Engineering
Simplicity

# Connected Security Distributed Services Architecture Deployment Guide

Published
2025-01-08

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

## YEAR 2000 NOTICE

## END USER LICENSE AGREEMENT

# Table of Contents

# About This Guide

Connected Security Distributed Services (CSDS) Architecture extends security services across your distributed centers of data by securing and operationalizing your data. In this guide, you'll learn how CSDS Architecture offers off-box security services solution and efficiently scales security services across distributed data centers. Read further to plan your environment and deploy the distributed security services.

1

CHAPTER

# CSDS Architecture Overview

# CSDS Overview

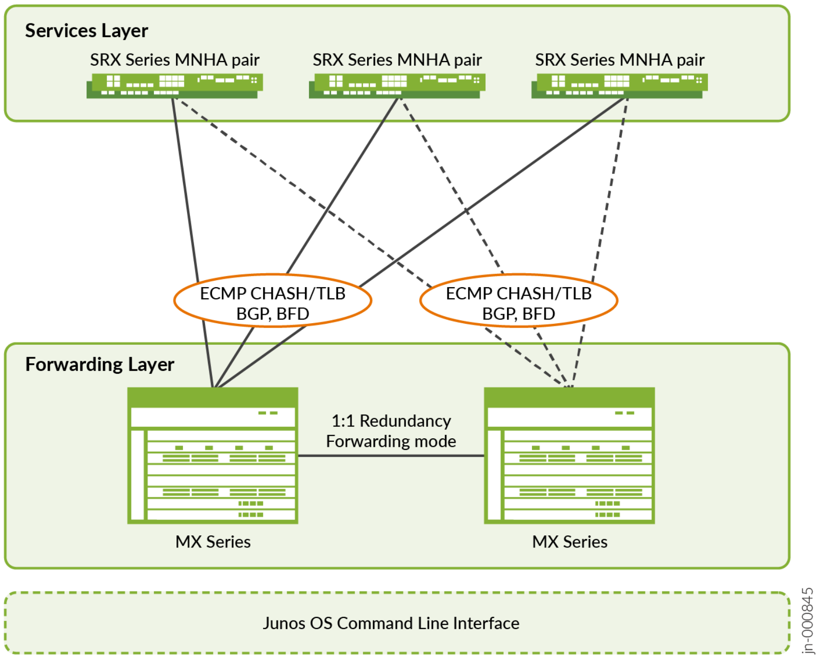
**SUMMARY**

Read this topic to learn about the Juniper's Connected Security Distributed Services (CSDS) Architecture and benefits.

**IN THIS SECTION**

Juniper's Connected Security Distributed Services Architecture delivers a scalable, distributed security architecture design that fully decouples the forwarding and security services layers. This approach enables existing Juniper MX series routers to act as intelligent forwarding engine and load balancer with path redundancy capability.

**Figure 1: CSDS Architecture**



## Benefits

CSDS Architecture delivers the following unique capabilities:

- Scalability: Ability to scale horizontally and elastically as needed, with no chassis limitations. All distributed firewalls function together as a fabric, enabling automated resiliency with multi-path redundancy. If one fails, others automatically load balance and pick up the slack.
- Simplicity: Manage all distributed firewall engines as a single logical element, regardless of the count. You can deploy in the right form at each site almost similar to adding virtual service cards to a chassis.
- Flexibility: Forwarding performance and services can scale independently by decoupling forwarding and services layers. This helps to deliver the right size security solution for every deployment and the ability to mix and match different form factors. Additionally, you can continue to use the existing Juniper firewalls in the new architecture, ensuring that the processes and policies all remain intact.

# CSDS Solution Architecture

**SUMMARY**

Read this topic to understand the components of CSDS Architecture.

The CSDS Architecture primarily consists of the following components:

- Forwarding Layer—The forwarding layer includes MX Series routers that receive and return traffic of the underlying network and distribute upwards to the different services layer devices. The MX Series routers in this layer serves as the single pane of glass responsible for synchronizing and distributing the configuration to the service layer devices. You can deploy the MX Series routers in 1:1 redundancy.
- Services Layer—The services layer provides security features using the SRX Series Firewalls. The layer supports different SRX Series Firewalls in the solution but a group of same firewall models together offer a security service offering such as carrier-grade NAT (CGNAT), IPsec VPN. Note that multiple groups, each hosting different security services can also co-exist. The guide covers configuration examples with one group of SRX Series Firewalls.

- Distribution Layer (Optional)—The distribution layer is placed between the forwarding and the services layer. The devices in this layer primarily provide additional port count, if needed, when enough ports are not available on the devices in the forwarding and the services layers. The devices can also offer different ports speeds and types that are not built in into the forwarding or services layer devices. These devices serve as a switch fabric that interconnects all the different devices. You can use QFX Series in this layer for large-scale deployments.

- Management Layer—The management layer provides a management platform for the entire CSDS solution and connects to the forwarding layer as a single pane of glass. The management layer includes the capability to monitor the utilization of the services layer devices. In this layer, you can optionally use EX Series switches for the management of devices.

Figure 1 on page 2 depicts the high-level architecture of the CSDS solution.

2

CHAPTER

# CSDS Deployment Overview

# Supported Platforms

**SUMMARY**

Read this topic to know about the supported platforms in the CSDS Architecture to help you prepare for the deployment.

Table 1 on page 6 shows the list of components and associated Junos OS devices.

**Table 1: Supported Platforms in CSDS Architecture**

| CSDS Components | Junos OS Devices | Junos OS Release |
|---|---|---|
| Forwarding Layer | MX304, MX960 | 23.4R1 or later |
| | MX240, MX304, MX480, MX960, MX10004, and MX10008 with centralized management | 24.4R1 or later |
| Services Layer | SRX4600, vSRX 3.0 | 23.4R1 or later |
| | SRX4600, vSRX 3.0 with centralized management | 24.4R1 or later |
| Distribution Layer (Optional)* | QFX5220-32CD, QFX5210-128 | 23.4R1 or later |

* You can use QFX Series for large-scale deployments.

# Supported Features in CSDS

SUMMARY

Read this topic to know about the various features supported in the CSDS Architecture to help you prepare for the deployment.

IN THIS SECTION

The solution offers carrier class security services such as:

- Stateful Firewall (Stateful FW)
- IPsec VPN
- Carrier-grade NAT (CGNAT)

## Stateful Firewall Services

The solution provides stateful firewall services offering an extra layer of security by using state information derived from past communications and other applications to make dynamic control decisions for new communication attempts. The stateful firewall service flow is identified by—source address, source port, destination address, destination port and protocol. The SRX Series Firewall enforces security policies to control transit traffic in terms of the traffic that can pass through the firewall, and the actions that need to take place on the traffic as it passes through the firewall.

See Security Policies User Guide for Security Devices for more details.

## IPsec VPN

The SRX Series Firewalls offer high-performance network security gateway solutions such as IPsec VPN with CSDS Architecture. As part of the IPsec VPN service, the solution offers encrypted tunnels for secure communications with IKE gateways. The solution supports the following features:

- Route based VPNs
- NAT-T

- AutoVPN
- Remote Access VPN using Juniper Secure Connect
- Dead Peer Detection (DPD)
- Power Mode IPsec VPN
- Initiator Mode VPN without load balancing/scale-out support. Ensure that the initiator and the responder do not coexist on the same SRX Series Firewalls group.

See IPsec VPN User Guide for more details.

## Carrier-Grade NAT

The CSDS Architecture provides carrier-grade NAT and Network Address Port Translation (NAPT) functionality for translating IP and port addresses. The solution supports the following features:

- NAPT44
- NAPT44 with Persistent-NAT
- NAPT44 with Address-Persistent
- Deterministic NAT44
- NAT with Policy
- NAT with Port-Overloading
- NAT with Hairpinning
- NAT with ALGs

See Network Address Translation User Guide for more details.

# Redundancy Support

**SUMMARY**

Read this topic to learn about the redundancy options available in the CSDS Architecture to help you prepare for the deployment.

The CSDS Architecture supports redundancy in the forwarding layer and the services layer.

- Forwarding layer redundancy uses ECMP based Consistent Hashing or Routing Engine-based Traffic Load Balancer (TLB) on MX Series for load balancing and SRD for redundancy between two MX Series.
- Services layer uses Multinode High Availability (MNHA) on SRX Series Firewalls.

See Understanding ECMP Groups for more details on ECMP based Consistent Hashing.

See "How CSDS Works with TLB" on page 33 for more details on RE-based TLB.

3

CHAPTER

# CSDS Deployment Scenarios and Topologies

# Deployment Scenarios and Topologies

**SUMMARY**

Read this topic to know about the deployment models supported in CSDS Architecture.

**IN THIS SECTION**

## CSDS Deployment Scenarios

Table 2 on page 11 provides details about the deployment models based on the Junos OS devices, load balancing type, and the security services.

**Table 2: CSDS Deployment Scenarios**

| Load Balancing Type | MX Series Deployment Mode | Security Features | SRX Series Firewalls in MNHA Mode | SRX Series Firewalls in Standalone Mode |
|---|---|---|---|---|
| ECMP-based Consistent Hashing | Single MX Series | Carrier-grade NAT, Stateful firewall | No | Yes |
| | | IPsec VPN | No | Yes |
| | Dual MX Series | Carrier-grade NAT, Stateful firewall | Yes | No |
| | | IPsec VPN | No | No |
| Traffic Load Balacer (TLB) | Single MX Series | Carrier-grade NAT, Stateful firewall | Yes | Yes |
| | | IPsec VPN | Yes | Yes |
| | Dual MX Series | Carrier-grade NAT, Stateful firewall | Yes | Yes |

Table 2: CSDS Deployment Scenarios *(Continued)*

| Load Balancing Type | MX Series Deployment Mode | Security Features | SRX Series Firewalls in MNHA Mode | SRX Series Firewalls in Standalone Mode |
|---|---|---|---|---|
| | | IPsec VPN | Yes | Yes |

**Change History Table**

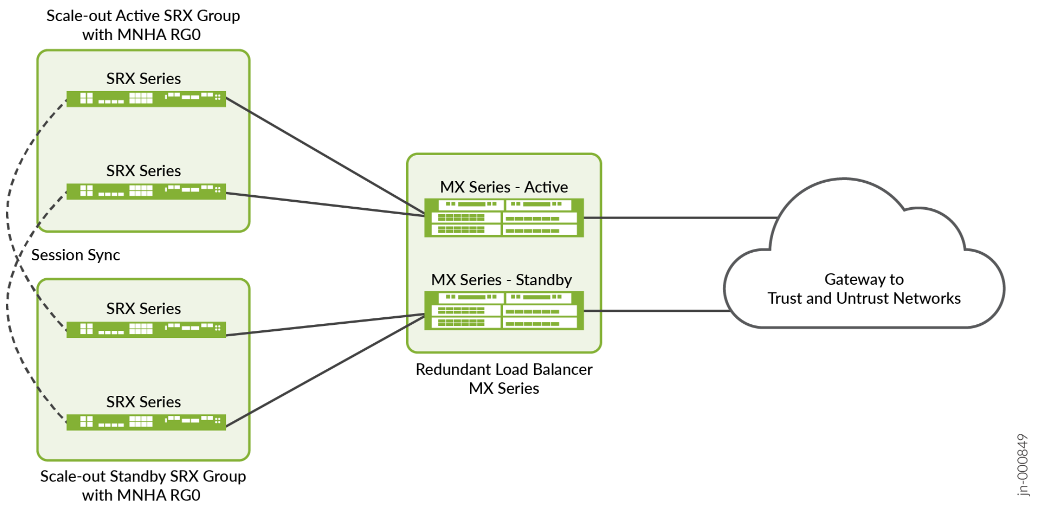
Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

| Release | Description |
|---|---|
| 24.4R1 | TLB support (RE-based TLB) for CSDS is added in Junos OS Release 24.4R1. |
| 23.4R1 | ECMP-based Consistent Hashing support for CSDS is introduced in Junos OS Release 23.4R1. |

## CSDS Dual MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Multinode HA)

Figure 2 on page 13 illustrates the CSDS topology with dual MX Series routers and scaled-out SRX Series Firewalls in Multinode HA mode.

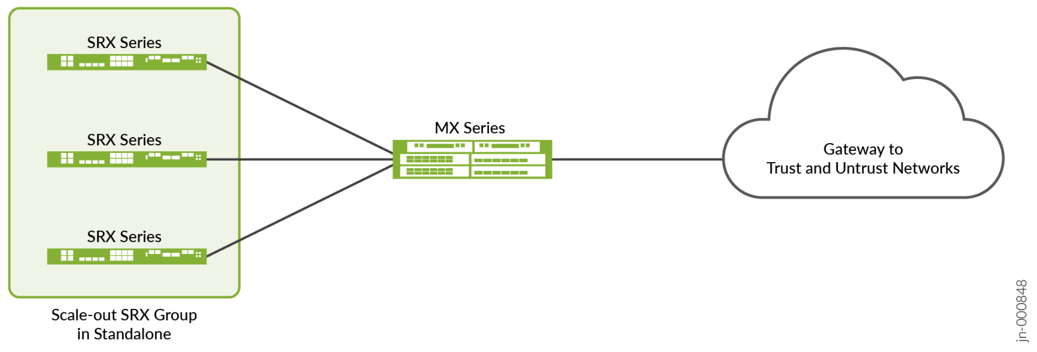
**Figure 2: Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (MNHA)**



## CSDS Single MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)

Figure 3 on page 14 illustrates the CSDS topology with single MX Series router and scaled-out SRX Series Firewalls in standalone mode.

**Figure 3: Single MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)**


SRX Series
SRX Series
SRX Series
Scale-out SRX Group in Standalone
MX Series
Gateway to Trust and Untrust Networks
jn-000848

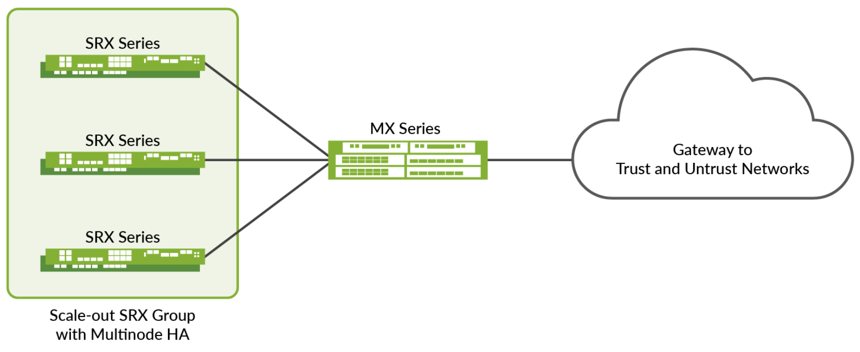
## CSDS Single MX Series (TLB) and Scaled-Out SRX Series Firewalls (Multinode HA)

**SUMMARY**

Learn about CSDS topology with single MX Series routers and scaled-out SRX Series Firewalls in Multinode HA mode.

Figure 4 on page 15 illustrates the CSDS topology with single MX Series routers and scaled-out SRX Series Firewalls in Multinode HA mode. The topology offers redundancy for the SRX Series Firewalls. The MX Series lacks redundancy, even though it has a second Routing Engine (RE).

**Figure 4: Single MX Series (TLB) and Scaled-Out SRX Series Firewalls (MNHA)**


SRX Series
SRX Series
SRX Series
Scale-out SRX Group with Multinode HA
MX Series
Gateway to Trust and Untrust Networks
jn-000847

Multinode HA offers sessions synchronization within a cluster addressing failure scenarios. So the topology provides each SRX Series Firewalls pair with redundancy and scaling although no redundnacy on MX Series except for the dual RE support.

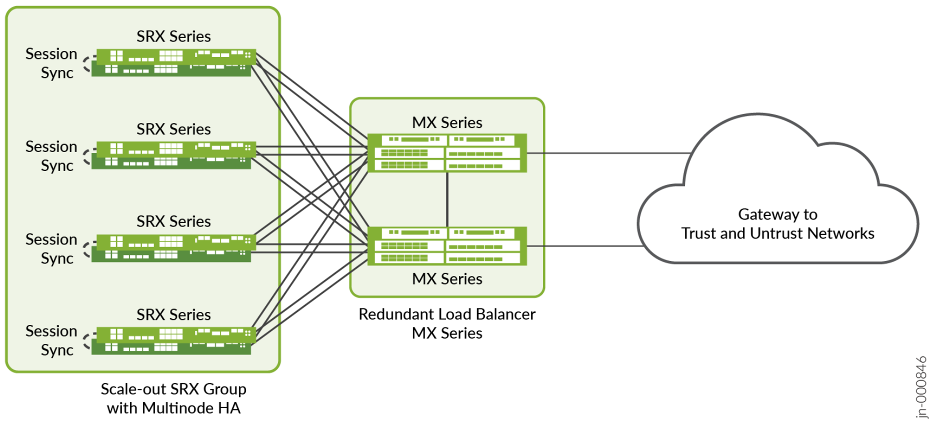
## CSDS Dual MX Series (TLB) and Scaled-Out SRX Series Firewalls (Multinode HA)

**SUMMARY**

Learn about CSDS topology with dual MX Series routers and scaled-out SRX Series Firewalls in Multinode HA mode.

Figure 5 on page 16 illustrates the CSDS topology with dual MX Series routers and scaled-out SRX Series Firewalls in Multinode HA mode. The topology ensures maximum redundancy for both MX Series and SRX Series Firewalls by making use of all nodes at the same time. It safeguards against all potential failover scenarios.

**Figure 5: Dual MX Series (TLB) and Scaled-Out SRX Series Firewalls (MNHA)**



MX Series routers control traffic on both routers. You can deploy SRX Series Firewalls either in active-backup role, or in active-active role making use of both nodes at the same time. This augments the capacity of the network during normal operation. But a failure results in only one active role at a time, considering a single Multinode HA cluster. This topology ensure full redundancy and scaling for MX Series routers and SRX Series Firewalls.

4

CHAPTER

# CSDS and ECMP Based Consistent Hashing

# How CSDS Works with ECMP Based Consistent Hashing

**SUMMARY**

In this topic, you'll learn how CSDS Architecture works with the ECMP based Consistent Hashing load balancer in the MX Series routers.

**IN THIS SECTION**

## What is ECMP and Consistent Hashing?

Equal-cost Multipath (ECMP) is a network routing strategy that allows to load balance traffic of the same session. The traffic in a session with the same source and destination transmits across multiple paths of equal cost.

When forwarding a packet, the routing decides which next-hop path to use. The device considers the packet header fields that identify a flow when determining the next-hop. When using ECMP, the device determines the next-hop paths of equal cost based on the routing metric calculations and hash algorithms. So, routes of equal cost have the same preference and metric values, and the same cost to the network. The ECMP process identifies a set of routers, each of which is a legitimate equal cost next-hop towards the destination. The routes that are identified are referred to as an ECMP set.

Consistent load balancing maintains all active links and instead remaps only those flows affected by one or more link failures. This feature ensures that flows connected to links that remain active continue uninterrupted. This feature applies to topologies where members of an ECMP group are external BGP neighbors in a single-hop BGP session. With the help of Bidirectional Forwarding Detection (BFD) over external BGP, faster link failure detection is possible.

## Benefits

- Increases bandwidth by fully utilizing otherwise unused bandwidth on links to the same destination with ECMP

- Even distribution of the workloads with Consistent Hashing
- Fast response in workload distribution with Consistent Hashing

## How ECMP and Consistent Hashing works in CSDS?

An ECMP set is formed when the routing table contains multiple next-hop addresses for the same destination with equal cost. If there is an ECMP set for the active route, Junos OS uses a hash algorithm to choose one of the next-hop addresses in the ECMP set to install in the forwarding table. You can configure the device so that multiple next-hop entries in an ECMP set are installed in the forwarding table. The Junos OS devices can perform per-packet load balancing to spread traffic across multiple paths between the routing devices.

The CSDS Architecture maintains the symmetricity of the flows in the SRX Series Firewalls. The user data client's (client device) incoming and outgoing traffic always reaches the same SRX Series Firewall (server device) that maintains the state. To reach the same SRX Series Firewall, the MX Series hashes the traffic onto the same link towards that firewall in both the directions.

A user data client is identified by the source IP address in the upstream direction (client-to-server) and the destination IP address in the downstream direction (server-to-client). The MX Series performs symmetric hashing for a given tuple—source IP address and destination IP address. The MX Series calculates the same hash irrespective of the direction of the flow, even if the source and destination IP addresses are swapped. To ensure all flows from a client reach the same SRX Series Firewall, MX Series performs hash only on source IP address (and not destination IP address) in one direction and vice versa in the reverse direction.

By default, when a failure occurs in one or more paths, the hashing algorithm recalculates the next-hop for all paths, typically resulting in the redistribution of all flows. Consistent load balancing with Consistent Hashing enables the MX Series to override this behavior so that only flows for links that are inactive are redirected. All existing active flows are maintained without disruption. When a link fails, redistribution of all flows could result in significant traffic loss to the SRX Series Firewalls that are active. Consistent load balancing maintains all active links and instead remaps only those flows affected by one or more link failures. This feature ensures that flows connected to links that remain active continue uninterrupted.

This feature applies to topologies where members of an ECMP group are external BGP neighbors in a single-hop BGP session. By running BFD over these external BGP neighbors, the MX Series ensures faster link failure detection between the MX Series and the ECMP SRX Series next-hops. Junos OS applies consistent load balancing when you add a new ECMP path or modify an existing path. You can add SRX Series Firewall gracefully with an intent to equally redistribute from each active SRX Series Firewall, causing minimal impact to existing ECMP flows. For example, if there are four active SRX Series Firewalls carrying 25% of total flows on each link and you add another SRX Series Firewall, 5% of flows from each existing SRX Series Firewalls moves to the new SRX Series Firewall. This ensures 20% of flow redistribution from the existing four SRX Series Firewalls to the new firewall. The application might restart the session on the new firewall as the flows don't have a matching session.

