on AWS Cloud formation templates for creating IAM role.

c. Attach IAM role to vSRX instance.

After creating the IAM role, attach it to the vSRX instance either from GUI or using AWS CLI. See [IAM roles for Amazon EC2](#).

3. Check HSM status using the **show security hsm status** command. This CLI output is updated to display DynamoDB being used along with HSM reachability, Master binding Key(MBK), and Master Encryption Key (MEK) status.
4. After enabling the KMS service, you need to specify the Master Encryption Key (MEK) using the **request security hsm master-encryption-password set plain-text-password** command on vSRX 3.0.

Once you specify the MEK, vSRX 3.0 creates the RSA 2048 key pair (MBK) in KMS and encrypts MEK using Master binding Key (MBK) in KMS. MEK is then used as a key for encrypting data at rest such as hash of configuration, private key pair files and master-password file, if present.

5. Change the Master Encryption Password.

If you want to change the master encryption password then you can run the **request security hsm master-encryption-password set plain-text-password** command from operational mode:

NOTE: It is recommended that no configuration changes are made while you are changing the master encryption password.

The system checks if the master encryption password is already configured. If master encryption password is configured, then you are prompted to enter the current master encryption password.

The entered master encryption password is validated against the current master encryption password to make sure these master encryption passwords match. If the validation succeeds, you will be prompted to enter the new master encryption password as plain text. You will be asked to enter the key twice to validate the password.

The system then proceeds to re-encrypt the sensitive data with the new master encryption password. You must wait for this process of re-encryption to complete before attempting to change the master encryption password again.

If the encrypted master encryption password file is lost or corrupted, the system will not be able to decrypt the sensitive data. The system can only be recovered by re-importing the sensitive data in clear text, and re-encrypting them.


```

String
Type:
Description: Name of DynamoDB Handle GSI
Default:
handle-index
Resources:
libpkcs11awsDDBTable:
Type:
AWS::DynamoDB::Table
Properties:
TableName: !Ref libpkcs11awsDDBTableName
ProvisionedThroughput:
ReadCapacityUnits: "5"
WriteCapacityUnits: "5"
AttributeDefinitions:
-
AttributeType: "S"
-
AttributeName: "handle"
AttributeType: "N"
KeySchema:
-
AttributeName: "uuid"
KeyType:
"HASH"
-

```

```

AttributeName: "handle"
KeyType:
"RANGE"

GlobalSecondaryIndexes:
-

IndexName: !Ref libpkcs11awsDDBGSIName

ProvisionedThroughput:

ReadCapacityUnits: "5"

WriteCapacityUnits: "5"

KeySchema:
-

AttributeName: "handle"
KeyType:
"HASH"

Projection:
KeyType:
"HASH"

Projection:

ProjectionType: "ALL"

Outputs:

libpkcs11awsDDBTableArn:
Value: !

GetAtt libpkcs11awsDDBTable.Arn

Export:
Name: !

Sub '${AWS::StackName}:libpkcs11awsDDBTableArn'

```



```
***** End *****
```

Cloud Formation Template to Create IAM Role

Deploy this template by executing the `$ aws cloudformation create-stack \ --stack-name libpkcs11aws-iam \ --template-body file:/// $PWD/iam_role.cfn.yaml \ --capabilities CAPABILITY_NAMED_IAM` AWS CLI.

NOTE: Change the names and IDs as required.

```
***** Start *****

AWSTemplateFormatVersion: '2010-09-09'

Description: "libpkcs11AWS EC2 IAM Instance Role"

Parameters:

libpkcs11awsDDBStackName:

Type: String

Default: libpkcs11aws-ddb

Resources:

libpkcs11awsEC2Role:

Type: AWS::IAM::Role

Properties:

RoleName: libpkcs11awsEC2Role

Path: "/"
```

ManagedPolicyArns:

"arn:aws:iam::aws:policy/CloudWatchFullAccess"

"arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess"

"arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess"

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

-

Effect: Allow

Principal:

Service:

ec2.amazonaws.com

Action:

sts:AssumeRole

Policies:

PolicyName: "DynamoDBTableFullAccess"

PolicyDocument:

Version: "2012-10-17"