## IPsec VPN Traffic Flow in Single MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)

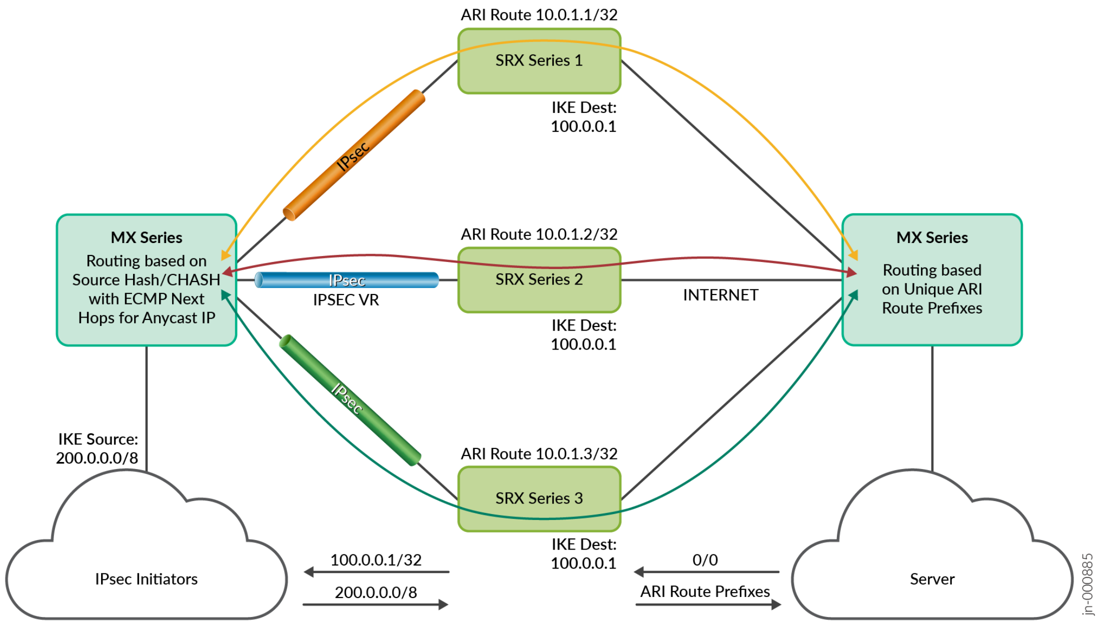
In this topic, you'll see how IPsec traffic flows in a single MX Series with ECMP based Consistent Hashing load balancing with the standalone SRX Series Firewalls.

In this topology, you must:

- Configure a single MX Series with two interfaces for the IPSEC VR and INTERNET routing instances.
- Configure the SRX Series Firewalls with the AutoVPN with the same anycast IP hosted on the loopback interface of the IKE endpoint IP address. Ensure all the SRX Series Firewalls are in IPsec responder only mode.
- The IPsec tunnels that are initiated from the IPsec initiator behind the MX Series use the same SRX Series Firewall IKE endpoint IP address with unique traffic selectors. This SRX Series Firewall uses the same traffic selector to install unique Auto Route Insertion (ARI) routes to attract the data return traffic from the server to the correct IPsec tunnel. ARI automatically inserts a static route for the remote network from the INTERNET side. A route is created based on the remote IP address configured in the traffic selector and is inserted into the IPSEC VR routing instance. See Understanding Auto route Insertion.
- Configure the forwarding table with the load balancing policy with source hash for anycast IP route. The MX Series receives the anycast IP route on IPSEC VR instance and advertises using eBGP to the MX Series on the IPSEC VR side. MX Series imports this routes on the IPSEC VR instance using load balancing consistent hash policy.
- MX Series on the TRUST side has ECMP routes for anycast IP address.

Figure 6 on page 21 illustrates the step-by-step traffic flow.

**Figure 6: IPsec Traffic Flow with Single MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)**



The MX Series is a single router configured with multiple logical interfaces towards scaled-out SRX Series Firewalls on the IPSEC VR and INTERNET VR direction.

1. The IKE traffic initiated from IPsec initiator router reaches the MX Series on IPSEC VR instance and matches the ECMP anycast IP route. The traffic takes one of the ECMP next-hops to SRX Series Firewalls based on the calculated source IP based hash value.

2. The SRX Series Firewalls anchors the IKE session and installs the ARI route. SRX Series Firewalls advertise the ARI route towards the INTERNET instance of the MX Series router.

3. The clients connected to the IPsec initiator router initiates the traffic that passes the IPsec VPN tunnel and reaches the anchored IPsec tunnel on the SRX Series Firewall. The clear text packets out of the IPsec VPN tunnel are routed towards the INTERNET direction to reach the server.

4. The IPsec data reply traffic from the server to the client reaches the MX Series on the INTERNET instance and is routed through the unique ARI route to the SRX Series Firewall where tunnel is anchored.

5. The SRX Series Firewall encrypts and sends the traffic over the tunnel to the IPsec initiator and then to the client.

6. When an SRX Series Firewall is down, Consistent Hashing on the MX Series router ensures that the sessions on the other SRX Series Firewall are not disturbed and only the IPsec sessions on the

impacted SRX Series Firewalls are redistributed and might start a new session. DPD timers ensure traffic redistribution during SRX Series Firewall failure or the link failure.

## NAT Traffic Flow in Single MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)

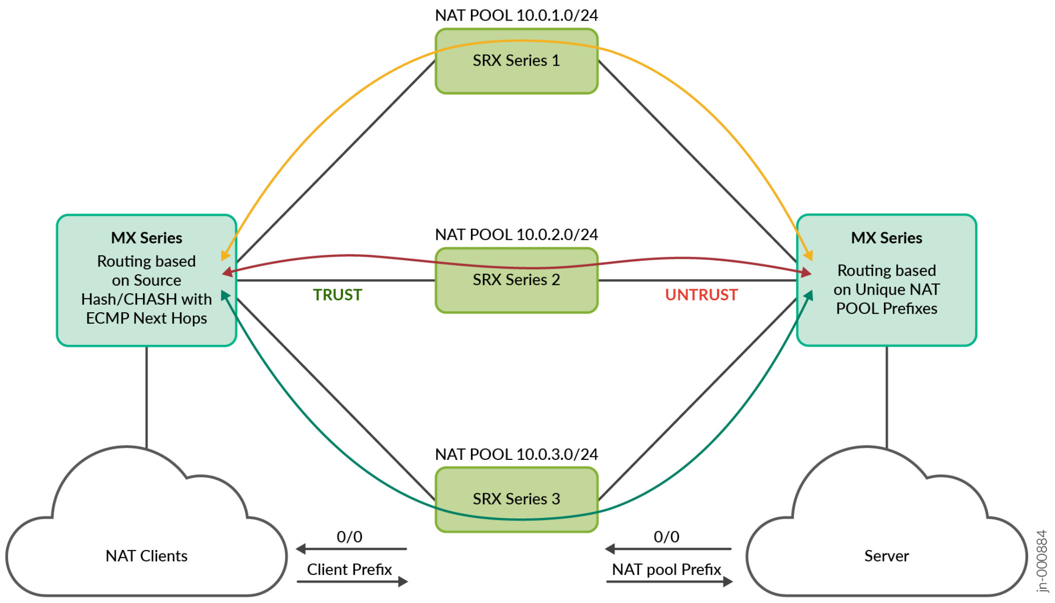
In this topic, you'll see how NAT traffic flows in a single MX Series with ECMP based Consistent Hashing load balancing with the standalone SRX Series Firewalls.

In this topology, you must:

- Configure a single MX Series with two interfaces for the TRUST and UNTRUST routing instances.
- Configure eBGP/BFD on each interface.
- Configure the load balancing policy with source hash for route 0/0 in the forwarding table.
- Configure unique NAT pool IP address range per SRX Series Firewall.

Figure 7 on page 23 illustrates the step-by-step traffic flow.

**Figure 7: NAT Traffic Flow with Single MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)**



The MX Series is a single router that is configured with multiple logical interfaces towards scaled-out SRX Series Firewalls on the TRUST and UNTRUST VR direction.

1. The SRX Series Firewalls receive 0/0 route on the UNTRUST side and advertises using eBGP to the MX Series on the TRUST side. The MX Series imports these routes on the TRUST side using the ECMP based Consistent Hashing policy.

2. The SRX Series Firewalls receive the client prefix route on the TRUST side and advertises the NAT pool route prefix using eBGP to the MX Series on the UNTRUST side.

3. MX Series TRUST side has the ECMP routes for 0/0 route, and the UNTRUST side has unique route for the NAT pool route prefix.

4. The forward traffic flow from client-to-server reaches the MX Series router on the TRUST instance and matches route 0/0 route and takes any one ECMP next-hop to the SRX Series Firewall based on the source IP address hash value.

5. The SRX Series Firewall creates a NAT flow session and routes the packet to the MX Series on the UNTRUST direction towards the server.

6. The reverse traffic flow from server-to-client reaches the MX Series on the UNTRUST instance and matches unique NAT pool prefix route and takes the same SRX Series Firewall where the forward flow is anchored. This ensures symmetricity is maintained in the SRX Series Firewalls.

7. When an SRX Series Firewall is down, Consistent Hashing on the MX Series router ensures that the sessions on the other SRX Series Firewall are not disturbed and only sessions on the impacted SRX Series Firewalls are redistributed. The redistributed sessions get an IP address from different NAT pool for source NAT and hence the application restarts its TCP/UDP session.

## Stateful Firewall Traffic Flow in Single MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)

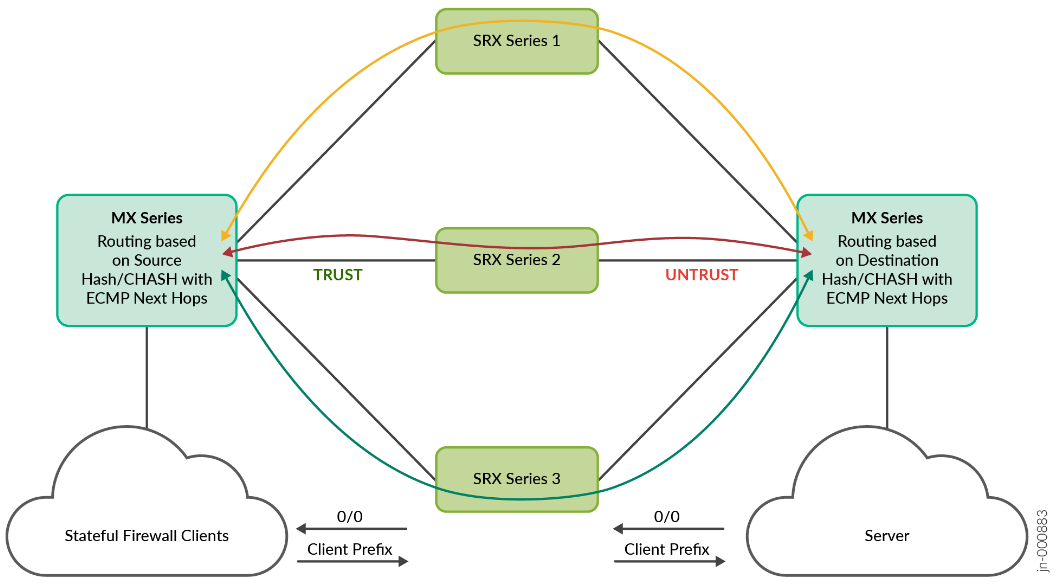
In this topic, you'll see how stateful firewall traffic flows in a single MX Series with ECMP based Consistent Hashing load balancing with the standalone SRX Series Firewalls.

In this topology, you must:

- Configure a single MX Series with two interfaces for the TRUST and UNTRUST routing instances.
- Configure eBGP/BFD on each interface.
- Configure the load balancing policy with source hash for route 0/0 in the forwarding table.
- Configure the load balancing policy with destination hash for client prefixes routes in the forwarding table.

Figure 8 on page 25 illustrates the step-by-step traffic flow.

**Figure 8: Stateful Firewall Traffic Flow with Single MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone)**


SRX Series 1
MX Series
Routing based on Source Hash/CHASH with ECMP Next Hops
SRX Series 2
TRUST
UNTRUST
MX Series
Routing based on Destination Hash/CHASH with ECMP Next Hops
SRX Series 3
Stateful Firewall Clients
0/0
Client Prefix
0/0
Client Prefix
Server
jn-000883

The MX Series is a single router configured with multiple logical interfaces towards scaled-out SRX Series Firewalls on the TRUST and UNTRUST VR direction.

1. The SRX Series Firewalls receive 0/0 route on the UNTRUST side and advertises using eBGP to the MX Series on the TRUST side. The MX Series imports these routes on the TRUST side using the ECMP based Consistent Hashing policy.

2. The SRX Series Firewalls receive the client prefix route on the TRUST side and advertises using eBGP to the MX Series on the UNTRUST side. The MX Series imports these routes on the UNTRUST side using the ECMP based Consistent Hashing policy.

3. MX Series TRUST side has the ECMP routes for 0/0 route, and the UNTRUST side has ECMP routes for the client prefix routes.

4. The forward traffic flow from client-to-server, reaches the MX Series router on the TRUST instance and matches route 0/0 route and takes any one ECMP next-hop to the SRX Series Firewall based on the source IP address hash value.

5. The SRX Series Firewall creates an stateful firewall flow session and routes the packet to the MX Series on the UNTRUST direction towards the server.

6. The reverse traffic flow from server-to-client, reaches the MX Series on the UNTRUST instance and matches client prefix route and takes the same ECMP next-hop based on the calculated destination IP based hash value.

7. The source and destination IP address of the stateful firewall session doesn't change. The calculated hash value remains the same and takes the same ECMP next-hop SRX Series Firewall on the forward and reverse flows. This ensures that the symmetricity is maintained in the SRX Series Firewalls.

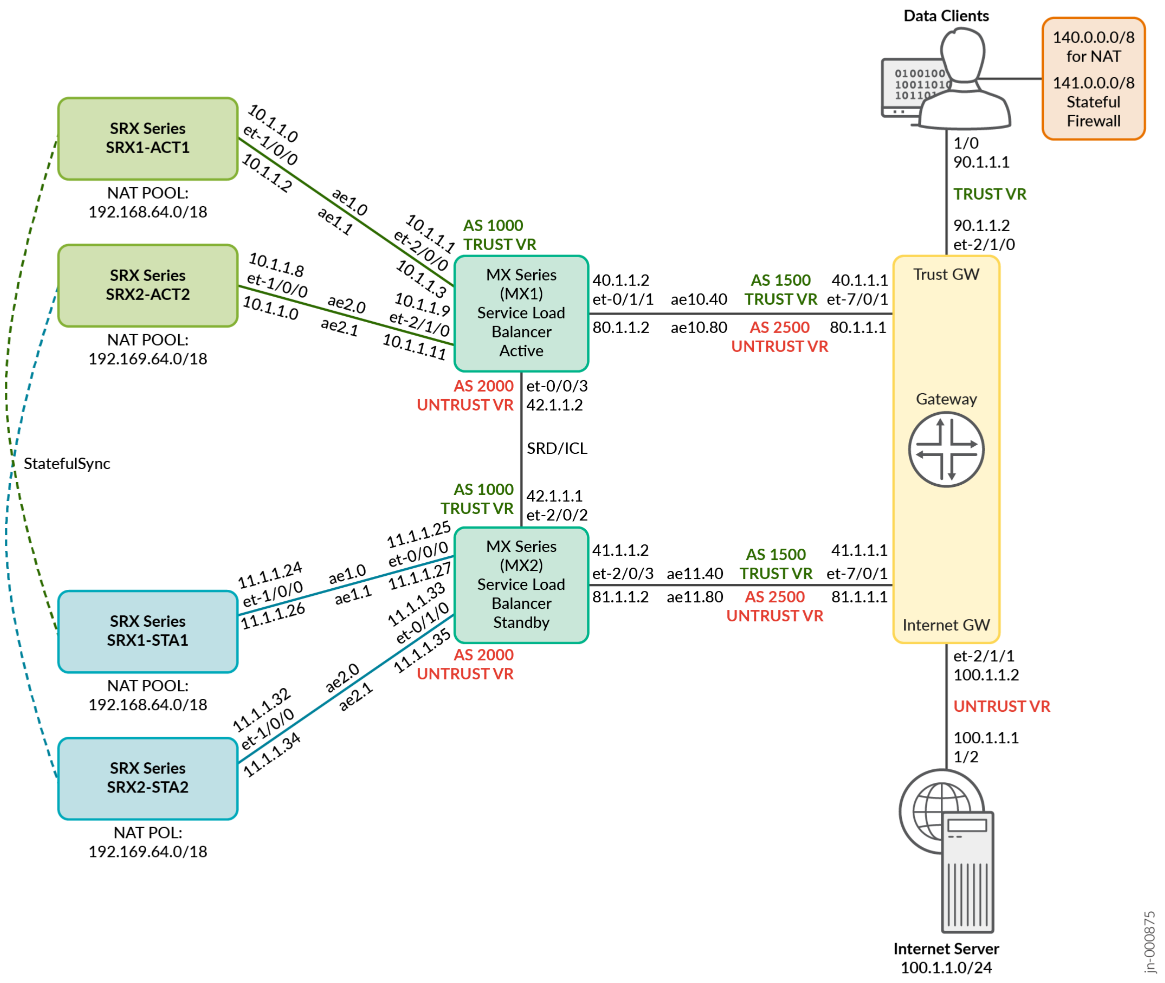
8. When an SRX Series Firewall is down, Consistent Hashing on the MX Series router ensures that the sessions on the other SRX Series Firewall are not disturbed and only sessions on the impacted SRX Series Firewalls are redistributed. But the applications might restart the session as the firewalls are deployed in standalone mode.

## Stateful Firewall and NAT Traffic Flow in Dual MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Multinode HA)

In this topic, you'll see how stateful firewall traffic flows in a dual MX Series with ECMP based Consistent Hashing load balancing with the SRX Series Firewalls in Multinode HA.

Figure 9 on page 27 illustrates the topology for dual MX Series with ECMP based Consistent Hashing and scaled-out SRX Series Firewalls using Multinode HA.

**Figure 9: Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Multinode HA)**



In this topology:

- Configure the MX Series pair in HA with Active/Standby mode.
- Configure the SRX Series Firewalls in Multinode HA pair in Active/Backup mode with session synchronization. SRX1-ACT1 and SRX2-ACT2 are in MNHA pair; and SRX1-STA1 and SRX2-STA2 are in MNHA pair. SRX1-ACT1 and SRX1-STA1 are in stateful synchronization and SRX1-ACT2 and SRX1-STA2 are in stateful synchronization.
- When you deploy SRX Series Firewalls in Multinode HA pair, the session synchronization occurs in both the directions depending on where traffic is received.
- Configure MX Series pair with Service Redundancy Daemon (SRD) redundancy for the user management of MX Series HA pair. See Service Redundancy Daemon.

- MX Series pair monitor links towards TRUST and INTERNET gateway router, and links between MX Series and SRX Series Firewalls. The SRD triggers automatic switchover to the other MX Series if any of the links fail. Failover happens even when the primary MX Series is down.
- MX Series with 4x100G interface connected to the SRX Series Firewalls as an AE bundle contains 3 VLANs (trust, untrust and HA management).
- Primary MX Series remains as the primary ECMP path and the secondary MX Series is the standby ECMP path.
- Use SRD for the MX Series redundancy and control the primary MX Series state transition.
- SRD installs a signal route on the primary MX Series that is used for route advertisement with preference.
- The primary MX Series advertises routes as it is, whereas the standby MX Series advertises routes with as-path-prepend. Expanding an AS path makes a shorter AS path look longer and therefore less preferable to BGP. See Understanding Adding AS Numbers to BGP AS Paths.
- Interfaces on the primary MX Series towards SRX Series Firewall and Secondary MX Series towards SRX Series Firewall must be provisioned using similar interface numbering with similar I/O card (IOC). This helps in maintaining the same unilist next-hop ordering on both the MX Series routers.
- Unilist next-hop ordering is decided by RPD based on the logical interface (ifl) index number (Ascending order of logical interface (ifl) numbers).
- Since unilist next-hop ordering is same in both MX Series routers, there won’t be any issue with hash (source or destination) post any MX Series switchover.

Figure 10 on page 29 illustrates the stateful synchronization in Multinode HA pair. Here, SRX1-ACT1 and SRX1-STA1 are in stateful synchronization and SRX1-ACT2 and SRX1-STA2 are in stateful synchronization.

**Figure 10: Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Multinode HA) with Stateful Synchronization Flow**

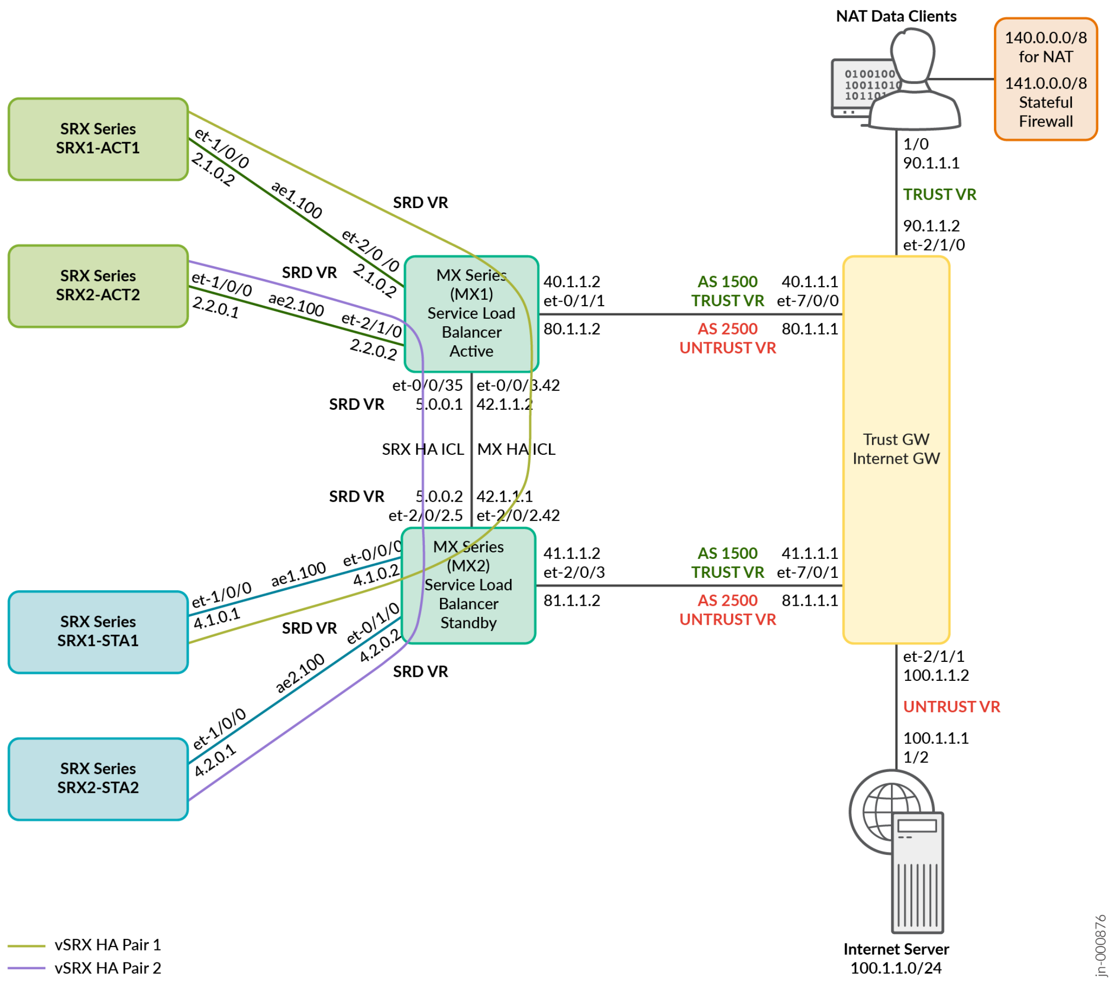


Figure 11 on page 30 illustrates the NAT traffic flow.