Statement:

```

Effect: "Allow"

Action:
    "dynamodb:*"

Resource:
    'Fn::ImportValue' : !Sub '${libpkcs11awsDDBStackName}:libpkcs11awsDDBTableArn'

PolicyName: "KMSFullAccess"

PolicyDocument:

Version: "2012-10-17"

Statement:

    Effect: "Allow"

    Action:

    Effect: "Allow"

    Action:

    "kms:*"

Resource:

libpkcs11awsEC2RoleIP:

DependsOn: libpkcs11awsEC2Role

Type: AWS::IAM::InstanceProfile

Properties:

Path: "/"

```

Roles:

-

libpkcs11awsEC2Role

InstanceProfileName: libpkcs11awsEC2Role

***** End *****

4

CHAPTER

vSRX in AWS Use Cases

Example: Configuring NAT for vSRX | 119

Example: Configure VPN on vSRX Between Amazon VPCs | 121

Example: Configure Juniper Sky ATP for vSRX | 127

Example: Configuring NAT for vSRX

IN THIS SECTION

- [Before You Begin | 119](#)
- [Overview | 119](#)
- [Configuration | 120](#)
- [Configuring NAT | 120](#)

This example shows how to configure vSRX to NAT all hosts behind the vSRX instance in the Amazon Virtual Private Cloud (Amazon VPC) to the IP address of the vSRX egress interface on the untrust zone. This configuration allows hosts behind vSRX in a cloud network to access the Internet.

Before You Begin

Ensure that you have installed and launched a vSRX instance in an Amazon VPC.

Overview

A common cloud configuration includes hosts that you want to grant access to the Internet, but you do not want anyone from outside your cloud to get access to your hosts. You can use vSRX in an Amazon VPC to NAT traffic inside the Amazon VPC from the public Internet.

Configuration

Configuring NAT

IN THIS SECTION

- [Procedure | 120](#)

Procedure

Step-by-Step Procedure

To configure NAT on the vSRX instance:

1. Log in to the vSRX console in configuration edit mode (See "[Configure vSRX Using the CLI](#)" on page 64).
2. Set the IP addresses for vSRX revenue interfaces.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.10.197/24
set interfaces ge-0/0/1 unit 0 family inet address 10.0.20.1/24
```

3. Set up the untrust security zone.

```
set security zones security-zone untrust host-inbound-traffic system-services https
set security zones security-zone untrust host-inbound-traffic system-services ssh
set security zones security-zone untrust interfaces ge-0/0/0.0
```

4. Set up the trust security zone.

```
set security zones security-zone trust host-inbound-traffic system-services https
set security zones security-zone trust host-inbound-traffic system-services ssh
```

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

5. Set up the security policies.

```
set security policies from-zone trust to-zone untrust policy test match source-address any
set security policies from-zone trust to-zone untrust policy test match destination-address any
set security policies from-zone trust to-zone untrust policy test match application any
set security policies from-zone trust to-zone untrust policy test then permit
```

6. Configure NAT.

```
set security nat source rule-set SNAT_RuleSet from zone trust
set security nat source rule-set SNAT_RuleSet to zone untrust
set security nat source rule-set SNAT_RuleSet rule SNAT_Rule match source-address 0.0.0.0/0
set security nat source rule-set SNAT_RuleSet rule SNAT_Rule then source-nat interface
commit
```

RELATED DOCUMENTATION

[vSRX Virtual Firewall-Based AWS Transit VPC](#)

[Day One: Amazon Web Services with vSRX Cookbook](#)

Example: Configure VPN on vSRX Between Amazon VPCs

IN THIS SECTION

- [Before You Begin | 122](#)
- [Overview | 122](#)
- [vSRX1 VPN Configuration | 122](#)

- Verification | 126

This example shows how to configure IPsec VPN between two instances of vSRX on different Amazon VPCs.

Before You Begin

Ensure that you have installed and launched a vSRX instance in an Amazon VPCs.

See [SRX Site-to-Site VPN Configuration Generator](#) and [How to troubleshoot a VPN tunnel that is down or not active](#) for additional information.

Overview

You can use IPsec VPN to secure traffic between two Amazon VPCs using two vSRX instances.

vSRX1 VPN Configuration

IN THIS SECTION

- Procedure | 122
- vSRX2 VPN Configuration | 124

Procedure

Step-by-Step Procedure

To configure IPsec VPN on vSRX1:

1. Log in to the vSRX1 console in configuration edit mode (See ["Configure vSRX Using the CLI" on page 64.](#))
2. Set the IP addresses for vSRX1 revenue interfaces.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.10/24
set interfaces ge-0/0/1 unit 0 family inet address 10.10.10.10/24
set interfaces st0 unit 1 family inet address 10.0.250.10/24
```