**Figure 11: NAT Traffic Flow in Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Multinode HA)**

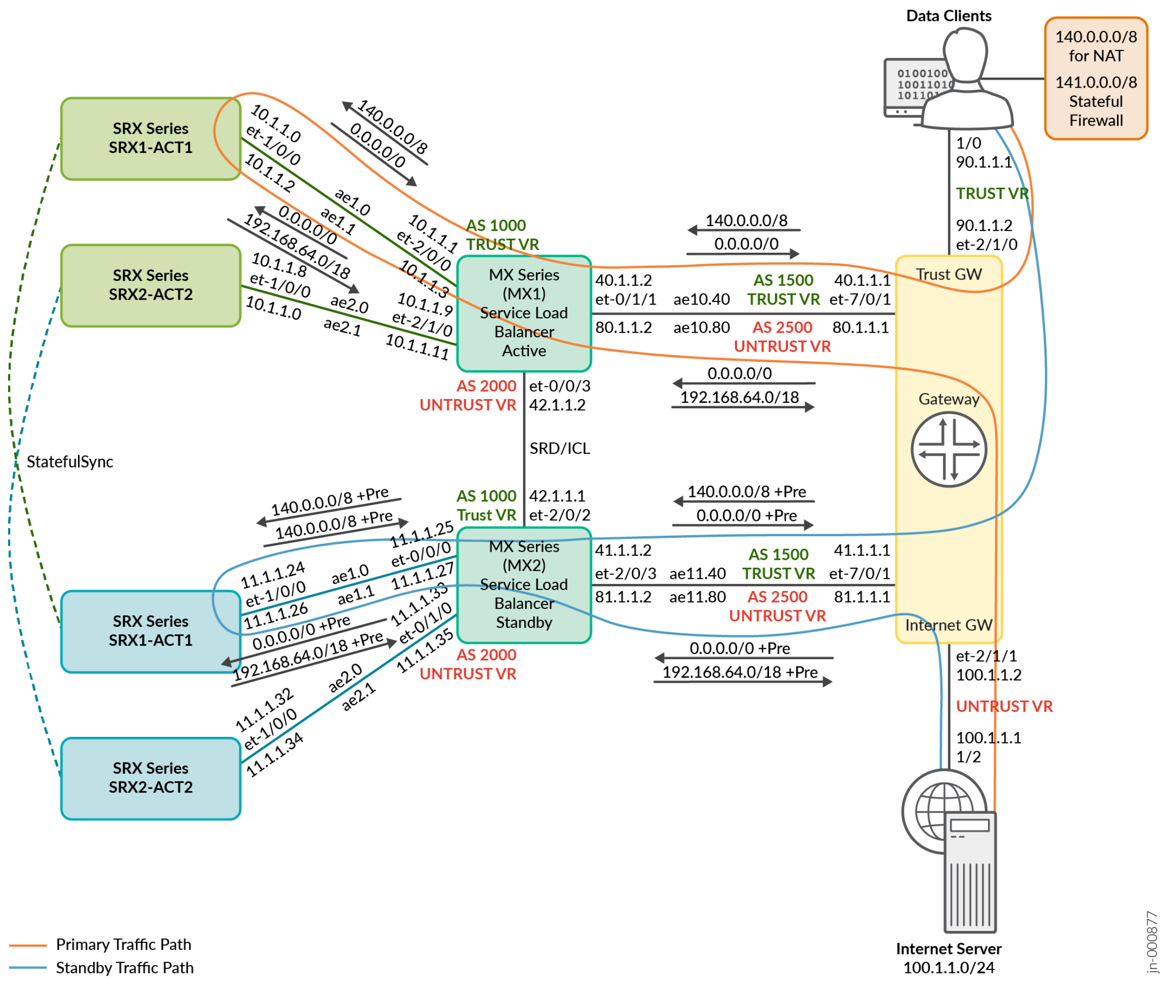
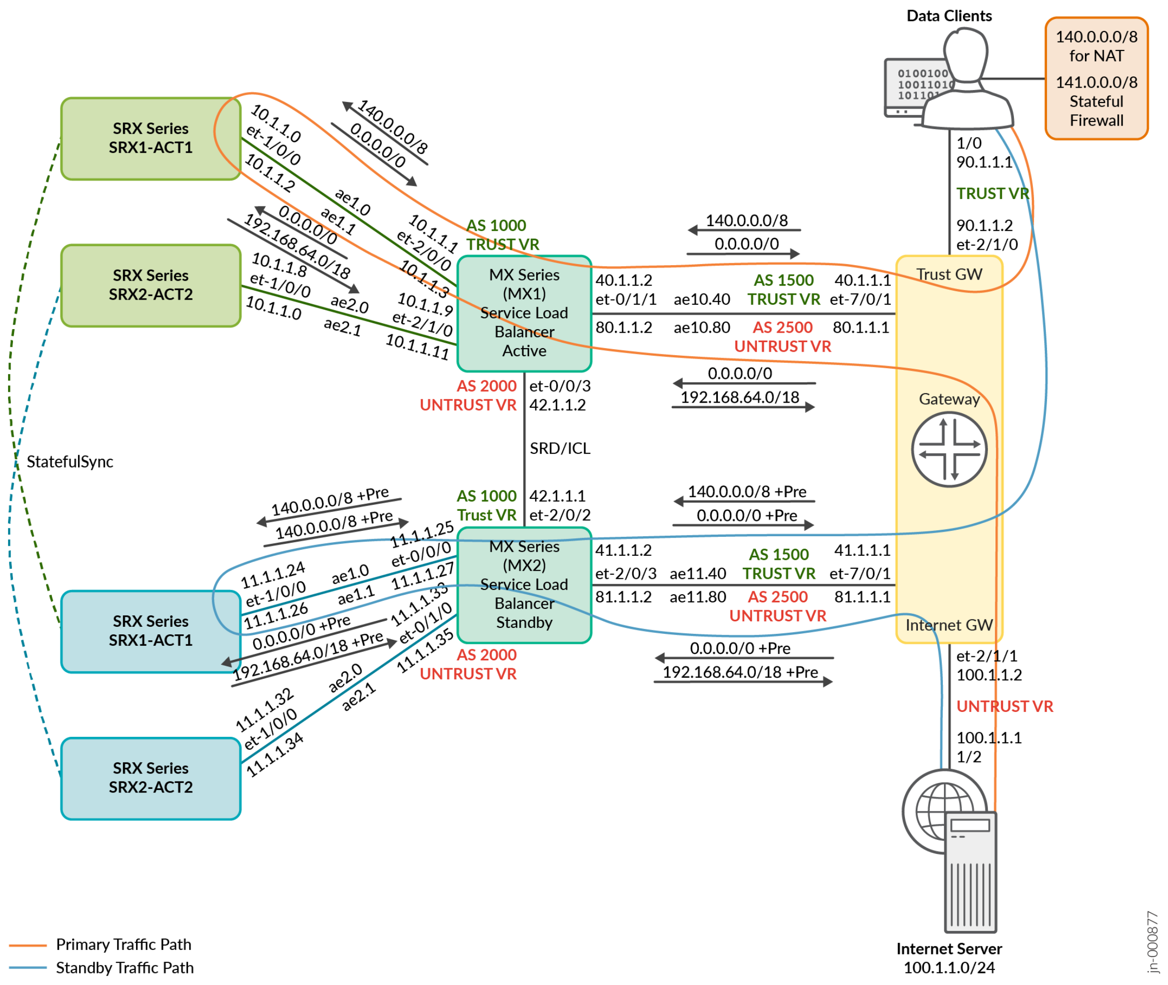


Figure 12 on page 31 illustrates the stateful firewall traffic flow.

**Figure 12: Stateful Firewall Traffic Flow in Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Multinode HA)**



5

CHAPTER

# CSDS and TLB

# How CSDS Works with TLB

**SUMMARY**

In this topic, you'll learn how CSDS Architecture works with the TLB-based load balancer in the MX Series routers.

**IN THIS SECTION**

## What is Traffic Load Balancer in MX Series

Traffic Load Balancer (TLB) provides stateless translated and non-translated traffic load balancing functionality as an inline Packet Forwarding Engine (PFE) service in MX Series routers. Load balancing in this context is a method where the device distributes the incoming transit traffic across the configured servers that are in service.

## Benefits

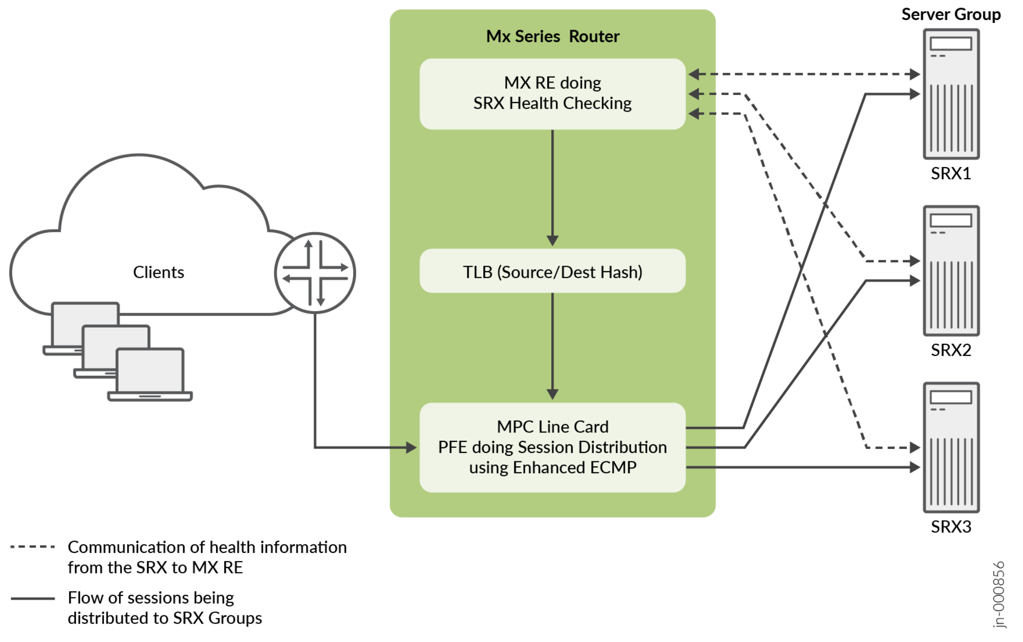
- TLB performs stateless load balancing, which means no state information is created for a connection.
- No scaling limitations.
- Throughput is close to line rate.

## TLB for CSDS

You can operate TLB in the following two modes in MX Series:

- Translated (Layer 3) mode—Do not use this mode, for CSDS solution .
- Non-translated Direct Server Return (Layer 3) mode—Use this mode for scale-out solution.

**Figure 13: TLB in MX Series for CSDS**


Mx Series Router
MX RE doing SRX Health Checking
TLB (Source/Dest Hash)
MPC Line Card PFE doing Session Distribution using Enhanced ECMP
Server Group
SRX1
SRX2
SRX3
Clients
Communication of health information from the SRX to MX RE
Flow of sessions being distributed to SRX Groups
jn-000856

See Figure 13 on page 34 to understand TLB in MX Series. TLB functionality on MX Series includes the following:

- TLB configures a list of available SRX Series Firewalls addresses. The Packet Forwarding Engine (PFE) programs the selector table based on these SRX Series Firewalls.
- TLB performs ICMP based health check for each SRX Series Firewalls using the MX Series Routing Engine (RE). The RE-based health checks support probe types such as ICMP, TCP, UDP, HTTP, and SSL. If health check passes for any firewalls, TLB installs specific IP route or wildcard IP route in the routing table with next-hop as composite next-hop.
- The MX Series programs composite next-hop in the PFE with all available SRX Series Firewalls in the selector table. Filter-based forwarding enables the client to server traffic to be directed to the TLB. The MX Series matches the specific IP route or wildcard IP route to distribute the traffic between the available SRX Series Firewalls using source or destination hash. The server to client is directly routed back to the client, bypassing the TLB.

## RE-Based Health Check for TLB

Traffic Load Balancer (TLB) is enhanced to be able to run the health check process on the Routing Engine (RE) instead of the service-PIC on the next generation MX routers. This feature is applicable for both MSP and USF.

To enable the health-check process (net-monitord) on RE, you can use the command, `set services traffic-load-balance routing-engine-mode`. This configuration and the TLB change ensure that the process responsible for managing and orchestrating traffic distribution and redirection connects to the local instance of the network monitoring process instead of the remote instance running on the service-PIC.

The health-check probe types that are supported on the RE-based health-checks are ICMP, TCP, UDP, HTTP and SSL probes.

Loopback interface is used instead of the service interface for the TLB configuration.

**NOTE**: The interfaces ms-x/y/0 or vms-x/y/0 respectively for MSP and USF are not needed by TLB when net-monitord is running on RE. Replace reference of the ms-x/y/0 or vms-x/y/0 interface with loopback interface lo.x.

```
instance Instance_V4 {
    interface lo0.0;     <<loop back interface instead of ms-interface>>
    client-interface ge-2/2/4.0;
    server-interface ge-2/2/5.0;
    client-vrf client_vrf_1;
    server-vrf server_vrf_1;
    group group1 {
        real-services [ rs_1 rs_2 rs_3 rs_4 rs_5 rs_6 rs_7 rs_8 ];
        routing-instance server_vrf_1;
        health-check-interface-subunit 0;
        network-monitoring-profile nm_prof_icmp;
```

**NOTE**: To enable RE-based TLB, you must configure the routing-engine-mode to enable net-monitord on RE. A validation for the configuration is added and both interface ms-x/y/0.0 or interface vms-x/y/0.0 cannot be configured together in the respective mode of operation, namely, MSP or USF.

RELATED DOCUMENTATION

routing-engine-mode (TLB)

# IPsec VPN Traffic Flow in Single MX Series (TLB) and Scaled-Out SRX Series Firewalls

**SUMMARY**

In this topic, you'll see how IPsec traffic flows in a single MX Series with TLB-based load balancing with SRX Series Firewalls.

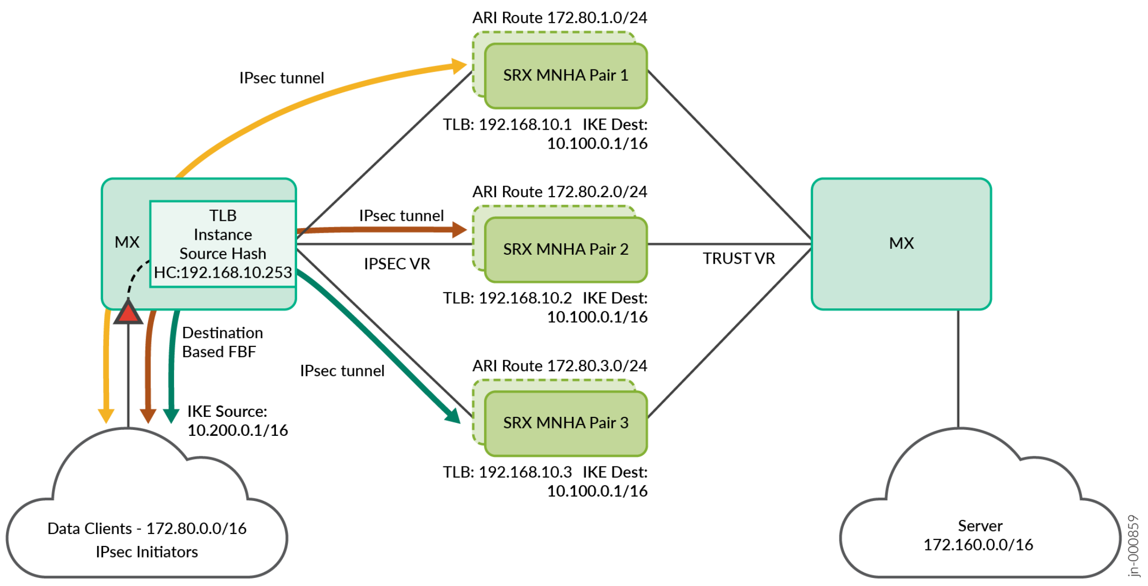
In this topology:

- Configure a single MX Series with two interfaces for the IPSEC VR and INTERNET routing instances. MX Series TLB does health check on all the scaled-out SRX Series Firewalls and builds the next-hop for load balancing the traffic.
- Connect all the scaled-out SRX Series Firewalls to the MX Series with BGP connections.
- Configure the MX Series with TLB on the IPSEC VR routing instance to perform the load balancing of data traffic coming from client-side gateway router towards the scaled-out SRX Series Firewalls.
- Configure unique IKE proposal, IKE policy, IPsec proposal, and IPsec policy per MNHA pair.
- Configure unique IP addresses for all the scaled-out SRX Series Firewalls connected to MX Series that is used by TLB to perform the health check and build up the selector table in the PFE. PFE uses this selector table to load balance the packet across the available next-hops. This health check is reachable through the BGP connection. The anycast IP address used for the IKE endpoint is reachable through this unique IP address on each SRX Series Firewalls.
- Configure all the scaled-out SRX Series Firewalls for AutoVPN with same anycast IP address as the IKE endpoint. All SRX Series Firewalls are in IPsec VPN responder-only mode..
- IPsec VPN clients initiated behind the MX Series use the same SRX Series Firewall's IKE endpoint with unique traffic-selectors. The SRX Series Firewalls use the traffic-selector to install unique Auto Route Insertion (ARI) routes to invite the data return traffic to the right IPsec VPN tunnel from the server.

- Configure the MX Series with TLB on the IPsec VPN VR routing instance to perform the load balancing of IKE traffic coming from MX Series router towards the scaled-out SRX Series Firewalls.

Figure 14 on page 37 illustrates the step-by-step traffic flow.

**Figure 14: IPsec Traffic Flow with Single MX Series (TLB) and SRX Series Firewalls**



The MX Series is a single router configured with multiple logical interfaces towards scaled-out SRX Series Firewalls on the IPSEC VR and INTERNET VR direction.

1. Filter-based forwarding based on IKE destination IP address match is used in MX Series router to push IPsec VPN traffic to the TLB IPsec forwarding instance. TLB forwarding instance includes a default route with next-hop as list of SRX Series Firewalls. TLB installs this default route when the health check passes for at least one SRX Series Firewalls.

2. TLB performs source-based hash load balancing across all the available SRX Series Firewall next-hop devices.

3. Load balanced IPsec VPN tunnel sessions are anchored on any available SRX Series Firewalls and installs the ARI route.

4. The packets get decrypted, and the clear text packets are routed to reach the server through the MX Series over the INTERNET routing instance. MX Series routes the packet to the server.

5. For the return traffic coming from server to client on the MX Series INTERNET routing instance, unique ARI routes are used to route the traffic back to same SRX Series Firewalls where the IPsec VPN tunnel is anchored.

6. SRX Series Firewalls use the same IPsec VPN tunnel session to encrypt the packet and route the IPsec VPN traffic towards MX Series on the IPsec VPN VR direction.

7. MX Series routes the IPsec VPN traffic back to IPsec VPN Initiators.

## NAT Traffic Flow in Single MX Series (TLB) and Scaled-Out SRX Series Firewalls

**SUMMARY**

In this topic, you'll see how NAT traffic flows in a single MX Series with TLB-based load balancing with SRX Series Firewalls.

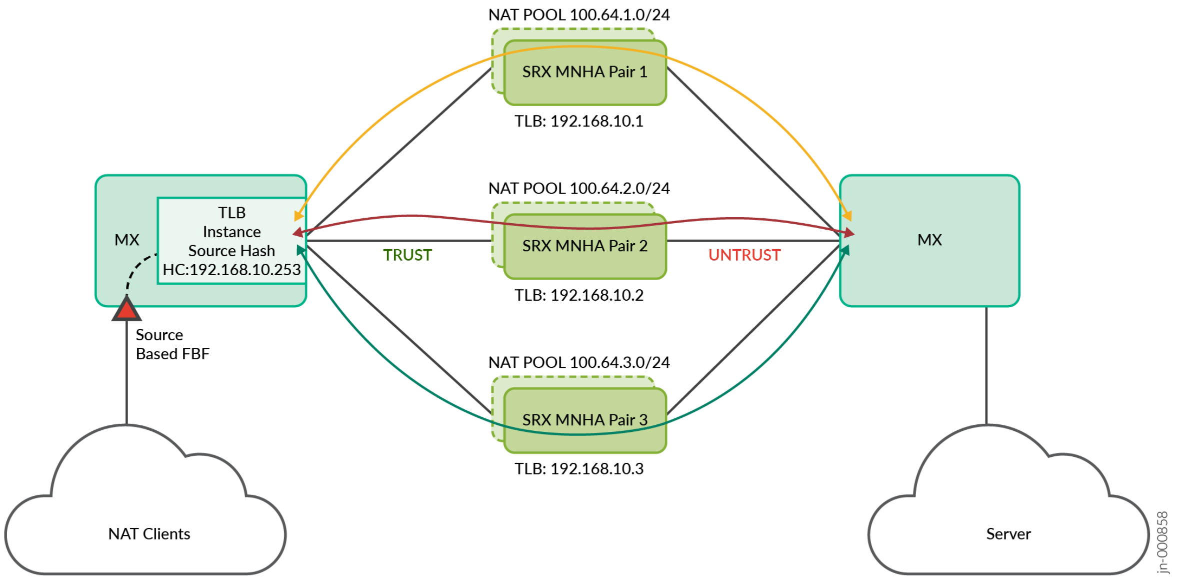
In this topology for NAT traffic:

- Configure a single MX Series with two interfaces logical interfaces (IFL) for TRUST and UNTRUST routing instances. MX Series TLB does health check on all the scaled-out SRX Series Firewalls and builds the next-hop for load balancing the traffic.
- Connect all the scaled-out SRX Series Firewalls to the MX Series with BGP connections.
- Configure the MX Series with TLB on the TRUST routing instance to perform the load balancing of data traffic coming from client-side gateway router towards the scaled-out SRX Series Firewalls.
- Each scaled-out SRX Series Firewalls must have a unique NAT pool range, advertised towards the MX Series UNTRUST direction.
- Configure unique IP addresses for all the scaled-out SRX Series Firewalls connected to MX Series that is used by TLB to perform the health check and build up the selector table in the PFE. PFE uses this selector table to load balance the packet across the available next-hops. This health check is reachable through the BGP connection.
- The filter-based forwarding on source IP address match is used in the MX Series router to push the NAT specific traffic to the TLB TRUST forwarding instance.
- The TLB forwarding instance has a default route with next-hop as list of SRX Series Firewalls. TLB installs this default route when its health check passes with at least one SRX Series Firewalls.

- TLB does source-based hash load balancing across all the available SRX Series Firewall next-hop devices.
- Load balanced NAT data sessions are anchored on any available SRX Series Firewalls and NAT flow gets created. Then it is routed to reach the server through MX Series router over UNTRUST routing instance.
- For the return traffic coming from server to client direction on the MX Series UNTRUST routing instance, unique NAT pool routes are used to route the traffic to the same SRX Series Firewalls.
- The SRX Series Firewalls use same NAT flow to process the return traffic and route the packet towards MX Series Router on the TRUST direction. The MX Series router routes the packet back to the client.

Figure 15 on page 39 illustrates the step-by-step traffic flow.

**Figure 15: NAT Traffic Flow with Single MX Series (TLB) and SRX Series Firewalls**



The MX Series is a single router configured with multiple logical interfaces towards scaled-out SRX Series Firewalls on the TRUST VR and UNTRUST VR direction.

1. For the forward traffic coming from client-to-server, the MX Series router uses filter-based forwarding based on the source IP address match to push the NAT traffic to the TLB TRUST forwarding instance. TLB forwarding instance includes a default route with next-hop as the list of SRX Series Firewalls. TLB installs this default route when the health check passes for at least one SRX Series Firewalls.

2. TLB performs source-based hash load balancing across all the available SRX Series Firewall next-hop devices.
3. Load balanced NAT data sessions are anchored on any available SRX Series Firewalls and NAT flow is created.
4. Then the traffic is routed to reach the server through the MX Series over UNTRUST routing instance.
5. For the return traffic coming from server-to-client on the MX Series UNTRUST routing instance, unique NAT pool routes are used to route the traffic to same SRX Series Firewalls.
6. SRX Series Firewalls use the same NAT flow to process the return traffic and routes the packet towards MX Series on the TRUST direction.
7. MX Series routes the packet back to the client.

## Stateful Firewall Traffic Flow in Single MX Series (TLB) and Scaled-Out SRX Series Firewalls

**SUMMARY**

In this topic, you'll see how stateful firewall traffic flows in a single MX Series with TLB-based load balancing with SRX Series Firewalls.

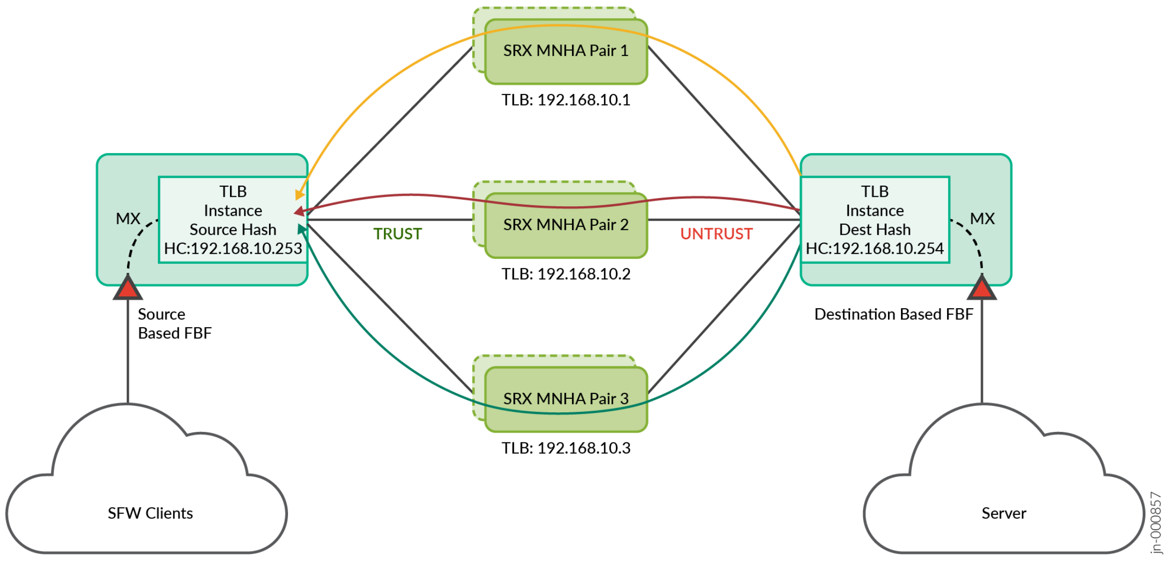
In this topology:

- Configure a single MX Series with two interfaces logical interfaces (IFL) for TRUST and UNTRUST routing instances. MX Series TLB does health check on all the scaled-out SRX Series Firewalls and builds the next-hop for load balancing the traffic.
- Connect all the scaled-out SRX Series Firewalls to the MX Series with BGP connections.
- Configure the MX Series with TLB on the TRUST routing instance to perform the load balancing of data traffic coming from client-side gateway router towards the scaled-out SRX Series Firewalls. For the return traffic coming from the server-side on the MX Series UNTRUST routing instance, another TLB instances is configured on MX Series UNTRUST routing instance.

- Configure unique IP addresses, such as loopback, for all the scaled-out SRX Series Firewalls connected to MX Series that is used by TLB to perform the health check and build up the selector table in the PFE. PFE uses this selector table to load balance the packet across the available next-hops. This health check is reachable through the BGP connection.
- Filter-based forwarding based on source IP address match is used in MX Series router to push stateful firewall traffic to the TLB TRUST forwarding instance.
- TLB forwarding instance has a default route with next-hop as list of SRX Series Firewalls. TLB installs this default route when its health check passes with at least one SRX Series Firewalls.
- TLB does source-based hash load balancing across all the available SRX next-hop devices
- Load balanced stateful firewall data sessions are anchored on any available SRX Series Firewalls and stateful firewall flow gets created. Then it is routed to reach the server through MX Series router over UNTRUST routing instance.
- For the return traffic coming from server to client direction on the MX Series UNTRUST routing instance, another TLB instance is configured on MX Series UNTRUST routing instance to do the load balancing back to the same SRX Series Firewalls.
- Filter-based forwarding of destination IP address match is used in MX Series router to push stateful firewall traffic to the TLB UNTRUST forwarding instance.
- TLB forwarding instance has a default route with next-hop as list of SRX Series Firewalls. TLB installs this default route when its health check passes with at least one SRX Series Firewalls.
- TLB does destination-based hash load balancing across all the available SRX Series Firewall next-hop devices.
- Load balanced stateful firewall data sessions are load balanced to the same SRX Series Firewalls on the return direction and uses the same flow to reach the client through MX Series router over TRUST routing instance.

Figure 16 on page 42 illustrates the step-by-step traffic flow.

**Figure 16: Stateful Firewall Traffic Flow with Single MX Series (TLB) and SRX Series Firewalls**



The MX Series is a single router configured with multiple logical interfaces towards scaled-out SRX Series Firewalls on the TRUST VR and UNTRUST VR direction.

1. For the forward traffic coming from client-to-server, the MX Series router uses filter-based forwarding based on the source IP address match to push the stateful firewall traffic to the TLB TRUST forwarding instance. TLB forwarding instance includes a default route with next-hop as the list of SRX Series Firewalls. TLB installs this default route when the health check passes for at least one SRX Series Firewalls.

2. TLB performs source-based hash load balancing across all the available SRX Series Firewall next-hop devices.

3. Load balanced statfeul firewall data sessions are anchored on any available SRX Series Firewalls and stateful firewall flow is created.

4. Then the traffic is routed to reach the server through the MX Series over UNTRUST routing instance.

5. For the return traffic coming from the server-to-client on the MX Series UNTRUST routing instance, another TLB instances is configured on MX Series UNTRUST routing instance to perform the load balancing back to the same SRX Series Firewalls. The filter-based forwarding based on the destination IP address match is used in MX Series router to push NGFW traffic to the TLB UNTRUST forwarding instance. TLB forwarding instance includes a default route with next-hop as list of SRX Series Firewalls. TLB installs this default route when the health check passes for at least one SRX Series Firewalls.

6. TLB performs destination-based hash load balancing across all the available SRX Series Firewall next-hop devices. Load balanced stateful firewall data sessions are anchored to the same SRX Series Firewalls.

7. SRX Series Firewalls use the same flow to process the return traffic and forwards to MX.

8. MX Series routes the packet back to the client.

6

CHAPTER

# Unified Management with JNU in CSDS

# Junos Node Unifier for CSDS

**SUMMARY**

Learn about Junos Node Unifier (JNU) for Connected Security and Distributed Services (CSDS) architecture and its benefits.

**IN THIS SECTION**

## What is JNU

Juniper Networks Connected Security Distributed Services (CSDS) architecture consists of multiple network devices to support forwarding and security services layers. Juniper Networks Junos Node Unifier (JNU) is a single touch point management solution to effectively manage the network devices in CSDS architecture. JNU simplifies the management and operation of the network devices running Junos operating system (OS). JNU provides unified command line interface (CLI) view of devices in the forwarding and services layers within the CSDS topology.

You can perform the following tasks using JNU:

- Configure and manage the nodes using the Junos OS configuration commands.
- Run the Junos OS operational mode commands.

## Benefits of JNU

- Simplified management—JNU provides a centralized view for managing multiple devices, allowing you to control the entire network infrastructure from a single touch point. This reduces the complexity and effort required to manage individual devices.
- Enhanced visibility—JNU consolidates various management tasks using a single platform streamlining operational workflows. This leads to increased efficiency and enhanced monitoring capabilities.
- Improved security—Centralized management ensures effective implementation and monitoring of security measures, reducing the vulnerabilities and enhancing the overall security posture of the network.

- Scalability—JNU accommodates the increasing number of nodes without much additional effort.

## WHAT's NEXT

Read the following topics to know more about JNU:

- "JNU Topology for CSDS" on page 46
- "JNU Use Cases for CSDS" on page 50
- "Configure Junos Node Unifier for CSDS" on page 189

**Change History Table**

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

| Release | Description |
|---|---|
| 24.4R1 | JNU support for CSDS is introduced in Junos OS Release 24.4R1. |

### RELATED DOCUMENTATION

## JNU Topology for CSDS

### SUMMARY

Learn about Junos Node Unifier (JNU) topology, and its nodes in the Connected Security Distributed Services (CSDS) architecture.

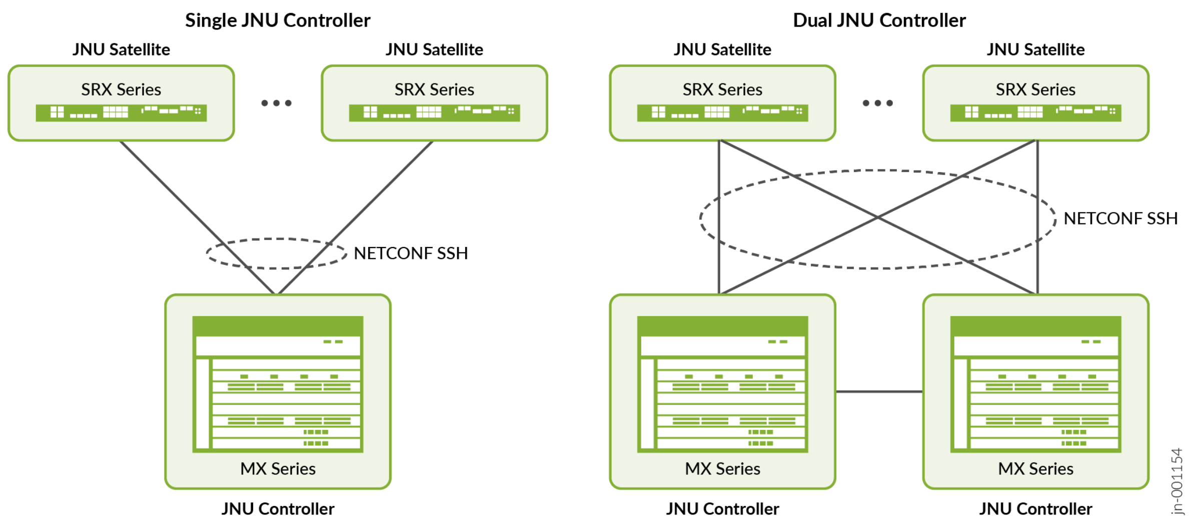
### IN THIS SECTION

Junos Node Unifier (JNU) can manage the network devices in Connected Security Distributed Services (CSDS) architecture using a single touchpoint management solution. It allows you to centrally configure and manage network devices running Junos OS from a single point.

## JNU Nodes

Figure 17 on page 47 shows the JNU topology and its components.

**Figure 17: JNU Topology**



- JNU controller—JNU controller is a centralized entity presenting the unified CLI view of multiple network devices that are added as JNU satellites. This node runs the *jnud* process in controller mode. JNU topology also supports active-active high availability with dual-controller setup. You can use dual controllers in JNU topology.

- JNU satellite—JNU satellites are the physical or virtual network devices that operate under the control of JNU controller. The node runs the *jnud* process in satellite mode.

The connectivity between the JNU nodes uses the CSDS management network, eliminating the need for a separate network. The controller and satellites communicate uses jnuadmin user credentials that gets created during JNU configuration. The communication channel is secure NETCONF over SSH connection. The controller learns satellite's device management schema. Device management schema is a unique data model that is specific to a network device. It describes the complete configuration and operational capabilities of the device. Once JNU is configured, you can access all the satellite's schema from the controller allowing you to centrally manage the nodes.

## JNU Topology Considerations

In a multi-node setup such as the CSDS architecture,

- The MX Series running the *jnud* process acts as the JNU controller. You can use MX Series for JNU controller. We support single touch point management solution with single controller or dual controllers.
- The SRX Series Firewalls, vSRX Virtual Firewalls, and Junos Device Manager (JDM) running the *jnud* process act as JNU satellites.

**NOTE:**

- The external Ubuntu server that hosts the JDM and vSRX Virtual Firewall instances is not part of the JNU topology.
- You must run the same Junos OS release on the controller and the satellite nodes.

Use Feature Explorer to confirm platform and release support for specific features.

## JNU Deployment Process

The following procedure describes JNU deployment process:

1. Configure MX Series as the controller. Note that you must configure the controller role on both the routing engines (RE).
2. Configure SRX Series Firewall, vSRX Virtual Firewall, and JDM container as satellites.
3. When satellites join the controller,
    - Satellites push their schema to the controller during the initial synchronization. The controller also learns the version and model of the satellite as part of the initial synchronization. The satellite has 30 minutes to synchronize with the controller, making 60 attempts at 30 second intervals. If the synchronization fails, you can run the command `request jnu satellite sync` on the satellite, to manually perform initial synchronization.

- The operational command `show chassis jnu satellites` lists all the satellites managed by the controller, including the JDM. Although JDM is added as a satellite, JDM doesn't send its configuration to the controller during the initial synchronization, unlike the other satellites. But you can run JDM specific operational commands using the controller.

- As the controller comes with dual REs, the controller synchronizes the other RE with the schema details.

- In a dual-controller setup, the satellite perform initial synchronization with both the controllers. The satellite fetches the controller IP address from the `[edit chassis jnu-management other-controller` *`controller-ip-address`*`]` hierarchy level and sends the schema to both controllers. If the other controller is unreachable, the commit fails.

- Controller merges the satellite's command hierarchy with its own, but excludes the satellite's configuration schema. The controller and the satellites separately maintain their configuration schemas.

- The controller dynamically learns different versions of the schema that are running on satellites.

**NOTE:**

- The satellites must join the controller without leaving any uncommitted changes on the controller. Avoid running configuration commands during the satellite's joining and upgrade process. Use the operational command `show chassis jnu satellites` to check the status of the satellites, then run the configuration commands.

- Do not run commands directly on the satellites once they join the JNU topology as the configuration may be overwritten by commits from the controller.

- You cannot perform XML subtree filtering of configuration for satellites from the controller.

4. Controller does the subsequent management of satellites. In the controller, you can run the Junos OS commands specific for the network devices that are added as satellites.

## RELATED DOCUMENTATION

Configure Junos Node Unifier for CSDS | 189

# JNU Use Cases for CSDS

**SUMMARY**

Learn about Junos Node Unifier (JNU) use cases and how they fit in the Connected Security Distributed Services (CSDS) architecture.

**IN THIS SECTION**

In JNU topology, JNU satellite communicates with JNU controller using the *jnud* process. Satellite exports its configuration and operational schema to controller. This enables the controller's *jnud* process to present a unified command line interface (CLI) view of both the nodes. Controller is the central point of configuration for all satellites.

In this topic, you'll see JNU use cases and how they work in CSDS.

## Run Configuration CLIs From a Single Touchpoint

You can run Junos OS configurations commands from the controller. Note the following points when you run configuration commands:

- The controller's configuration hierarchy `show chassis jnu-management` lists the names of all satellites in the JNU topology. This command shows the corresponding Junos device model and Junos OS release details. The satellite's configuration hierarchy is available in `[edit chassis satellite` *`satellite-name`*`]` hierarchy level. From this configuration hierarchy, you can run all the satellite-specific configuration commands. Once controller learns the corresponding configuration schema of satellite, the controller uses the configuration schema of that Junos device model and Junos OS release to make configuration changes on that satellite.
- You should not commit in the satellite context. Instead, use `top commit` for the satellite configuration.
- When you perform a satellite-specific configuration commit in the controller, the controller sends commit check to the satellite. One of the following happens:
  - If the commit check succeeds, controller sends commit to the satellite. If the commit check succeeds on the satellite, then the commit also succeeds on the controller. The satellite then

applies the configuration. If the commit fails in any of the satellites, the controller logs the failure, while the other satellites proceed with the commit successfully.

  - If the commit check fails, the controller aborts the commit and rolls back the configuration on the satellites.

- In dual-controller JNU topology, the first controller sends the satellite configuration to the other controller. If the first controller cannot reach the other controller, the first controller's commit fails.

## Run Operational CLIs From a Single Touchpoint

- You can run the satellite-specific operational commands by appending `device-list` *`instance-name`* option to the command. Here *instance-name* is the name of the satellite. See the following sample configuration command:
  - To run an operational command on a single satellite, use `show security ipsec security-associations -device-list 10.10.10.8`.
  - To run an operational command on multiple satellites, use `show security ipsec security-associations -device-list [10.10.10.8 10.10.10.19 10.10.10.15]`.
- The controller propagates the operational command to the satellite. The satellite runs the operational command and returns the output to be displayed on the controller.

**NOTE**: The controller's configuration mode does not support the satellite's operational mode commands. You need to run the satellite's operational commands in the controller's operational mode.

## Upgrade Devices in JNU Topology

You need to upgrade the nodes in JNU topology locally. Adhere to the following sequence when you're upgrading Junos OS on the nodes in JNU topology:

**Table 3: Upgrade Sequence in JNU Topology**

| Deployment Scenario | Upgrade Sequence |
|---|---|
| Single MX Series and standalone SRX Series | 1. Upgrade the controller:<br>• Upgrade MX Series, including both the routing engines (REs).<br>2. Upgrade the satellite:<br>• Upgrade the SRX Series Firewall.<br>or<br>• If you have vSRX Virtual Firewall,<br>a. Upgrade JDM.<br>b. Upgrade vSRX Virtual Firewall. |
| Dual MX Series and standalone SRX Series | 1. Upgrade both the controllers:<br>• Upgrade MX1, including both the REs.<br>• Upgrade MX2, including both the REs.<br>2. Upgrade the satellite:<br>• Upgrade the SRX Series Firewall.<br>or<br>• If you have vSRX Virtual Firewall,<br>a. Upgrade JDM.<br>b. Upgrade vSRX Virtual Firewall. |

**Table 3: Upgrade Sequence in JNU Topology** ***(Continued)***

| Deployment Scenario | Upgrade Sequence |
|---|---|
| Single MX Series and SRX Series in Multinode High Availability | **1.** Upgrade the controller:<br>• Upgrade MX Series, including both the REs.<br>**2.** Upgrade the satellites:<br>• Upgrade the SRX Series Firewalls.<br>**a.** Upgrade SRX node 1 in Multinode High Availability.<br>**b.** Upgrade SRX node 2 in Multinode High Availability.<br>or<br>• If you have vSRX Virtual Firewalls,<br>**a.** Upgrade JDM.<br>**b.** Upgrade vSRX node 1 in Multinode High Availability.<br>**c.** Upgrade vSRX node 2 in Multinode High Availability. |

**Table 3: Upgrade Sequence in JNU Topology *(Continued)***

| Deployment Scenario | Upgrade Sequence |
|---|---|
| Dual MX Series and SRX Series in Multinode High Availability | 1. Upgrade both the controllers:<br>• Upgrade MX1, including both the REs.<br>• Upgrade MX2, including both the REs.<br>2. Upgrade the satellites:<br>• Upgrade the SRX Series Firewalls.<br>a. Upgrade SRX node 1 in Multinode High Availability.<br>b. Upgrade SRX node 2 in Multinode High Availability.<br>or<br>• If you have vSRX Virtual Firewalls,<br>a. Upgrade JDM.<br>b. Upgrade vSRX node 1 in Multinode High Availability.<br>c. Upgrade vSRX node 2 in Multinode High Availability. |

Consider the following points when upgrading the nodes in JNU topology:

- To upgrade MX Series, see Junos® OS Software Installation and Upgrade Guide.