3. Set up the untrust security zone.

```
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust host-inbound-traffic system-services https
set security zones security-zone untrust host-inbound-traffic system-services ssh
set security security-zone untrust interfaces ge-0/0/0.0
set security security-zone untrust interfaces st0.1
```

4. Set up the trust security zone.

```
set security zone trust host-inbound-traffic system-services https
set security zone trust host-inbound-traffic system-services ssh
set security zone trust host-inbound-traffic system-services ping
set security security-zone trust interfaces ge-0/0/1.0
```

5. Configure IKE.

```
set security ike proposal AWS_IKE_Proposal authentication-method pre-shared-keys
set security ike proposal AWS_IKE_Proposal dh-group group2
set security ike proposal AWS_IKE_Proposal authentication-algorithm sha-256
set security ike proposal AWS_IKE_Proposal encryption-algorithm aes-256-cbc
set security ike proposal AWS_IKE_Proposal lifetime-seconds 1800
set security ike policy AWS-R mode aggressive
set security ike policy AWS-R proposals AWS_IKE_Proposal
set security ike policy AWS-R pre-shared-key ascii-text preshared-key
set security ike gateway AWS-R ike-policy AWS-R
set security ike gateway AWS-R address 198.51.100.10
set security ike gateway AWS-R local-identity user-at-hostname "source@example.net"
```

```
set security ike gateway AWS-R remote-identity user-at-hostname "dest@example.net"
set security ike gateway AWS-R external-interface ge-0/0/0
```

6. Configure IPsec.

```
set security ipsec proposal AWS_IPSEC protocol esp
set security ipsec proposal AWS_IPSEC authentication-algorithm hmac-sha1-96
set security ipsec proposal AWS_IPSEC encryption-algorithm aes-256-cbc
set security ipsec policy AWS_IPSEC_POL proposals AWS_IPSEC
set security ipsec vpn aws-aws bind-interface st0.1
set security ipsec vpn aws-aws ike gateway AWS-R
set security ipsec vpn aws-aws ike ipsec-policy AWS_IPSEC_POL
set security ipsec vpn aws-aws establish-tunnels immediately
```

7. Configure routing.

```
set routing-instances aws instance-type virtual-router
set routing-instances aws interface ge-0/0/0.0
set routing-instances aws interface ge-0/0/1.0
set routing-instances aws interface st0.1
set routing-instances aws routing-options static route 0.0.0.0/0 next-hop 10.0.0.1
set routing-instances aws routing-options static route 10.20.20.0/24 next-hop st0.1
commit
```

vSRX2 VPN Configuration

Step-by-Step Procedure

To configure IPsec VPN on vSRX2:

1. Log in to the vSRX2 console in configuration edit mode (See ["Configure vSRX Using the CLI"](#) on page 64).
2. Set the IP addresses for the vSRX2 revenue interfaces.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.0.10/24
set interfaces ge-0/0/1 unit 0 family inet address 10.20.20.10/24
set interfaces st0 unit 1 family inet address 10.0.250.20/24
```

3. Set up the untrust security zone.

```
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust host-inbound-traffic system-services https
set security zones security-zone untrust host-inbound-traffic system-services ssh
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces st0.1
```

4. Set up the trust security zone.

```
set security zones security-zone trust host-inbound-traffic system-services https
set security zones security-zone trust host-inbound-traffic system-services ssh
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces ge-0/0/1.0
```

5. Configure IKE.