  When you upgrade both the controllers, ensure to upgrade both the REs. After the controller upgrade, you may notice warning message as shown below if you run a configuration command even before the satellites are upgraded. You must not run Junos configuration commands until the satellites are upgraded.

  ```
  user@mx1# exit
  Exiting configuration mode
  warning: schema /var/db/jnu-controller-schema.db not found
  error: Command remap failed
  ```

  Ignore any warning messages that you might notice during the commit operations until the satellites are also upgraded.

After the controller upgrades, the device removes JNU-specific merged controller schema. The device recreates the JNU controller merged schema after the satellite upgrade and join operations. Until then you cannot run the satellite-specific operational commands from the controller.

- If you have JDM, see "JDM and vSRX Virtual Firewall Upgrade Process" on page 71.
- To upgrade the standalone SRX Series Firewall, see Installing Software on SRX Series Firewalls.

  To upgrade both the firewalls in Multinode High Availability, see Software Upgrade in Multinode High Availability
- After the upgrade, ensure that all the nodes are in the same version. Two SRX Series Firewalls cannot have different version. The same applies to MX Series.

To downgrade Junos OS, follows the same sequence.

For more details on JDM and vSRX Virtual Firewall downgrade process, see "JDM and vSRX Virtual Firewall Deletion Process" on page 72.

## RELATED DOCUMENTATION

*jnu-management*

*request jnu satellite sync*

*show chassis jnu satellite*

7

CHAPTER

# vSRX Orchestration with JDM in CSDS

# Junos Device Manager for CSDS

**SUMMARY**

Learn about Junos Device Manager (JDM) for Connected Security and Distributed Services (CSDS) architecture and its benefits.

**IN THIS SECTION**

## What is JDM

Junos Device Manager (JDM) is a Linux container that provides virtualized root file system in a Junos OS-like CLI environment. The CSDS solution uses JDM to orchestrate vSRX Virtual Firewalls on baremetal server with Ubuntu Operating System (OS). If your CSDS services layer only includes physical SRX Series Firewalls, you don't need JDM.

## Benefits of JDM

- Leverages open source software—Uses the existing open source Ubuntu root file system that serves as the foundation for JDM making adaptation easier.
- Simplified orchestration capabilities—Provides simple Junos-like CLI interface to configure and spawn vSRX Virtual Firewalls ensuring ease of use.
- Unified management—Integrates into Junos Node Unifier (JNU) topology as JNU Satellite, making easier manageability with single touchpoint JNU unified CLI management solution.
- Optimized resource allocation—Fetches baremetal server capabilities and allocates resources optimally to virtual instances enhancing system performance.

## WHAT's NEXT

Read the following topics to know more about JDM:

- "JDM Components for CSDS" on page 58
- "Understand vSRX Orchestration with JDM for CSDS" on page 65
- "Install and Configure Junos Device Manager for CSDS" on page 209

**Change History Table**

Feature support is determined by the platform and release you are using. Use Feature Explorer to determine if a feature is supported on your platform.

| Release | Description |
|---|---|
| 24.4R1 | JDM support for CSDS is introduced in Junos OS Release 24.4R1. |

RELATED DOCUMENTATION

CSDS Overview | 2

Junos Node Unifier for CSDS | 45

# JDM Components for CSDS

**SUMMARY**

Learn about Junos Device Manger (JDM) components, and their requirements in Connected Security Distributed Services (CSDS) architecture.

**IN THIS SECTION**

In Connected Security Distributed Services (CSDS) solution, Junos Device Manager (JDM) performs services layer orchestration of vSRX Virtual Firewalls on baremetal servers running Ubuntu Operating System (OS). JDM interacts with the libvirtd process for virtual machine (VM) lifecycle management.

## JDM Infrastructure

Figure 18 on page 59 illustrates the JDM infrastructure for CSDS architecture.

**Figure 18: JDM Infrastructure for CSDS**

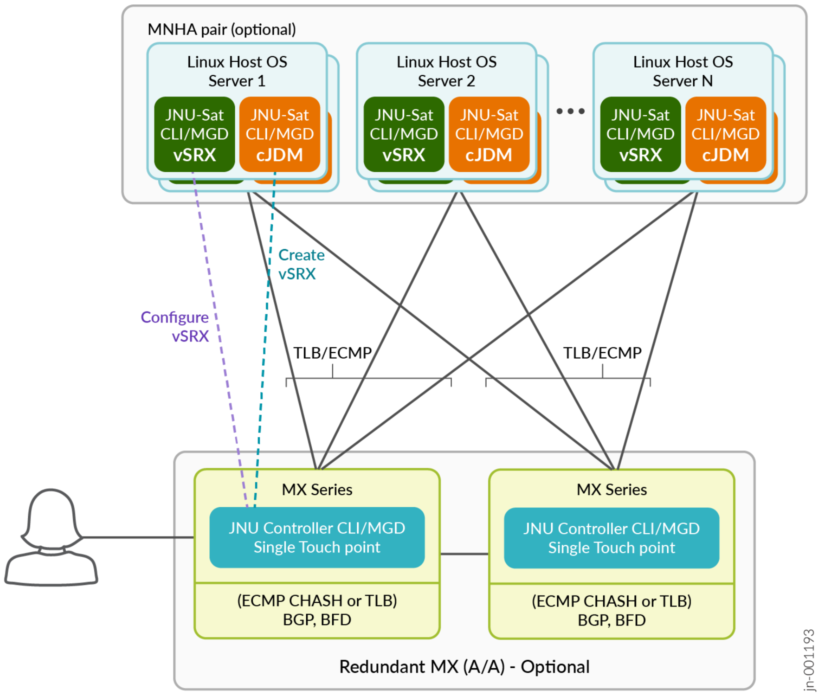


Table 4 on page 59 lists the components in JDM infrastructure.

**Table 4: Components in JDM Infrastructure**

| JDM Component | Description |
|---|---|
| Ubuntu Host Server | It's a baremetal server for hosting JDM container and vSRX Virtual Firewalls spawned by JDM. |
| JDM Package | You install the software package on baremetal Ubuntu host server for running the JDM software. |

**Table 4: Components in JDM Infrastructure *(Continued)***

| JDM Component | Description |
|---|---|
| JDM Processes | JDM container runs the following processes to carry out key tasks:<br>• *mgd*—Helps JDM to present a Junos like CLI and configuration commit model.<br>• *jdmd*—Interfaces with libvirtd, and handles VM lifecycle management.<br>• *jnud*—Provides unified user experience by running in either satellite or controller modes. |
| JDM Container | It's a Linux Container (LXC) that runs in the host server to perform vSRX orchestration. |
| vSRX Image | It's an image used to spawn vSRX Virtual Firewalls. |
| vSRX Virtual Firewalls | They run in Ubuntu host server and play the role of CSDS services layer components for running security services. |

## Host Software Requirements for CSDS

The host that runs Ubuntu OS is a baremetal server with specifications outlined in this section.

Table 5 on page 60 lists the software requirement specifications for host server for running JDM container and vSRX Virtual Firewalls.

**Table 5: Host Software Specifications**

| Software Components | Specifications |
|---|---|
| OS support | Ubuntu 22.04.4 LTS<br><br>Ensure that you contact your vendor partner for the update and troubleshooting support of the host OS. |
| qemu-system-x86 | 6.2.0 (Debian 1:6.2+dfsg-2ubuntu6.16) |

**Table 5: Host Software Specifications *(Continued)***

| Software Components | Specifications |
|---|---|
| libvirt-daemon-system | 8.0.0-1ubuntu7.8 |
| bridge-utils | 1.7-1ubuntu3 |
| xml2 | 0.5-4build1 for amd64 |
| libvirt-clients | 8.0.0-1ubuntu7.8 |
| Hyperthreading | Enabled in BIOS |

## Host Hardware Requirements for CSDS

You can use the baremetal servers litsed in Table 6 on page 61. The table lists the hardware requirement specifications of the host for running JDM container and vSRX Virtual Firewalls. Additionally, the host also has a management interface. You have the option to utilize multiple baremetal servers simultaneously. Each host server runs the JDM, and the JDM actively spawns vSRX Virtual Firewalls on the host.

**Table 6: Host Hardware Specifications**

| Server Profile | Intel Xeon Gold 6438N 2 GHz | Intel Platinum 8571N 2.4 GHz | AMD EPYC 8534P 2.3 GHz | AMD EPYC 9554P 3.10 GHz | AMD EPYC 9754 2.25 GHz |
|---|---|---|---|---|---|
| Cores | 32 | 52 | 64 | 64 | 128 |
| NUMA nodes | 1 | 1 | 8 | 1 | 8 |
| Cores/NUMA | 32 | 52 | 8 | 64 | 16 |

**Table 6: Host Hardware Specifications *(Continued)***

| Server Profile | Intel Xeon Gold 6438N 2 GHz | Intel Platinum 8571N 2.4 GHz | AMD EPYC 8534P 2.3 GHz | AMD EPYC 9554P 3.10 GHz | AMD EPYC 9754 2.25 GHz |
|---|---|---|---|---|---|
| Memory GB | 256 | 256 | 512 | 512 | 1024 |
| Management NICs | NetXtreme-E Series BCM57504 | NetXtreme-E Series BCM57504 | NetXtreme-E Series BCM57504 | Intel Ethernet Server Adapter I210 | NetXtreme-E Series BCM57504 |
| Data NICs | 2 x 200 G Mellanox ConnectX Adapter | 2 x 200 G Mellanox ConnectX Adapter | 2 x 200 G Mellanox ConnectX Adapter | 2 x 200 G Mellanox ConnectX Adapter | 400 G or 4 x 100 G Mellanox ConnectX Adapter |
| Storage Size TB | 1 | 1 | 2 | 1 | 2 |
| vSRX Virtual Firewalls support | 1 vSRX Virtual Firewall utilizing 31 cores and 128 GB RAM | 1 vSRX Virtual Firewall utilizing 51 cores and 128 GB RAM | 7 vSRX Virtual Firewalls utilizing 9 cores and 64 GB RAM | 7 vSRX Virtual Firewalls utilizing 9 cores and 64 GB RAM | 7 vSRX Virtual Firewalls utilizing 18 cores and 128 GB RAM |

You must ensure that the hardware profile matches the specifications, including the processor's model and network cards, to prevent JDM installation failure. You can also use the following minimum supported hardware requirements:

- Processor—3rd Generation Intel Xeon processor and above, or 4th Generation AMD processor and above
- Network adapter—MT2910 Family (ConnectX-7), or MT2892 Family (ConnectX-6 Dx)

## CSDS Services Plane Resource Distribution in the Host

The vSRX Virtual Firewalls support listed in Table 6 on page 61 is based on the CSDS services plane resource distribution. Based on the server hardware specification, the system allocates compute, storage and network resources for JDM container and vSRX Series Virtual Firewalls.

See Table 7 on page 63, Table 8 on page 63, Table 9 on page 64, Table 10 on page 64, and Table 11 on page 65 to see the CSDS services plane resource distribution for the host server, JDM, and vSRX Virtual Firewalls. When JDM spawns the vSRX Virtual Firewalls, you'll notice the resource allocation is based on the following tables.

**Table 7: CSDS Services Plane Resource Distribution on Intel Gold Processor**

| Feature | Host | JDM | vSRX1 |
|---|---|---|---|
| Cores | 0 | 0 | 1 to 31 |
| NUMA | 0 | 0 | 0 |
| Memory GB | 32 | 2 (Shared with host) | 224 |
| Network | - | - | 2 x 200 GB |
| Storage GB | 128 | 5 (Shared with host) | 896 GB |

**Table 8: CSDS Services Plane Resource Distribution on Intel Platinum Processor**

| Feature | Host | JDM | vSRX1 |
|---|---|---|---|
| Cores | 0 | 0 | 1 to 51 |
| NUMA | 0 | 0 | 0 |
| Memory GB | 32 | 2 (Shared with host) | 224 |
| Network | - | - | 2 x 200 GB |
| Storage GB | 128 | 5 (Shared with host) | 896 GB |

**Table 9: CSDS Services Plane Resource Distribution on AMD EPYC 8534P Processor**

| Feature | Host | JDM | vSRX1 | vSRX2 | vSRX3 | vSRX4 | vSRX5 | vSRX6 | vSRX7 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Cores | 0 | 0 | 1, 8 to 15 | 2, 16 to 23 | 3, 24 to 31 | 4, 32 to 39 | 5, 40 to 47 | 6, 48 to 55 | 7, 56 to 63 | 64 |
| NUMA | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | - |
| Memory GB | 64 | 2 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 512 |
| Network | - | - | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | - |
| Storage GB | 144 | 5 | 272 | 272 | 272 | 272 | 272 | 272 | 272 | 2048 |

**Table 10: CSDS Services Plane Resource Distribution on AMD EPYC 9554P Processor**

| Feature | Host | JDM | vSRX1 | vSRX2 | vSRX3 | vSRX4 | vSRX5 | vSRX6 | vSRX7 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Cores | 0 | 0 | 1, 8 to 15 | 2, 16 to 23 | 3, 24 to 31 | 4, 32 to 39 | 5, 40 to 47 | 6, 48 to 55 | 7, 56 to 63 | 64 |
| NUMA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - |
| Memory GB | 64 | 2 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 512 |
| Network | - | - | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | - |
| Storage GB | 128 | 5 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 1024 |

**Table 11: CSDS Services Plane Resource Distribution on AMD EPYC 9754 Processor**

| Feature | Host | JDM | vSRX1 | vSRX2 | vSRX3 | vSRX4 | vSRX5 | vSRX6 | vSRX7 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Cores | 0, 1 | 0, 1 | 2-3, 16-31 | 4-5, 32-47 | 6-7, 48-63 | 8-9, 64-79 | 10-11, 80-95 | 12-13, 96-111 | 14-15, 112-127 | 128 |
| NUMA | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | - |
| Memory GB | 128 | 2 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 1024 |
| Network | - | - | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | 2 VFs | - |
| Storage GB | 144 | 5 | 272 | 272 | 272 | 272 | 272 | 272 | 272 | 2048 |

RELATED DOCUMENTATION

Understand vSRX Orchestration with JDM for CSDS | 65

# Understand vSRX Orchestration with JDM for CSDS

**SUMMARY**

In this topic, you'll learn about using Junos Device Manager (JDM) for vSRX orchestration and how you can manage JDM with Junos Node Unifier (JNU) in Connected Security Distributed Services (CSDS) architecture.

**IN THIS SECTION**

You can orchestrate vSRX Virtual Firewalls using Junos Device Manager (JDM). Ensure that the Ubuntu baremetal host meets the necessary hardware and software requirements. See "JDM Components for CSDS" on page 58.

Before installing and configuring JDM, read this topic to understand and prepare your environment. You'll see details about:

- JDM package download and JDM software installation on the host server.
- Managing JDM from JNU controller
- Configuring JDM for vSRX orchestration

Familiarize yourself with the JDM package, the installation process, and the changes that will occur on the Ubuntu host server as part of the installation.

## JDM Installation Requirements

If your CSDS architecture services plane includes vSRX Virtual Firewall, you need JDM. We recommend that you use JNU controller to centrally manage JDM, which is the JNU satellite in JNU topology. If you already manage the SRX Series Firewall satellites in your CSDS architecture using the *jnud* process, you can add JDM and vSRX Virtual Firewall as satellites to the MX Series controller. If not, you must configure JNU before configuring JDM.

Familiarize yourself with the specific JDM package for CSDS. You can download the JDM software package (e.g., `csds-jdm-jdm-<release-number>.x86_64.deb`) from the Juniper Networks Downloads page at https://support.juniper.net/support/downloads/. Save the package on your MX Series router that acts as the JNU controller. During JDM installation, the MX Series router copies the JDM package to the Ubuntu host for installation of the JDM LXC container that runs the JDM software.

You can install or uninstall JDM from the controller, provided that the SSH keys are exchanged between the MX Series and the Ubuntu host server.

To know about the JDM installation process, see "Host Configuration Changes Post JDM Installation" on page 67 and "JDM Installation Workflow for vSRX Orchestration" on page 70. See "Install and Configure Junos Device Manager for CSDS" on page 209 to follow the step-by-step procedure for JDM installation and configuration.

## Host Configuration Changes Post JDM Installation

In this section, you'll see the modifications that the JDM software makes to the host server as part of the JDM installation. Make sure you understand these modifications before you install JDM.

### OS Configuration Changes

When installing JDM, the software modifies the OS configuration on the host using GRUB. See Table 12 on page 67 for the list of changes.

**Table 12: OS Configuration Changes on Host After JDM Software Installation**

| Description | Expected Configuration |
|---|---|
| CPU Isolation *Isolcpus* | Reserves socket ID 0 (*socket.id 0*) and core ID 0 (*core id 0*) for JDM, host server and emulator Pin |
| Hugepages Size | 1 G |
| Hugepages Count | 1 G x 16 |
| Input–Output Memory Management Unit (IOMMU) | *intel_iommu=on* on Intel server<br><br>*amd_iommu=on* on AMD server |
| Iommu passthrough | *iommu=pt* on Intel server<br><br>*amd_iommu=pt* on AMD server |
| AppArmour | Disabled |

### Directory Structure

JDM software installation creates a new directory, `/juniper`. This directory contains all the JDM binaries, installation packages and temporary files created during installation.

### Cgroups Configuration

JDM LXC container uses cgroups to limit its resources footprint on the host. JDM limits its primary memory to 2 GB. It changes the following controller groups for the machine.slice settings:

- Cpuset
- Memset

### Network Interfaces

JDM installation creates the required JDM network interfaces. Table 13 on page 68 shows the list of network interfaces created after JDM installation.

**Table 13: Network Interfaces After JDM Software Installation**

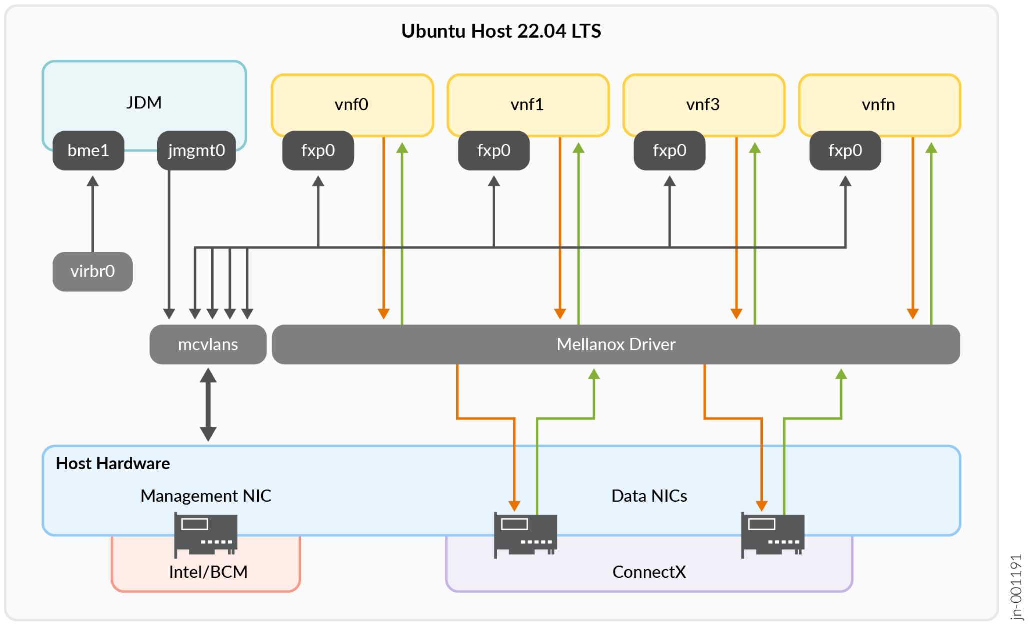
| Interface Type | Interface Name | Interface Description |
|---|---|---|
| JDM Management Interface | *jmgmt0* | • Connects to external facing customer's management network.<br>• Connects host server's physical management interface.<br>• In CSDS, JNU controller communicates with JDM satellite over this interface. |
| vSRX Management Interface | *fxp0* | • Macvlan-based bridged network interface.<br>• Colocates with jmgmt0 over host server's physical management interface.<br>• In CSDS, JNU controller communicates with vSRX satellite over this interface. |

**Table 13: Network Interfaces After JDM Software Installation *(Continued)***

| Interface Type | Interface Name | Interface Description |
|---|---|---|
| vSRX Datapath Interface | SR-IOV VFs *vf-1, vf-2* | • Service plane datapath is over SR-IOV VFs for the firewall's inbound and outbound traffic.<br>• JDM creates these interfaces over host server's physical interfaces.<br>• Creates two virtual functions (VFs) per vSRX Virtual Firewall with each VF over a separate physical function (PF).<br>• Host with two NUMA nodes spawn's two vSRX Virtual Firewalls. JDM creates four VFs over four PFs. Each PF has separate PCIe address. |

Figure 19 on page 69 illustrates the internal networking in the host server.

**Figure 19: Internal Networking in the Ubuntu Host Server**



## JDM Installation Workflow for vSRX Orchestration

The procedure below outlines the installation and configuration of JDM for vSRX orchestration:

1. Ensure all your nodes have a management IP address. Ensure you've downloaded the JDM image and the vSRX Virtual Firewall image in MX Series JNU controller.
2. Add the management IP address of the Ubuntu host server in the MX Series for JDM management. Every host server is referred with a `node-instance` ID.
3. Set up the MX Series as the JNU controller. If you have already setup JNU controller, skip this step.
4. Install JDM from the JNU controller. During installation, the system will:
   a. Create the Linux known_hosts file in JDM for both root and jnuadmin user accounts to store the host's public keys that the users access.
   b. Set up two-way SSH keys between the MX Series and JDM for the jnuadmin user.
   c. Provision the JNU controller IP address in JDM.
   d. Assign an IP address for the JNU satellite in JDM.
   e. Launch JDM. JDM uses the management interface of the host server. During the installation of JDM, MX Series allocates a pool of IP addresses to JDM. JDM designates the first IP address from the pool to itself. The installation process also provisions JDM with controller and satellite IP addresses for jnud process.
5. Spawn vSRX Virtual Firewall instances from the JNU controller. During this process, the system will:
   a. Copy the vSRX Virtual Firewall installer image from the MX Series to JDM.
   b. Use the baseline configuration to initiate the vSRX Virtual Firewall instance in JDM.
   c. Copy the SSH keypair to the home directories of the firewall's root and jnuadmin user accounts from JDM.
   d. Send the second IP in the IP-prefix-range for spawning the firewall from JDM. The system assigns the IP addresses in sequence order based on the number of instances it spawns.
   e. Apply the baseline configuration, that includes the controller IP address, the controller's public SSH key, and the *jnud* process-specific settings. The MX controller adds its jnuadmin user’s public SSH key to the firewall. This allows the JNU controller to securely manage the firewall once it has an IP address. The controller also establishes a one-way trust relationship with the firewall.
   f. Wait for approximately 10 minutes for the instances to spawn. JDM spawns the instances based on the server hardware you choose for the Ubuntu host. Based on server configuration, JDM assigns a pair of 100 or 200 GB interfaces for the firewall's SR-IOV VFs.

6. With one-way SSH key access to the firewall granted to the jnuadmin user on MX Series, the JNU controller retrieves the public key from the firewall and adds it to the local authorized_keys file of the MX Series. The MX Series uses the IP address of the firewall and starts the *jund* process to make the firewall a satellite node.

7. Wait for approximately 10 minutes for the satellites to synchronize with the controller.

The names of all satellites are listed in `[edit chassis jnu-management]` hierarchy level. The satellite schema for a particular satellite is available in `[edit chassis satellites` *`satellite-name`*`]` hierarchy level.

For step-by-step configuration, see "Install and Configure Junos Device Manager for CSDS" on page 209

## JDM and vSRX Virtual Firewall Upgrade Process

Follow the sequence below to upgrade JDM and vSRX Virtual Firewall.

1. Upgrade both the controllers including both the Routing Engines (REs). Follow the Junos OS upgrade process. See Junos® OS Software Installation and Upgrade Guide.

2. Upgrade JDM from the controller using the following command.

   ```
   user@mx1> request csds jdm add csds-instance-id csds-instance-id image jdm-image-path
   ```

3. Upgrade the vSRX Virtual Firewalls using the following commands to copy the image and SSH login to run the Junos OS installation command.

   ```
   user@mx1:~# file copy source-image-path root@satellite-ip-address:image-path-at-destination
   ```

   or

   ```
   user@mx1:~# scp -O source-image-path root@satellite-ip-address:image-path-at-destination
   ```

   and

   ```
   user@vsrx1> request system software add package-name
   ```

Follow the same sequence to downgrade. Ensure to use the correct Junos OS image for the downgrade process.

## JDM and vSRX Virtual Firewall Deletion Process

### Delete vSRX Virtual Firewall Without Deleting JDM

1. Delete vSRX Virtual Firewall from the controller using the following command. Note that this command doesn't delete the JDM.

```
user@mx1> request csds delete-vsx csds-instance-id csds-instance-id
```

### Delete JDM

1. Use the following command to delete an existing JDM instance.

```
user@mx1> request csds jdm delete csds-instance-id csds-instance-id
```

**CAUTION**: Proceed with caution when using the command `request csds jdm delete csds-instance-id csds-instance-id` to delete a JDM instance. The command also removes any vSRX Virtual Firewall instances that the JDM has created. As a results, you must reinstall JDM and vSRX Virtual Firewalls. Ensure that you do not manually delete JDM from the host. Doing so may cause JDM or vSRX Virtual Firewall deployments to fail due to known-host entries on the controller.

To retain vSRX Virtual Firewall, perform only JDM add operation to either upgrade or downgrade JDM. See "JDM and vSRX Virtual Firewall Upgrade Process" on page 71.

RELATED DOCUMENTATION

Install and Configure Junos Device Manager for CSDS | **209**

*csds*

*jnu-management*

*request csds add-vsrx*

*request csds authenticate-host*

*request csds delete-vsrx*

*request csds extract-vsrx-keys*

*request csds jdm*

*request csds sync-controller*

*request jnu satellite sync*

*show chassis jnu satellite*

8

CHAPTER

# Configure CSDS

# Example: Single MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone) for IPsec VPN

SUMMARY

In this configuration, you'll learn to setup a single MX Series with scaled-out SRX Series Firewalls in standalone for IPsec VPN services.

IN THIS SECTION

## Overview

Table 14 on page 75 shows the deployment components used in the example.

**Table 14: Deployment Details**

| CSDS Components | Details |
|---|---|
| Forwarding Layer | MX304 with Junos OS Release 23.4R1 or later |
| Services Layer | vSRX 3.0 with Junos OS Release 23.4R1 or later |
| Redundancy | Single MX Series with ECMP based Consistent Hashing for load balancer.<br><br>SRX Series Firewalls (Standalone) |
| Features | IPsec VPN |
| Additional Component | IPsec initiator device – MX router with SPC3 card. You can use any IPsec initiator device. |

See Table 15 on page 76 and Table 16 on page 76 for traffic flow and VPN details.

**Table 15: Traffic Flows for IPsec VPN**

| Feature | Traffic Flow Component | IP Address |
|---|---|---|
| IPsec VPN on SRX1, SRX2 and SRX3 | IKE Gateway Source (IPsec Initiator) | 200.0.0.0/8 |
| | IKE Gateway Destination (IPsec Responder) | 100.0.0.1/32 |
| | IPsec Data Source | 6.0.0.0/8 |
| | IPsec Data Destination | 75.0.0.0/8 |

**Table 16: IPsec VPN Details**

| Device | IKE Gateways | IPsec Data Endpoints |
|---|---|---|
| SRX1 | 200.0.0.1 and 100.0.0.1 | Tunnel 1 between 6.0.0.3 and 75.0.0.3 |
| SRX2 | 200.0.0.2 and 100.0.0.1 | Tunnel 2 between 6.0.0.2 and 75.0.0.2 |
| SRX3 | 200.0.0.6 and 100.0.0.1 | Tunnel 3 between 6.0.0.1 and 75.0.0.1 |

See Table 17 on page 76 for traffic flow.

**Table 17: Load Balancer to SRX Series Firewalls for IPsec VPN Services**

| Flow Type | Traffic Flow Component | IP Address |
|---|---|---|
| IKE Initiator to SRX | Source Load Balancer (Route Filter on MX) | 100.0.0.1/32 |
| IPsec VPN forward Flow | Routing-Based | |
| IPsec VPN Reverse Flow | Routing-Based | Unique ARI route per SRX |

## Topology Illustration

**Figure 20: Single MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls for IPsec VPN Services**

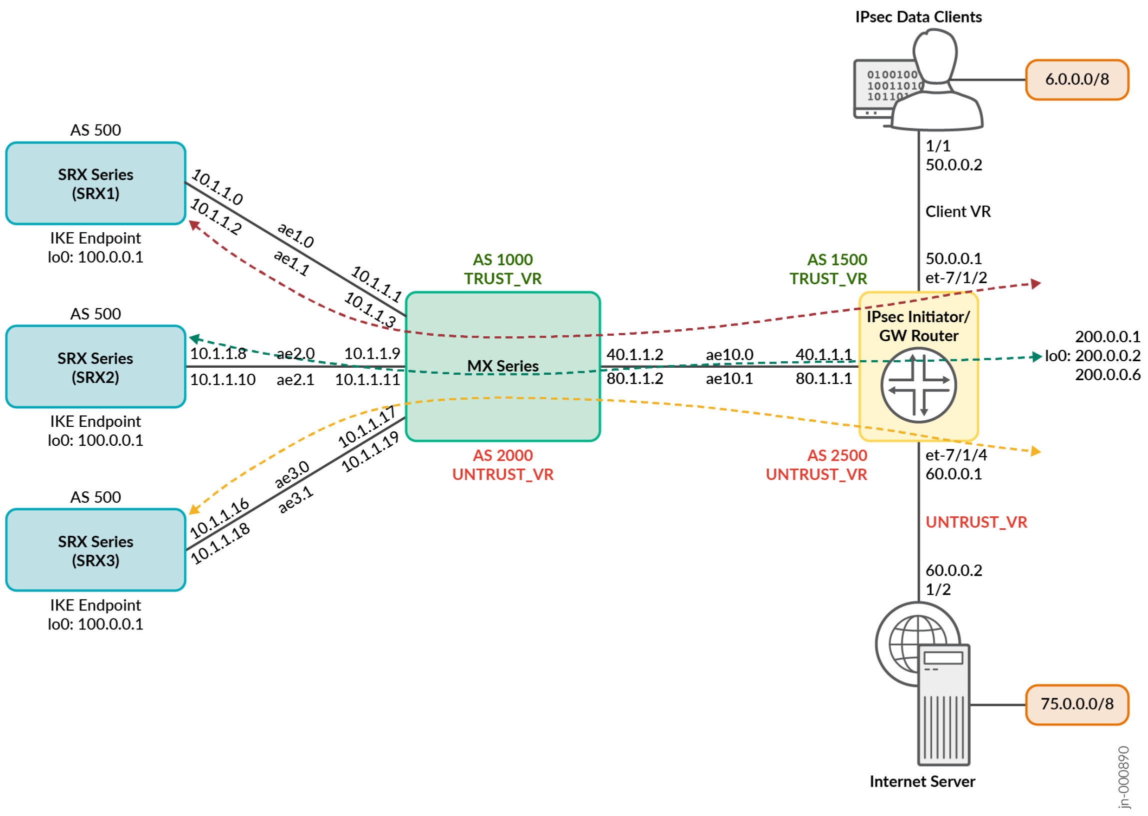


**Figure 21: Route Advertisements in IKE Gateway for IPsec VPN Services**

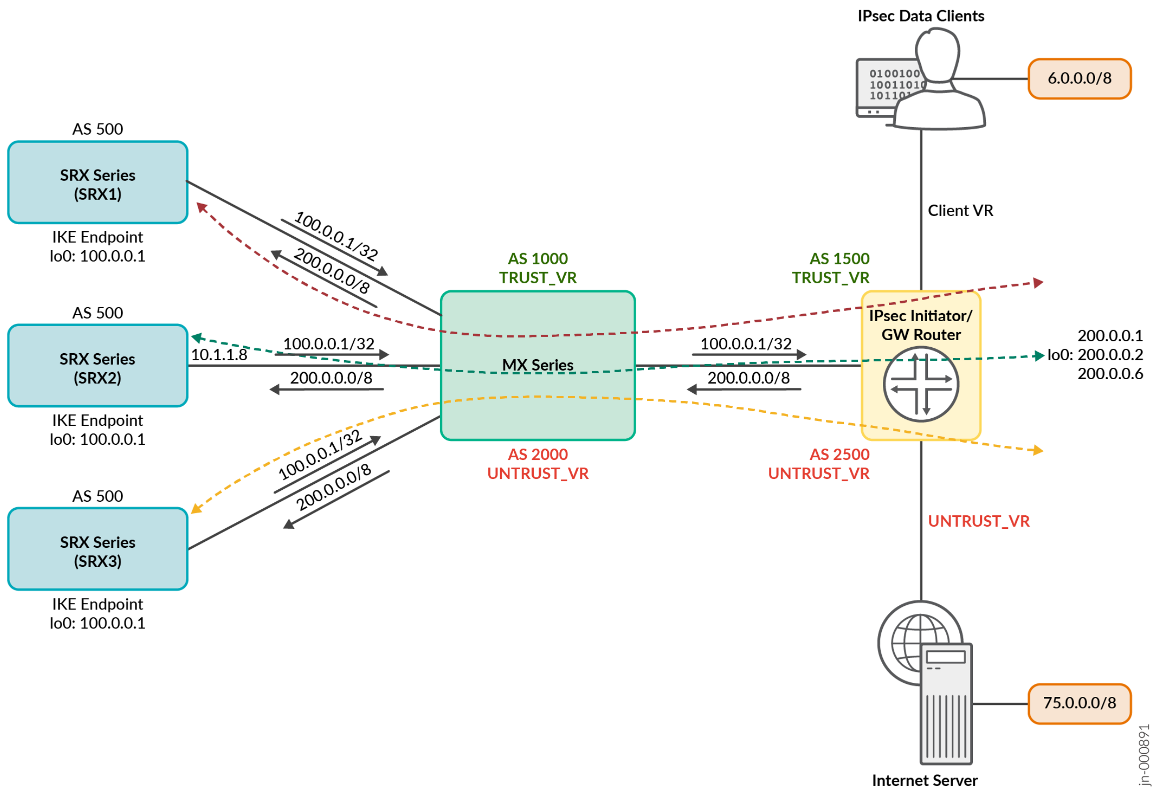
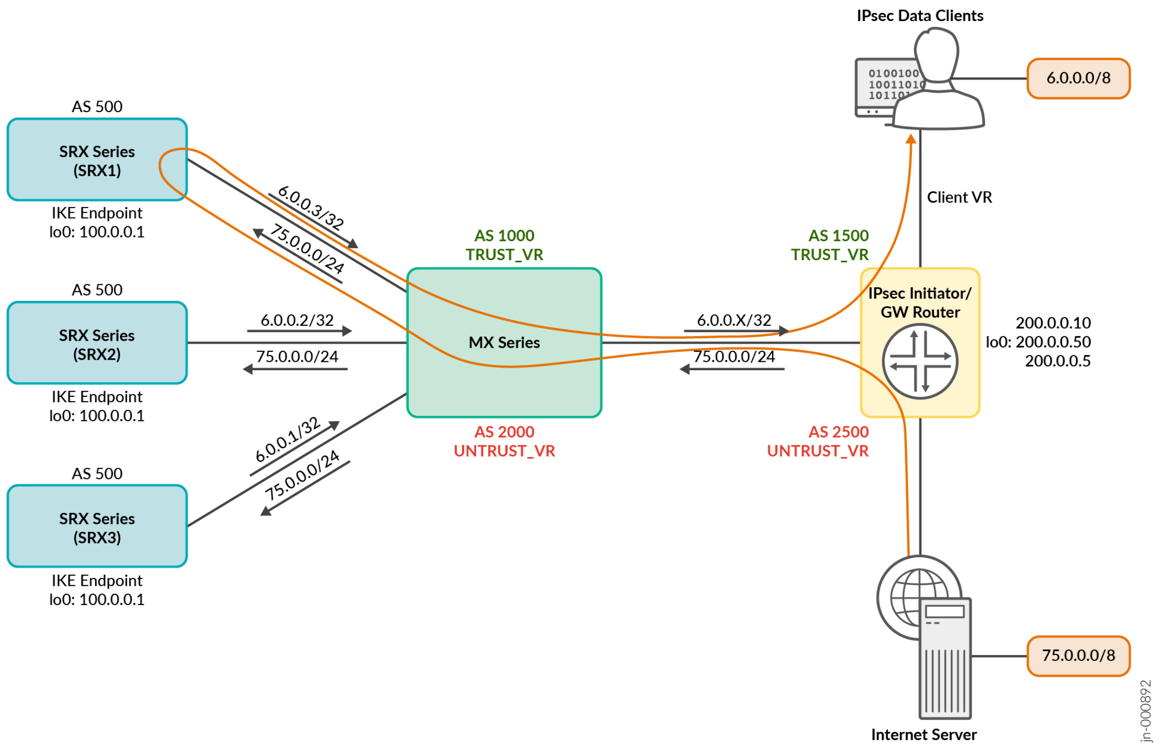


**Figure 22: Route Advertisements for IPsec Endpoint for IPsec VPN Services**



## Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

These configurations are captured from a lab environment and are provided for reference only. Actual configurations might vary based on the specific requirements of your environment.

The following items show a list of configuration components for this example:

- Configure MX Series
- Configure the IPsec Initiator
- Configure SRX1
- Configure SRX2
- Configure SRX3

## Configure MX Series

```
[edit]
set interfaces et-0/0/0 gigether-options 802.3ad ae1
set interfaces et-0/0/1 gigether-options 802.3ad ae2
set interfaces et-0/0/2 gigether-options 802.3ad ae3
set interfaces et-0/0/7 gigether-options 802.3ad ae1
set interfaces et-0/0/8 gigether-options 802.3ad ae2
set interfaces et-0/0/9 gigether-options 802.3ad ae3
set interfaces et-0/0/10 gigether-options 802.3ad ae10
set interfaces et-0/0/11 gigether-options 802.3ad ae10
set interfaces et-0/1/0 gigether-options 802.3ad ae10
set interfaces et-0/1/1 gigether-options 802.3ad ae10
set interfaces et-0/1/2 gigether-options 802.3ad ae10
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 1
set interfaces ae1 unit 0 family inet address 10.1.1.1/31
set interfaces ae1 unit 0 family inet6 address 10:1:1::1/127
set interfaces ae1 unit 1 vlan-id 2
set interfaces ae1 unit 1 family inet address 10.1.1.3/31
set interfaces ae1 unit 1 family inet6 address 10:1:1::3/127
set interfaces ae2 vlan-tagging
set interfaces ae2 aggregated-ether-options minimum-links 1
set interfaces ae2 aggregated-ether-options lacp active
set interfaces ae2 aggregated-ether-options lacp periodic fast
set interfaces ae2 unit 0 vlan-id 9
set interfaces ae2 unit 0 family inet address 10.1.1.9/31
set interfaces ae2 unit 0 family inet6 address 10:2:2::1/127
set interfaces ae2 unit 1 vlan-id 10
set interfaces ae2 unit 1 family inet address 10.1.1.11/31
set interfaces ae2 unit 1 family inet6 address 10:2:2::3/127
set interfaces ae3 vlan-tagging
set interfaces ae3 aggregated-ether-options minimum-links 1
set interfaces ae3 aggregated-ether-options lacp active
set interfaces ae3 aggregated-ether-options lacp periodic fast
set interfaces ae3 unit 0 vlan-id 9
set interfaces ae3 unit 0 family inet address 10.1.1.17/31
set interfaces ae3 unit 0 family inet6 address 10:3:3::1/127
set interfaces ae3 unit 1 vlan-id 10
set interfaces ae3 unit 1 family inet address 10.1.1.19/31
set interfaces ae3 unit 1 family inet6 address 10:3:3::3/127
set interfaces ae10 flexible-vlan-tagging
set interfaces ae10 encapsulation flexible-ethernet-services
set interfaces ae10 aggregated-ether-options minimum-links 1
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae10 unit 40 vlan-id 40
set interfaces ae10 unit 40 family inet address 40.1.1.2/30
set interfaces ae10 unit 40 family inet6 address 40:1:1::2/124
set interfaces ae10 unit 80 vlan-id 80
set interfaces ae10 unit 80 family inet address 80.1.1.2/30
set interfaces ae10 unit 80 family inet6 address 80:1:1::2/124
set routing-instances TRUST_VR instance-type virtual-router
set routing-instances TRUST_VR routing-options autonomous-system 1000
set routing-instances TRUST_VR routing-options autonomous-system independent-domain no-attrset
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router type external
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router export
srx_ike_endpoint_export
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router peer-as 1500
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router neighbor 40.1.1.1
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 neighbor 10.1.1.0
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 neighbor 10.1.1.8
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 neighbor 10.1.1.16
set routing-instances TRUST_VR interface ae1.0
set routing-instances TRUST_VR interface ae2.0
set routing-instances TRUST_VR interface ae3.0
set routing-instances TRUST_VR interface ae10.40
set policy-options policy-statement srx_ike_endpoint_export term 1 from protocol bgp
set policy-options policy-statement srx_ike_endpoint_export term 1 from route-filter
100.0.0.1/32 exact
set policy-options policy-statement srx_ike_endpoint_export term 1 then next-hop self
set policy-options policy-statement srx_ike_endpoint_export term 1 then accept
set policy-options policy-statement srx_ike_endpoint_export term 2 then reject
set policy-options policy-statement trust-to-untrust-export term 1 from protocol bgp
set policy-options policy-statement trust-to-untrust-export term 1 from protocol static
set policy-options policy-statement trust-to-untrust-export term 1 then next-hop self
set policy-options policy-statement trust-to-untrust-export term 1 then accept
set policy-options policy-statement trust-to-untrust-export term 2 then reject
set routing-instances UNTRUST_VR instance-type virtual-router
set routing-instances UNTRUST_VR routing-options autonomous-system 2000
set routing-instances UNTRUST_VR routing-options autonomous-system independent-domain no-attrset
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router export
srx_ari_route_export
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router peer-as 2500
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection minimum-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection minimum-receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router neighbor 80.1.1.1
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 neighbor 10.1.1.2
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 neighbor 10.1.1.10
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
```

```
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 neighbor 10.1.1.18
set routing-instances UNTRUST_VR interface ae1.1
set routing-instances UNTRUST_VR interface ae2.1
set routing-instances UNTRUST_VR interface ae3.1
set routing-instances UNTRUST_VR interface ae10.80
set policy-options policy-statement srx_ari_route_export term 1 from protocol bgp
set policy-options policy-statement srx_ari_route_export term 1 from route-filter 6.0.0.0/8
orlonger
set policy-options policy-statement srx_ari_route_export term 1 then next-hop self
set policy-options policy-statement srx_ari_route_export term 1 then accept
set policy-options policy-statement srx_ari_route_export term 2 then reject
set policy-options policy-statement untrust-to-trust-export term 1 from protocol bgp
set policy-options policy-statement untrust-to-trust-export term 1 from protocol static
set policy-options policy-statement untrust-to-trust-export term 1 then next-hop self
set policy-options policy-statement untrust-to-trust-export term 1 then accept
set policy-options policy-statement untrust-to-trust-export term 2 then reject
set policy-options policy-statement pfe_consistent_hash from route-filter 100.0.0.1/32 exact
set policy-options policy-statement pfe_consistent_hash then load-balance consistent-hash
set policy-options policy-statement pfe_consistent_hash then accept
set policy-options policy-statement pfe_lb_hash term source_hash from route-filter 100.0.0.1/32
exact
set policy-options policy-statement pfe_lb_hash term source_hash then load-balance source-ip-only
set policy-options policy-statement pfe_lb_hash term source_hash then accept
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then load-balance per-packet
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then accept
set routing-options forwarding-table export pfe_lb_hash
```