```
set security ike proposal AWS_IKE_Proposal authentication-method pre-shared-keys
set security ike proposal AWS_IKE_Proposal dh-group group2
set security ike proposal AWS_IKE_Proposal authentication-algorithm sha-256
set security ike proposal AWS_IKE_Proposal encryption-algorithm aes-256-cbc
set security ike proposal AWS_IKE_Proposal lifetime-seconds 1800
set security ike policy AWS-R mode aggressive
set security ike policy AWS-R proposals AWS_IKE_Proposal
set security ike policy AWS-R pre-shared-key ascii-text preshared-key
set security ike gateway AWS-R ike-policy AWS-R
set security ike gateway AWS-R address 203.0.113.10
set security ike gateway AWS-R local-identity user-at-hostname "dest@example.net"
set security ike gateway AWS-R remote-identity user-at-hostname "source@example.net"
set security ike gateway AWS-R external-interface ge-0/0/0
```

6. Configure IPsec.

```
set security ipsec proposal AWS_IPSEC protocol esp
set security ipsec proposal AWS_IPSEC authentication-algorithm hmac-sha1-96
set security ipsec proposal AWS_IPSEC encryption-algorithm aes-256-cbc
set security ipsec policy AWS_IPSEC_POL proposals AWS_IPSEC
set security ipsec vpn aws-aws bind-interface st0.1
```

```
set security ipsec vpn aws-aws ike gateway AWS-R
set security ipsec vpn aws-aws ike ipsec-policy AWS_IPSEC_POL
set security ipsec vpn aws-aws establish-tunnels immediately
```

7. Configure routing.

```
set routing-instances aws instance-type virtual-router
set routing-instances aws interface ge-0/0/0.0
set routing-instances aws interface ge-0/0/1.0
set routing-instances aws interface st0.1
set routing-instances aws routing-options static route 0.0.0/0 next-hop 10.0.0.1
set routing-instances aws routing-options static route 10.10.10.0/24 next-hop st0.1
commit
```

Verification

IN THIS SECTION

- [Verify Active VPN Tunnels | 126](#)

Verify Active VPN Tunnels

Purpose

Verify that the tunnel is up on both vSRX instances on AWS.

Action

```
ec2-user@> show security ipsec security-associations
```

```
Total active tunnels: 1
ID          Algorithm          SPI          Life:sec/kb      Mon lsys Port  Gateway
```

```
<131074 ESP:aes--cbc--256/sha1 de836105 1504/ unlim -- root 4500 52.200.89.XXX  
>131074 ESP:aes--cbc--256/sha1 b349bc84 1504/ unlim -- root 4500 52.200.89.XXX
```

NOTE: Starting in Junos OS Release 17.4R1, the default user name has changed from **root@** to **ec2-user@**.

RELATED DOCUMENTATION

[vSRX Virtual Firewall-Based AWS Transit VPC](#)

[Day One: Amazon Web Services with vSRX Cookbook](#)

[VPN Feature Guide for Security](#)

[Application Firewall Overview](#)

Example: Configure Juniper Sky ATP for vSRX

IN THIS SECTION

- [Before You Begin | 127](#)
- [Overview | 128](#)
- [Juniper Sky ATP Configuration | 128](#)

This example shows how to configure Juniper Sky™ Advanced Threat Prevention (Juniper Sky ATP) on a vSRX instance that is deployed in a virtual private cloud (VPC).

Before You Begin

Ensure that you have installed and launched a vSRX instance in a VPC.

Overview

You can use Juniper Sky ATP, a cloud-based solution, along with vSRX to protect all hosts in your network against evolving security threats.

Juniper Sky ATP Configuration

IN THIS SECTION

- [Procedure | 128](#)

Procedure

Step-by-Step Procedure

To configure Juniper Sky ATP on a vSRX instance:

1. Log in to the vSRX instance using SSH and start the CLI.

```
root@% cli
root@>
```

2. Enter configuration mode.

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

3. Set up the correct data interface for the active advanced antimalware (AAMW) service instead of using the default fxp0 interface.

```
root@# set services advanced-anti-malware connection source-interface ge-0/0/0.0
```

4. Configure NAT.

```
root@# set security nat source rule-set rs1 from zone trust
root@# set security nat source rule-set rs1 to zone untrust
root@# set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
root@# set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
root@# set security nat source rule-set rs1 rule r1 then source-nat interface
```

5. Set up virtual routing instance for the correct data interface for AAMW service.

```
root@# set routing-instances vsrx-vr1 instance-type virtual-router
root@# set routing-instances vsrx-vr1 routing-options static route 0.0.0.0/0 next-hop 10.4.1.1
root@# set routing-instances vsrx-vr1 interface ge-0/0/0.0
root@# set routing-instances vsrx-vr1 interface ge-0/0/1.0
```

6. Verify the configuration.