### Configure IPsec Initiator

```
[edit]
set services service-set IPSEC_TUN_1 next-hop-service inside-service-interface vms-3/0/0.1
set services service-set IPSEC_TUN_1 next-hop-service outside-service-interface vms-3/0/0.2001
set services service-set IPSEC_TUN_1 ipsec-vpn TUN_1
set services service-set IPSEC_TUN_2 next-hop-service inside-service-interface vms-3/0/0.2
set services service-set IPSEC_TUN_2 next-hop-service outside-service-interface vms-3/0/0.2002
set services service-set IPSEC_TUN_2 ipsec-vpn TUN_2
set services service-set IPSEC_TUN_3 next-hop-service inside-service-interface vms-3/0/0.3
set services service-set IPSEC_TUN_3 next-hop-service outside-service-interface vms-3/0/0.2003
set services service-set IPSEC_TUN_3 ipsec-vpn TUN_3
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group2
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POLICY proposals IKE_PROP
set security ike policy IKE_POLICY pre-shared-key ascii-text "$ABC123"
set security ike gateway IKE_GW_1 ike-policy IKE_POLICY
set security ike gateway IKE_GW_1 address 100.0.0.1
set security ike gateway IKE_GW_1 dead-peer-detection probe-idle-tunnel
set security ike gateway IKE_GW_1 dead-peer-detection interval 10
set security ike gateway IKE_GW_1 dead-peer-detection threshold 3
set security ike gateway IKE_GW_1 local-identity hostname peer1.juniper.net
set security ike gateway IKE_GW_1 remote-identity hostname vsrx.juniper.net
set security ike gateway IKE_GW_1 external-interface lo0.0
set security ike gateway IKE_GW_1 local-address 200.0.0.1
set security ike gateway IKE_GW_1 version v2-only
set security ike gateway IKE_GW_2 ike-policy IKE_POLICY
set security ike gateway IKE_GW_2 address 100.0.0.1
set security ike gateway IKE_GW_2 dead-peer-detection probe-idle-tunnel
set security ike gateway IKE_GW_2 dead-peer-detection interval 10
set security ike gateway IKE_GW_2 dead-peer-detection threshold 3
set security ike gateway IKE_GW_2 local-identity hostname peer2.juniper.net
set security ike gateway IKE_GW_2 remote-identity hostname vsrx.juniper.net
set security ike gateway IKE_GW_2 external-interface lo0.0
set security ike gateway IKE_GW_2 local-address 200.0.0.2
set security ike gateway IKE_GW_2 version v2-only
set security ike gateway IKE_GW_3 ike-policy IKE_POLICY
set security ike gateway IKE_GW_3 address 100.0.0.1
set security ike gateway IKE_GW_3 dead-peer-detection probe-idle-tunnel
set security ike gateway IKE_GW_3 dead-peer-detection interval 10
set security ike gateway IKE_GW_3 dead-peer-detection threshold 3
set security ike gateway IKE_GW_3 local-identity hostname peer3.juniper.net
set security ike gateway IKE_GW_3 remote-identity hostname vsrx.juniper.net
set security ike gateway IKE_GW_3 external-interface lo0.0
set security ike gateway IKE_GW_3 local-address 200.0.0.6
set security ike gateway IKE_GW_3 version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3600
set security ipsec policy IPSEC_POLICY proposals IPSEC_PROP
set security ipsec vpn TUN_1 bind-interface st0.1
set security ipsec vpn TUN_1 ike gateway IKE_GW_1
set security ipsec vpn TUN_1 ike ipsec-policy IPSEC_POLICY
set security ipsec vpn TUN_1 traffic-selector ts1 local-ip 6.0.0.1/32
set security ipsec vpn TUN_1 traffic-selector ts1 remote-ip 75.0.0.1/32
set security ipsec vpn TUN_1 establish-tunnels immediately
set security ipsec vpn TUN_2 bind-interface st0.2
set security ipsec vpn TUN_2 ike gateway IKE_GW_2
set security ipsec vpn TUN_2 ike ipsec-policy IPSEC_POLICY
set security ipsec vpn TUN_2 traffic-selector ts1 local-ip 6.0.0.2/32
set security ipsec vpn TUN_2 traffic-selector ts1 remote-ip 75.0.0.2/32
set security ipsec vpn TUN_2 establish-tunnels immediately
set security ipsec vpn TUN_3 bind-interface st0.3
set security ipsec vpn TUN_3 ike gateway IKE_GW_3
set security ipsec vpn TUN_3 ike ipsec-policy IPSEC_POLICY
set security ipsec vpn TUN_3 traffic-selector ts1 local-ip 6.0.0.3/32
set security ipsec vpn TUN_3 traffic-selector ts1 remote-ip 75.0.0.3/32
set security ipsec vpn TUN_3 establish-tunnels immediately
set security ipsec anti-replay-window-size 512
set security flow power-mode-ipsec
set interfaces vms-3/0/0 unit 1 family inet
set interfaces vms-3/0/0 unit 1 service-domain inside
set interfaces vms-3/0/0 unit 2 family inet
set interfaces vms-3/0/0 unit 2 service-domain inside
set interfaces vms-3/0/0 unit 3 family inet
set interfaces vms-3/0/0 unit 3 service-domain inside
set interfaces vms-3/0/0 unit 2001 family inet
set interfaces vms-3/0/0 unit 2001 service-domain outside
set interfaces vms-3/0/0 unit 2002 family inet
set interfaces vms-3/0/0 unit 2002 service-domain outside
set interfaces vms-3/0/0 unit 2003 family inet
set interfaces vms-3/0/0 unit 2003 service-domain outside
set interfaces lo0 unit 0 family inet address 200.0.0.1/32
set interfaces lo0 unit 0 family inet address 200.0.0.2/32
set interfaces lo0 unit 0 family inet address 200.0.0.6/32
set interfaces st0 unit 1 family inet
set interfaces st0 unit 2 family inet
set interfaces st0 unit 3 family inet
set interfaces et-7/0/0 gigether-options 802.3ad ae10
set interfaces et-7/1/3 gigether-options 802.3ad ae10
set interfaces et-7/0/3 gigether-options 802.3ad ae10
set interfaces et-7/0/4 gigether-options 802.3ad ae10
set interfaces et-7/0/1 gigether-options 802.3ad ae11
set interfaces et-7/0/2 gigether-options 802.3ad ae11
set interfaces et-7/1/0 gigether-options 802.3ad ae11
set interfaces et-7/1/1 gigether-options 802.3ad ae11
set interfaces et-7/1/2 mtu 9192
set interfaces et-7/1/2 unit 0 family inet address 50.0.0.1/30
set interfaces et-7/1/4 mtu 9192
set interfaces et-7/1/4 unit 0 family inet address 60.0.0.1/30
set interfaces ae10 flexible-vlan-tagging
set interfaces ae10 encapsulation flexible-ethernet-services
set interfaces ae10 aggregated-ether-options minimum-links 1
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae10 unit 40 vlan-id 40
set interfaces ae10 unit 40 family inet address 40.1.1.1/30
set interfaces ae10 unit 40 family inet6 address 40:1:1::1/124
set interfaces ae10 unit 80 vlan-id 80
set interfaces ae10 unit 80 family inet address 80.1.1.1/30
set interfaces ae10 unit 80 family inet6 address 80:1:1::1/124
set interfaces ae11 flexible-vlan-tagging
set interfaces ae11 encapsulation flexible-ethernet-services
set interfaces ae11 aggregated-ether-options minimum-links 1
set interfaces ae11 aggregated-ether-options lacp active
set interfaces ae11 aggregated-ether-options lacp periodic fast
set interfaces ae11 unit 41 vlan-id 41
set interfaces ae11 unit 41 family inet address 41.1.1.1/30
set interfaces ae11 unit 41 family inet6 address 41:1:1::1/124
set interfaces ae11 unit 81 vlan-id 81
set interfaces ae11 unit 81 family inet address 81.1.1.1/30
set interfaces ae11 unit 81 family inet6 address 81:1:1::1/124
set routing-instances TRUST_VR instance-type virtual-router
set routing-instances TRUST_VR routing-options autonomous-system 1500
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust type external
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust export
client_to_server_export
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust peer-as 1000
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust local-as 1500
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust neighbor 40.1.1.2
set routing-instances TRUST_VR protocols bgp multipath
set routing-instances TRUST_VR interface vms-3/0/0.2001
set routing-instances TRUST_VR interface vms-3/0/0.2002
set routing-instances TRUST_VR interface vms-3/0/0.2003
set routing-instances TRUST_VR interface ae10.40
set routing-instances TRUST_VR interface lo0.0
set policy-options policy-statement client_to_server_export term 1 from protocol direct
set policy-options policy-statement client_to_server_export term 1 from route-filter 200.0.0.0/8
orlonger
set policy-options policy-statement client_to_server_export term 1 then accept
set policy-options policy-statement client_to_server_export term 2 then reject
set policy-options policy-statement client_to_server_export_mx2 term 1 from protocol static
set policy-options policy-statement client_to_server_export_mx2 term 1 from route-filter
141.0.0.0/8 orlonger
set policy-options policy-statement client_to_server_export_mx2 term 1 from route-filter
140.0.0.0/8 orlonger
set policy-options policy-statement client_to_server_export_mx2 term 1 then accept
set policy-options policy-statement client_to_server_export_mx2 term 2 then reject
set routing-instances UNTRUST_VR instance-type virtual-router
set routing-instances UNTRUST_VR routing-options autonomous-system 2500
set routing-instances UNTRUST_VR routing-options static route 75.0.0.0/8 next-hop 60.0.0.2
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust type external
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust export
server_to_client_export
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust peer-as 2000
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust local-as 2500
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection minimum-interval 300
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection minimum-receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection multiplier 3
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust neighbor 80.1.1.2
set routing-instances UNTRUST_VR protocols bgp multipath
set routing-instances UNTRUST_VR interface et-7/1/4.0
set routing-instances UNTRUST_VR interface ae10.80
set policy-options policy-statement server_to_client_export term t1 from protocol static
set policy-options policy-statement server_to_client_export term t1 from route-filter 75.0.0.0/8
exact
set policy-options policy-statement server_to_client_export term t1 then accept
set policy-options policy-statement server_to_client_export term t2 then reject
set policy-options policy-statement server_to_client_export_mx2 term t1 from protocol static
set policy-options policy-statement server_to_client_export_mx2 term t1 from route-filter
0.0.0.0/0 exact
set policy-options policy-statement server_to_client_export_mx2 term t1 then accept
set policy-options policy-statement server_to_client_export_mx2 term t2 then reject
set routing-instances client instance-type virtual-router
set routing-instances client routing-options static route 6.0.0.0/8 next-hop 50.0.0.2
set routing-instances client interface vms-3/0/0.1
set routing-instances client interface vms-3/0/0.2
set routing-instances client interface vms-3/0/0.3
set routing-instances client interface et-7/1/2.0
set routing-instances client interface st0.1
set routing-instances client interface st0.2
set routing-instances client interface st0.3
set policy-options policy-statement ECMP_POLICY-LB then load-balance per-packet
set routing-options forwarding-table export ECMP_POLICY-LB
```

Configure SRX1

```
[edit]
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group2
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POLICY proposals IKE_PROP
set security ike policy IKE_POLICY pre-shared-key ascii-text "$ABC123"
set security ike gateway avpn_ike_gw ike-policy IKE_POLICY
set security ike gateway avpn_ike_gw dynamic hostname .juniper.net
set security ike gateway avpn_ike_gw dynamic ike-user-type group-ike-id
set security ike gateway avpn_ike_gw dead-peer-detection probe-idle-tunnel
set security ike gateway avpn_ike_gw dead-peer-detection interval 10
set security ike gateway avpn_ike_gw dead-peer-detection threshold 3
set security ike gateway avpn_ike_gw local-identity hostname vsrx.juniper.net
set security ike gateway avpn_ike_gw external-interface lo0.0
set security ike gateway avpn_ike_gw local-address 100.0.0.1
set security ike gateway avpn_ike_gw version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3600
set security ipsec policy IPSEC_POLICY proposals IPSEC_PROP
set security ipsec vpn avpn_ipsec_vpn bind-interface st0.1
set security ipsec vpn avpn_ipsec_vpn ike gateway avpn_ike_gw
set security ipsec vpn avpn_ipsec_vpn ike ipsec-policy IPSEC_POLICY
set security ipsec vpn avpn_ipsec_vpn traffic-selector ts local-ip 0.0.0.0/0
set security ipsec vpn avpn_ipsec_vpn traffic-selector ts remote-ip 0.0.0.0/0
set security ipsec anti-replay-window-size 512
set interfaces lo0 unit 0 family inet address 100.0.0.1/32
set interfaces st0 unit 1 family inet
set interfaces st0 unit 2 family inet
set interfaces st0 unit 3 family inet
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_trust_zone interfaces lo0.0
set security zones security-zone vr-1_trust_zone interfaces st0.1
set security zones security-zone vr-1_trust_zone interfaces st0.2
set security zones security-zone vr-1_trust_zone interfaces st0.3
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match source-address ipsec_data_source_prefix_6.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match source-address ike_source_prefix_200.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
then permit
set security policies default-policy permit-all
set security address-book global address ipsec_data_source_prefix_6.0.0.0/8 6.0.0.0/8
set security address-book global address ike_source_prefix_200.0.0.0/8 200.0.0.0/8
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 1
set interfaces ae1 unit 0 family inet address 10.1.1.0/31
set interfaces ae1 unit 0 family inet6 address 10:1:1::0/127
set interfaces ae1 unit 1 vlan-id 2
set interfaces ae1 unit 1 family inet address 10.1.1.2/31
set interfaces ae1 unit 1 family inet6 address 10:1:1::2/127
set protocols bgp group Vsrx-to-MX_TRUST type external
set protocols bgp group Vsrx-to-MX_TRUST export ike_endpoint_export_policy
set protocols bgp group Vsrx-to-MX_TRUST local-as 500
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-interval 300
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-receive-interval 300
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection multiplier 3
set protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.1 peer-as 1000
set protocols bgp group Vsrx-to-MX_UNTRUST type external
set protocols bgp group Vsrx-to-MX_UNTRUST export ari_export_untrust
set protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-interval 300
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-receive-interval 300
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection multiplier 3
set protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.3 peer-as 2000
set policy-options policy-statement ari_export_untrust term 1 from protocol ari-ts
set policy-options policy-statement ari_export_untrust term 1 then accept
set policy-options policy-statement ari_export_untrust term defualt then reject
set policy-options policy-statement ike_endpoint_export_policy term 1 from protocol direct
set policy-options policy-statement ike_endpoint_export_policy term 1 from route-filter
100.0.0.1/32 exact
set policy-options policy-statement ike_endpoint_export_policy term 1 then next-hop self
set policy-options policy-statement ike_endpoint_export_policy term 1 then accept
set policy-options policy-statement ike_endpoint_export_policy term 2 then reject
set policy-options policy-statement ecmp_policy_lab then load-balance per-packet
set routing-options forwarding-table export ecmp_policy_lab
```

Configure SRX2

```
[edit]
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group2
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POLICY proposals IKE_PROP
set security ike policy IKE_POLICY pre-shared-key ascii-text "$ABC123"
set security ike gateway avpn_ike_gw ike-policy IKE_POLICY
set security ike gateway avpn_ike_gw dynamic hostname .juniper.net
set security ike gateway avpn_ike_gw dynamic ike-user-type group-ike-id
set security ike gateway avpn_ike_gw dead-peer-detection probe-idle-tunnel
set security ike gateway avpn_ike_gw dead-peer-detection interval 10
set security ike gateway avpn_ike_gw dead-peer-detection threshold 3
set security ike gateway avpn_ike_gw local-identity hostname vsrx.juniper.net
set security ike gateway avpn_ike_gw external-interface lo0.0
set security ike gateway avpn_ike_gw local-address 100.0.0.1
set security ike gateway avpn_ike_gw version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3600
set security ipsec policy IPSEC_POLICY proposals IPSEC_PROP
set security ipsec vpn avpn_ipsec_vpn bind-interface st0.1
set security ipsec vpn avpn_ipsec_vpn ike gateway avpn_ike_gw
set security ipsec vpn avpn_ipsec_vpn ike ipsec-policy IPSEC_POLICY
set security ipsec vpn avpn_ipsec_vpn traffic-selector ts local-ip 0.0.0.0/0
set security ipsec vpn avpn_ipsec_vpn traffic-selector ts remote-ip 0.0.0.0/0
set security ipsec anti-replay-window-size 512
set interfaces lo0 unit 0 family inet address 100.0.0.1/32
set interfaces st0 unit 1 family inet
set interfaces st0 unit 2 family inet
set interfaces st0 unit 3 family inet
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_trust_zone interfaces lo0.0
set security zones security-zone vr-1_trust_zone interfaces st0.1
set security zones security-zone vr-1_trust_zone interfaces st0.2
set security zones security-zone vr-1_trust_zone interfaces st0.3
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match source-address ipsec_data_source_prefix_6.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match source-address ike_source_prefix_200.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match destination-address any
```

```
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
then permit
set security policies default-policy permit-all
set security address-book global address ipsec_data_source_prefix_6.0.0.0/8 6.0.0.0/8
set security address-book global address ike_source_prefix_200.0.0.0/8 200.0.0.0/8
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 9
set interfaces ae1 unit 0 family inet address 10.1.1.8/31
set interfaces ae1 unit 0 family inet6 address 10:2:2::0/127
set interfaces ae1 unit 1 vlan-id 10
set interfaces ae1 unit 1 family inet address 10.1.1.10/31
set interfaces ae1 unit 1 family inet6 address 10:2:2::2/127
set protocols bgp group Vsrx-to-MX_TRUST type external
set protocols bgp group Vsrx-to-MX_TRUST export ike_endpoint_export_policy
set protocols bgp group Vsrx-to-MX_TRUST local-as 500
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-interval 300
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-receive-interval 300
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection multiplier 3
set protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.9 peer-as 1000
set protocols bgp group Vsrx-to-MX_UNTRUST type external
set protocols bgp group Vsrx-to-MX_UNTRUST export ari_export_untrust
set protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-interval 300
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-receive-interval 300
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection multiplier 3
set protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.11 peer-as 2000
set policy-options policy-statement ari_export_untrust term 1 from protocol ari-ts
set policy-options policy-statement ari_export_untrust term 1 then accept
set policy-options policy-statement ari_export_untrust term defualt then reject
set policy-options policy-statement ike_endpoint_export_policy term 1 from protocol direct
set policy-options policy-statement ike_endpoint_export_policy term 1 from route-filter
100.0.0.1/32 exact
set policy-options policy-statement ike_endpoint_export_policy term 1 then next-hop self
set policy-options policy-statement ike_endpoint_export_policy term 1 then accept
set policy-options policy-statement ike_endpoint_export_policy term 2 then reject
set policy-options policy-statement ecmp_policy_lab then load-balance per-packet
set routing-options forwarding-table export ecmp_policy_lab
```

### Configure SRX3

```
[edit]
set security ike proposal IKE_PROP authentication-method pre-shared-keys
set security ike proposal IKE_PROP dh-group group2
set security ike proposal IKE_PROP authentication-algorithm sha1
set security ike proposal IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal IKE_PROP lifetime-seconds 3600
set security ike policy IKE_POLICY proposals IKE_PROP
set security ike policy IKE_POLICY pre-shared-key ascii-text "$ABC123"
set security ike gateway avpn_ike_gw ike-policy IKE_POLICY
set security ike gateway avpn_ike_gw dynamic hostname .juniper.net
set security ike gateway avpn_ike_gw dynamic ike-user-type group-ike-id
set security ike gateway avpn_ike_gw dead-peer-detection probe-idle-tunnel
set security ike gateway avpn_ike_gw dead-peer-detection interval 10
set security ike gateway avpn_ike_gw dead-peer-detection threshold 3
set security ike gateway avpn_ike_gw local-identity hostname vsrx.juniper.net
set security ike gateway avpn_ike_gw external-interface lo0.0
set security ike gateway avpn_ike_gw local-address 100.0.0.1
set security ike gateway avpn_ike_gw version v2-only
set security ipsec proposal IPSEC_PROP protocol esp
set security ipsec proposal IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal IPSEC_PROP lifetime-seconds 3600
set security ipsec policy IPSEC_POLICY proposals IPSEC_PROP
set security ipsec vpn avpn_ipsec_vpn bind-interface st0.1
set security ipsec vpn avpn_ipsec_vpn ike gateway avpn_ike_gw
set security ipsec vpn avpn_ipsec_vpn ike ipsec-policy IPSEC_POLICY
set security ipsec vpn avpn_ipsec_vpn traffic-selector ts local-ip 0.0.0.0/0
set security ipsec vpn avpn_ipsec_vpn traffic-selector ts remote-ip 0.0.0.0/0
set security ipsec anti-replay-window-size 512
set interfaces lo0 unit 0 family inet address 100.0.0.1/32
set interfaces st0 unit 1 family inet
set interfaces st0 unit 2 family inet
set interfaces st0 unit 3 family inet
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_trust_zone interfaces lo0.0
set security zones security-zone vr-1_trust_zone interfaces st0.1
set security zones security-zone vr-1_trust_zone interfaces st0.2
set security zones security-zone vr-1_trust_zone interfaces st0.3
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match source-address ipsec_data_source_prefix_6.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
IPSEC_DATA_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match source-address ike_source_prefix_200.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_trust_zone policy IKE_ALLOW_POLICY
then permit
set security policies default-policy permit-all
set security address-book global address ipsec_data_source_prefix_6.0.0.0/8 6.0.0.0/8
set security address-book global address ike_source_prefix_200.0.0.0/8 200.0.0.0/8
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 9
set interfaces ae1 unit 0 family inet address 10.1.1.16/31
set interfaces ae1 unit 1 vlan-id 10
set interfaces ae1 unit 1 family inet address 10.1.1.18/31
set protocols bgp group Vsrx-to-MX_TRUST type external
set protocols bgp group Vsrx-to-MX_TRUST export ike_endpoint_export_policy
set protocols bgp group Vsrx-to-MX_TRUST local-as 500
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-interval 300
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-receive-interval 300
set protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection multiplier 3
set protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.17 peer-as 1000
set protocols bgp group Vsrx-to-MX_UNTRUST type external
set protocols bgp group Vsrx-to-MX_UNTRUST export ari_export_untrust
set protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-interval 300
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-receive-interval 300
set protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection multiplier 3
set protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.19 peer-as 2000
set policy-options policy-statement ari_export_untrust term 1 from protocol ari-ts
set policy-options policy-statement ari_export_untrust term 1 then accept
set policy-options policy-statement ari_export_untrust term defualt then reject
set policy-options policy-statement ike_endpoint_export_policy term 1 from protocol direct
set policy-options policy-statement ike_endpoint_export_policy term 1 from route-filter
100.0.0.1/32 exact
set policy-options policy-statement ike_endpoint_export_policy term 1 then next-hop self
set policy-options policy-statement ike_endpoint_export_policy term 1 then accept
set policy-options policy-statement ike_endpoint_export_policy term 2 then reject
set policy-options policy-statement ecmp_policy_lab then load-balance per-packet
set routing-options forwarding-table export ecmp_policy_lab
```

## Verification

The following items highlight a list of show commands used to verify the feature in this example.