```
root@# commit check
configuration check succeeds
```

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

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

8. Optionally, you can verify the configuration by running the following show commands in the configuration mode:

- show services advanced-anti-malware connection | display set
- show security nat | display set
- show routing-instances vsrx-vr1 | display set

RELATED DOCUMENTATION

| [Juniper Sky Advanced Threat Prevention Administration Guide](#)

5

CHAPTER

Monitoring and Troubleshooting

Monitoring | 131

Backup and Recovery | 132

Finding the Software Serial Number for vSRX | 133

Monitoring

IN THIS SECTION

- [Monitoring vSRX Instances Using SNMP | 131](#)
- [Monitoring vSRX Instances Using AWS Features | 132](#)

This topic provides details on how you can monitor your vSRX instances using SNMP and AWS monitoring features.

Monitoring helps in maintaining the reliability, availability, and performance of your vSRX instances and your AWS solutions. You should collect monitoring data from all your AWS solutions so that you can easily debug any multi-point failure.

Monitoring vSRX Instances Using SNMP

You can monitor your vSRX instance details such as health and storage at instance level, using SNMP monitoring.

For details on SNMP monitoring, refer the SNMP MIB information in the MIB Explorer at: <https://apps.juniper.net/mib-explorer/>.

You can also find all the applicable SNMP OIDs from the Juniper MIB from the vSRX CLI, using the **show snmp mib walk 1.3.6.1.4.1.2636** command.

Some examples of useful OID's for monitoring system health are:

```
jnxOperatingCPU.1.1.0.0
jnxOperating5MinAvgCPU.1.1.0.0
jnxFwddMicroKernelCPUUsage.0
jnxFwddRtThreadsCPUUsage.0
jnxHrStoragePercentUsed.1
jnxJsNodeCurrentTotalSession.0
jnxJsNodeMaxTotalSession.0
jnxJsNodeSessionCreationPerSecond.0
```

NOTE: For monitoring storage capacity on the vSRX instance you can use SNMP monitoring. Using SNMP monitoring, you can be notified for any vSRX instance storage that is impacted. The storage related OID indicates the storage percentage, which is used to detect the storage capacity.

For best practices for enabling SNMP monitoring in Junos, see https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/snmp-best-practices-basic-config.html.

Monitoring vSRX Instances Using AWS Features

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of the tools to do the monitoring for you, while some of the tools require manual intervention. For more information, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_automated_manual.html.

Monitoring Your Instances Using CloudWatch—You can monitor your instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. For more information see:

- **Monitoring Amazon EC2**—https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html.
- **Monitoring Your Instances Using CloudWatch**—<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html> and <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch.html>.

Backup and Recovery

This topic provides details on how you can backup and recover your configuration files in case of instance or service failure, both externally within AWS and locally on your vSRX instance console

To save the vSRX configuration file locally, perform the following steps:

1. Log into the vSRX instance and go to the configuration mode.

2. Execute the command `save /var/tmp/ <file-name>`

The current vSRX configurations are saved in the above mentioned path.

3. Using your Secure Copy Protocol (SCP) client, download the saved configuration files to your local system.
4. Using the instructions at https://aws.amazon.com/getting-started/tutorials/backup-files-to-amazon-s3/?trk=gs_card, create a S3 bucket on AWS and upload the saved configuration file. You can retrieve the saved configuration file as well.

For backup and recovery of configuration files within AWS:

NOTE: You must have an FTP server that is accessible from the vSRX instance.

1. Run the below configuration.

```
External example system {
  archival {
    configuration {
      transfer-on-commit;
      archive-sites {
        "ftp://username:password@192.168.1.10";
      }
    }
  }
}
```

2. You can then run and commit the following configuration command on the vSRX instance.

```
set system archival configuration transfer-on-commit archive-sites ftp://
username:password@<FTP_Server_IP_Address> .
```

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 used	Licenses installed	Licenses needed	Expiry
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: 9XXXXAXXXXXX9
```

```
Customer ID: MyCompany
```

```
Features:
```

```
Virtual Appliance - Virtual Appliance  
permanent
```

For more information, see [Licenses for vSRX](#)