- Verify MX Series configuration
- Verify SRX1 configuration
- Verify SRX2 configuration
- Verify SRX3 configuration

### Verify MX Series Configuration

```
user@MX304# run show route 100.0.0.1/32 active-path
              TRUST_VR.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              100.0.0.1/32       *[BGP/170] 03:14:10, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                    to 10.1.1.0 via ae1.0
                                 >  to 10.1.1.8 via ae2.0
                                    to 10.1.1.16 via ae3.0
```

```
user@MX304# run show route 100.0.0.1/32 active-path extensive
              TRUST_VR.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
              100.0.0.1/32 (3 entries, 1 announced)
              TSI:
              KRT in-kernel 100.0.0.1/32 -> {list:10.1.1.0, 10.1.1.8, 10.1.1.16 Flags source ip 
load-balance}
              Page 0 idx 1, (group MX-to-TRUST_GW_Router type External) Type 1 val 0x12b04ce0 
(adv_entry)
                Advertised metrics:
                  Flags: Nexthop Change
                  Nexthop: Self
                  AS path: [1000] 500 I
                  Communities:
                 Advertise: 00000001
              Path 100.0.0.1
              from 10.1.1.8
              Vector len 4.  Val: 1
                     *BGP    Preference: 170/-101
                             Next hop type: Router, Next hop index: 0
                             Address: 0xf918b24
                             Next-hop reference count: 2, Next-hop session id: 0
                             Kernel Table Id: 0
                             Source: 10.1.1.8
                             Next hop: 10.1.1.0 via ae1.0
                             Session Id: 0
                             Next hop: 10.1.1.8 via ae2.0, selected
                             Session Id: 0
                             Next hop: 10.1.1.16 via ae3.0
                             Session Id: 0
                             State: <Active Ext LoadBalConsistentHash>
                             Local AS:  1000 Peer AS:   500
                             Age: 3:14:15
                             Validation State: unverified
                             Task: BGP_500_1000.10.1.1.8
                             Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
                             AS path: 500 I
                             Accepted Multipath
                             Localpref: 100
```

```
                              Router ID: 10.255.33.26
                              Thread: junos-main
```

```
user@MX304# run show route 75/8
              UNTRUST_VR.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              75.0.0.0/8         *[BGP/170] 06:27:07, localpref 100
                                   AS path: 2500 I, validation-state: unverified
                                 >  to 80.1.1.1 via ae10.80
```

```
user@MX304# run show route 6/8
              UNTRUST_VR.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              6.0.0.1/32         *[BGP/170] 03:13:30, MED 5, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.18 via ae3.1
              6.0.0.2/32         *[BGP/170] 03:13:31, MED 5, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.10 via ae2.1
              6.0.0.3/32         *[BGP/170] 02:12:57, MED 5, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.2 via ae1.1
```

```
user@MX304# run show route 200/8
              TRUST_VR.inet.0: 12 destinations, 14 routes (12 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              200.0.0.1/32       *[BGP/170] 06:26:30, localpref 100
                                   AS path: 1500 I, validation-state: unverified
                                 >  to 40.1.1.1 via ae10.40
              200.0.0.2/32       *[BGP/170] 06:26:30, localpref 100
                                   AS path: 1500 I, validation-state: unverified
                                 >  to 40.1.1.1 via ae10.40
              200.0.0.6/32       *[BGP/170] 02:14:13, localpref 100
```

```
                                     AS path: 1500 I, validation-state: unverified
                                   >  to 40.1.1.1 via ae10.40
```

```
user@MX304# run show bgp summary
              Warning: License key missing; requires 'bgp' license
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 8 Peers: 8 Down peers: 0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              10.1.1.0                500        501        493       0       6     3:44:50
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.2                500        466        449       0       6     3:25:47
Establ
               UNTRUST_VR.inet.0: 1/1/1/0
              10.1.1.8                500        503        495       0       5     3:45:35
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.10               500        529        504       0       3     3:50:55
Establ
               UNTRUST_VR.inet.0: 1/1/1/0
              10.1.1.16               500        780        768       0       3     5:50:32
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.18               500        792        763       0       2     5:50:37
Establ
               UNTRUST_VR.inet.0: 1/1/1/0
              40.1.1.1               1500      13601      13345       0       1  4d 7:42:56
Establ
               TRUST_VR.inet.0: 3/3/3/0
              80.1.1.1               2500      13588      13405       0       1  4d 7:42:56
Establ
               UNTRUST_VR.inet.0: 1/1/1/0
```

```
user@MX304# run show bfd session
                                                               Detect   Transmit
              Address                   State     Interface     Time     Interval  Multiplier
              10.1.1.0                  Up        ae1.0         0.900     0.300        3
              10.1.1.2                  Up        ae1.1         0.900     0.300        3
              10.1.1.8                 Up        ae2.0          0.900     0.300        3
              10.1.1.10                Up        ae2.1          0.900     0.300        3
              10.1.1.16                Up        ae3.0          0.900     0.300        3
              10.1.1.18                Up        ae3.1          0.900     0.300        3
              40.1.1.1                 Up        ae10.40        0.900     0.300        3
              80.1.1.1                 Up        ae10.80        0.900     0.300        3
              8 sessions, 8 clients
              Cumulative transmit rate 26.7 pps, cumulative receive rate 26.7 pps
```

## Verify IPsec Initiator Configuration

```
user@IPsec# run show security ike security-associations
              Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
              380363  UP     a8b642f8a828eb57  de97df1ba140e292  IKEv2          100.0.0.1
              380364  UP     55b7e5a43d7462ba  201a1b9523442c50  IKEv2          100.0.0.1
              380365  UP     3484ff0e307d1ddc  869cabffae9d261e  IKEv2          100.0.0.1
```

```
user@IPsec# run show security ipsec security-associations
               Total active tunnels: 3     Total IPsec sas: 3
               ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
               <542828 ESP:aes-gcm-256/aes256-gcm 0xd23dbafa 3009/ unlim - root 500 100.0.0.1
               >542828 ESP:aes-gcm-256/aes256-gcm 0xb74e6311 3009/ unlim - root 500 100.0.0.1
               <542827 ESP:aes-gcm-256/aes256-gcm 0xb2943202 3053/ unlim - root 500 100.0.0.1
               >542827 ESP:aes-gcm-256/aes256-gcm 0xd87a527b 3053/ unlim - root 500 100.0.0.1
               <542832 ESP:aes-gcm-256/aes256-gcm 0x960b3fe9 834/ unlim - root 500 100.0.0.1
               >542832 ESP:aes-gcm-256/aes256-gcm 0x1143a22f 834/ unlim - root 500 100.0.0.1
```

## Verify SRX1 Configuration

```
user@SRX1> show security ike security-associations
              Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
              20      UP     a8b642f8a828eb57  de97df1ba140e292  IKEv2          200.0.0.6
```

```
user@SRX1> show security ipsec security-associations
               Total active tunnels: 1     Total IPsec sas: 1
               ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
```

```
                <500017 ESP:aes-gcm-256/aes256-gcm 0x1143a22f 1314/ unlim - root 500 200.0.0.6
                >500017 ESP:aes-gcm-256/aes256-gcm 0x960b3fe9 1314/ unlim - root 500 200.0.0.6
```

```
user@SRX1> show bgp summary
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 2 Peers: 2 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              inet.0
                                    4          4          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              10.1.1.1              1000        500        505       0       4     3:47:18
Establ
               inet.0: 3/3/3/0
              10.1.1.3              2000        456        470       0       4     3:28:15
Establ
               inet.0: 1/1/1/0
```

```
user@SRX1> show bfd session
                                                              Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              10.1.1.1                 Up        ae1.0          0.900     0.300        3
              10.1.1.3                 Up        ae1.1          0.900     0.300        3
              2 sessions, 2 clients
              Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
```

```
user@SRX1> show route 200.0.0.0/8
              inet.0: 27 destinations, 27 routes (26 active, 0 holddown, 1 hidden)
              + = Active Route, - = Last Active, * = Both
              200.0.0.1/32       *[BGP/170] 03:47:45, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.1 via ae1.0
              200.0.0.2/32       *[BGP/170] 03:47:45, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.1 via ae1.0
              200.0.0.6/32       *[BGP/170] 02:16:35, localpref 100
```

```
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.1 via ae1.0
```

```
user@SRX1> show route 6.0.0.0/8
              inet.0: 27 destinations, 27 routes (26 active, 0 holddown, 1 hidden)
              + = Active Route, - = Last Active, * = Both
              6.0.0.3/32         *[ARI-TS/5] 02:16:49, metric 5
                                 >  via st0.1
```

```
user@SRX1> show route 75.0.0.0/8
              inet.0: 27 destinations, 27 routes (26 active, 0 holddown, 1 hidden)
              + = Active Route, - = Last Active, * = Both
              75.0.0.0/8         *[BGP/170] 03:29:51, localpref 100
                                   AS path: 2000 2500 I, validation-state: unverified
                                 >  to 10.1.1.3 via ae1.1
```

```
user@SRX1> show security flow session protocol esp
              Session ID: 2894133, Policy name: N/A, Timeout: N/A, Session State: Valid
               In: 200.0.0.6/0 --> 100.0.0.1/0;esp, Conn Tag: 0x0, If: ae1.0, Pkts: 0, Bytes: 0,
              Session ID: 2894160, Policy name: N/A, Timeout: N/A, Session State: Valid
               In: 200.0.0.6/4419 --> 100.0.0.1/41519;esp, Conn Tag: 0x0, If: lo0.0, Pkts: 0,
Bytes: 0,
              Total sessions: 2
```

```
user@SRX1> show security flow session protocol udp source-prefix 75.0.0.0/8
              Session ID: 2894145, Policy name: IPSEC_DATA_POLICY, Timeout: 60, Session State:
Valid
               In: 75.0.0.3/2001 --> 6.0.0.3/1002;udp, Conn Tag: 0x0, If: ae1.1, Pkts: 51609457,
Bytes: 30036703974,
               Out: 6.0.0.3/1002 --> 75.0.0.3/2001;udp, Conn Tag: 0x0, If: st0.1, Pkts: 7741418,
Bytes: 4505505276,
```

## Verify SRX2 Configuration

```
user@SRX2> show security ike security-associations
              Index    State  Initiator cookie  Responder cookie  Mode           Remote Address
              26       UP     3484ff0e307d1ddc  869cabffae9d261e  IKEv2          200.0.0.2
```

```
user@SRX2> show security ipsec security-associations
               Total active tunnels: 1     Total IPsec sas: 1
               ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
               <500018 ESP:aes-gcm-256/aes256-gcm 0xd87a527b 3257/ unlim - root 500 200.0.0.2
               >500018 ESP:aes-gcm-256/aes256-gcm 0xb2943202 3257/ unlim - root 500 200.0.0.2
```

```
user@SRX2> show bgp summary
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 2 Peers: 2 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              inet.0
                                     4          4          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              10.1.1.9              1000        511        516       0       3     3:52:21
Establ
               inet.0: 3/3/3/0
              10.1.1.11             2000        520        542       0       1     3:57:40
Establ
               inet.0: 1/1/1/0
```

```
user@SRX2> show bfd session
                                                               Detect   Transmit
              Address                   State     Interface      Time     Interval  Multiplier
              10.1.1.9                  Up        ae1.0          0.900     0.300        3
              10.1.1.11                 Up        ae1.1          0.900     0.300        3
              2 sessions, 2 clients
              Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
```

```
user@SRX2> show route 200.0.0.0/8
              inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
              + = Active Route, - = Last Active, * = Both
              200.0.0.1/32       *[BGP/170] 03:52:29, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.9 via ae1.0
              200.0.0.2/32       *[BGP/170] 03:52:29, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.9 via ae1.0
              200.0.0.6/32       *[BGP/170] 02:20:34, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.9 via ae1.0
```

```
user@SRX2> show route 6.0.0.0/8
              inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
              + = Active Route, - = Last Active, * = Both
              6.0.0.2/32         *[ARI-TS/5] 03:21:10, metric 5
                                 >  via st0.1
```

```
user@SRX2> show route 75.0.0.0/8
              inet.0: 29 destinations, 29 routes (28 active, 0 holddown, 1 hidden)
              + = Active Route, - = Last Active, * = Both
              75.0.0.0/8         *[BGP/170] 03:58:00, localpref 100
                                   AS path: 2000 2500 I, validation-state: unverified
                                 >  to 10.1.1.11 via ae1.1
```

```
user@SRX2> show security flow session protocol esp
              Session ID: 2897660, Policy name: N/A, Timeout: N/A, Session State: Valid
               In: 200.0.0.2/0 --> 100.0.0.1/0;esp, Conn Tag: 0x0, If: ae1.0, Pkts: 0, Bytes: 0,
              Session ID: 2897694, Policy name: N/A, Timeout: N/A, Session State: Valid
               In: 200.0.0.2/55418 --> 100.0.0.1/21115;esp, Conn Tag: 0x0, If: lo0.0, Pkts: 0,
Bytes: 0,
              Total sessions: 2
```

```
user@SRX2> show security flow session protocol udp source-prefix 75.0.0.0/8
              Session ID: 2897677, Policy name: IPSEC_DATA_POLICY, Timeout: 60, Session State:
Valid
               In: 75.0.0.2/2001 --> 6.0.0.2/1009;udp, Conn Tag: 0x0, If: ae1.1, Pkts: 52336685,
Bytes: 30459950670,
               Out: 6.0.0.2/1009 --> 75.0.0.2/2001;udp, Conn Tag: 0x0, If: st0.1, Pkts: 7850503,
Bytes: 4568992746,
```

### Verify SRX3 Configuration

```
user@SRX3> show security ike security-associations
              Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
              19      UP     55b7e5a43d7462ba  201a1b9523442c50  IKEv2          200.0.0.1
```

```
user@SRX3> show security ipsec security-associations
               Total active tunnels: 1     Total IPsec sas: 1
               ID      Algorithm       SPI      Life:sec/kb  Mon lsys Port  Gateway
               <500009 ESP:aes-gcm-256/aes256-gcm 0xb74e6311 3107/ unlim - root 500 200.0.0.1
               >500009 ESP:aes-gcm-256/aes256-gcm 0xd23dbafa 3107/ unlim - root 500 200.0.0.1
```

```
user@SRX3> show bgp summary
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 2 Peers: 2 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              inet.0
                                    4          4          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              10.1.1.17             1000        787        797       0       1     5:58:59
Establ
               inet.0: 3/3/3/0
              10.1.1.19             2000        783        810       0       0     5:59:04
```

```
Establ
               inet.0: 1/1/1/0
```

```
user@SRX3> show bfd session
                                                          Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              10.1.1.17                Up        ae1.0          0.900     0.300        3
              10.1.1.19                Up        ae1.1          0.900     0.300        3
              2 sessions, 2 clients
              Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
```

```
user@SRX3> show route 200.0.0.0/8
              inet.0: 26 destinations, 26 routes (26 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              200.0.0.1/32       *[BGP/170] 05:59:07, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.17 via ae1.0
              200.0.0.2/32       *[BGP/170] 05:59:07, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.17 via ae1.0
              200.0.0.6/32       *[BGP/170] 02:22:15, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.17 via ae1.0
```

```
user@SRX3> show route 6.0.0.0/8
              inet.0: 26 destinations, 26 routes (26 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              6.0.0.1/32         *[ARI-TS/5] 03:22:51, metric 5
                                 >  via st0.1
```

```
user@SRX3> show route 75.0.0.0/8
              inet.0: 26 destinations, 26 routes (26 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              75.0.0.0/8         *[BGP/170] 05:59:22, localpref 100
```

```
                                     AS path: 2000 2500 I, validation-state: unverified
                                   >  to 10.1.1.19 via ae1.1
```

```
user@SRX3> show security flow session protocol esp
              Session ID: 2889066, Policy name: N/A, Timeout: N/A, Session State: Valid
               In: 200.0.0.1/0 --> 100.0.0.1/0;esp, Conn Tag: 0x0, If: ae1.0, Pkts: 0, Bytes: 0,
              Session ID: 2889104, Policy name: N/A, Timeout: N/A, Session State: Valid
               In: 200.0.0.1/46926 --> 100.0.0.1/25361;esp, Conn Tag: 0x0, If: lo0.0, Pkts: 0,
Bytes: 0,
              Total sessions: 2
```

```
user@SRX3> show security flow session protocol udp source-prefix 75.0.0.0/8
              Session ID: 2889087, Policy name: IPSEC_DATA_POLICY, Timeout: 60, Session State:
Valid
               In: 75.0.0.1/2001 --> 6.0.0.1/1005;udp, Conn Tag: 0x0, If: ae1.1, Pkts: 53008715,
Bytes: 30851072130,
               Out: 6.0.0.1/1005 --> 75.0.0.1/2001;udp, Conn Tag: 0x0, If: st0.1, Pkts: 7951308,
Bytes: 4627661256,
```

## Example: Single MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Standalone) for NAT and Stateful Firewall

**SUMMARY**

In this configuration, you'll learn to set up a single MX Series with scaled-out SRX Series Firewalls in standalone for NAT and stateful firewall services.

**IN THIS SECTION**

## Overview

Table 18 on page 108 shows the deployment components used in the example.

**Table 18: Deployment Details**

| CSDS Components | Details |
|---|---|
| Forwarding Layer | MX304 with Junos OS Release 23.4R1 or later |
| Services Layer | vSRX 3.0 with Junos OS Release 23.4R1 or later |
| Redundancy | Single MX Series with ECMP based Consistent Hashing for load balancer.<br><br>SRX Series Firewalls (Standalone) |
| Features | NAPT44 and stateful firewall (IPv4 Support) |
| Additional Component | Gateway router for TRUST and UNTRUST networks. The example uses MX Series. You can use any device. |

See Table 19 on page 108 and Table 20 on page 109 for traffic flow.

**Table 19: Traffic Flows for NAT**

| Feature | Traffic Flow Component | IP Address and Port Number |
|---|---|---|
| NAPT44 on SRX Series Firewall (SRX1) | Original source data client | 140.0.0.0/8 and port 22279 |
| | Original destination Internet server | 100.1.1.0/24 and port 70 |
| | After NAT source | 192.168.64.0/24 and port 2480 |
| | After NAT destination | 100.1.1.0/24 and port 70 |
| NAPT44 on SRX Series Firewall (SRX2) | Original source data client | 140.0.0.0/8 and port 22279 |
| | Original destination Internet server | 100.1.1.0/24 and port 70 |
| | After NAT source | 192.169.64.0/24 and port 2480 |

**Table 19: Traffic Flows for NAT *(Continued)***

| Feature | Traffic Flow Component | IP Address and Port Number |
|---|---|---|
| | After NAT destination | 100.1.1.0/24 and port 70 |
| NAPT44 on SRX Series Firewall (SRX3) | Original source data client | 140.0.0.0/8 and port 22279 |
| | Original destination Internet server | 100.1.1.0/24 and port 70 |
| | After NAT source | 192.170.64.0/24 and port 2480 |
| | After NAT destination | 100.1.1.0/24 and port 70 |

**Table 20: Traffic Flows for Stateful Firewall Services**

| Feature | Traffic Flow Component | IP Address |
|---|---|---|
| Stateful firewall services on SRX Series Firewalls<br><br>(SRX1, SRX2 and SRX3) | Source data client | 141.0.0.0/8 |
| | Destination Internet server | 100.1.1.0/24 |
| | SRX Series with stateful firewall - Source | 141.0.0.0/8 |
| | SRX Series with stateful firewall - Destination | 100.1.1.0/24 |

See Table 21 on page 109 and Table 22 on page 110 for traffic flow.

**Table 21: Load Balancer to SRX Series Firewalls for NAT Services**

| Flow Type | Traffic Flow Component | IP Address |
|---|---|---|
| Forward Flow | Source Load Balancer (Route Filter on MX Series) | 0.0.0.0/0 |
| Reverse Flow | Destination Load Balancer (Routing-Based) | Based on unique NAT pool IP address range |

**Table 22: Load Balancer to SRX Series Firewalls for Stateful Firewall Services**

| Flow Type | Traffic Flow Component | IP Address |
|---|---|---|
| Forward Flow | Source Load Balancer (Route Filter on MX Series) | 0.0.0.0/0 |
| Reverse Flow | Destination Load Balancer (Route Filter on MX Series) | 141.0.0.0/8 |

## Topology Illustration

**Figure 23: Single MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls for NAT and Stateful Firewall Services**

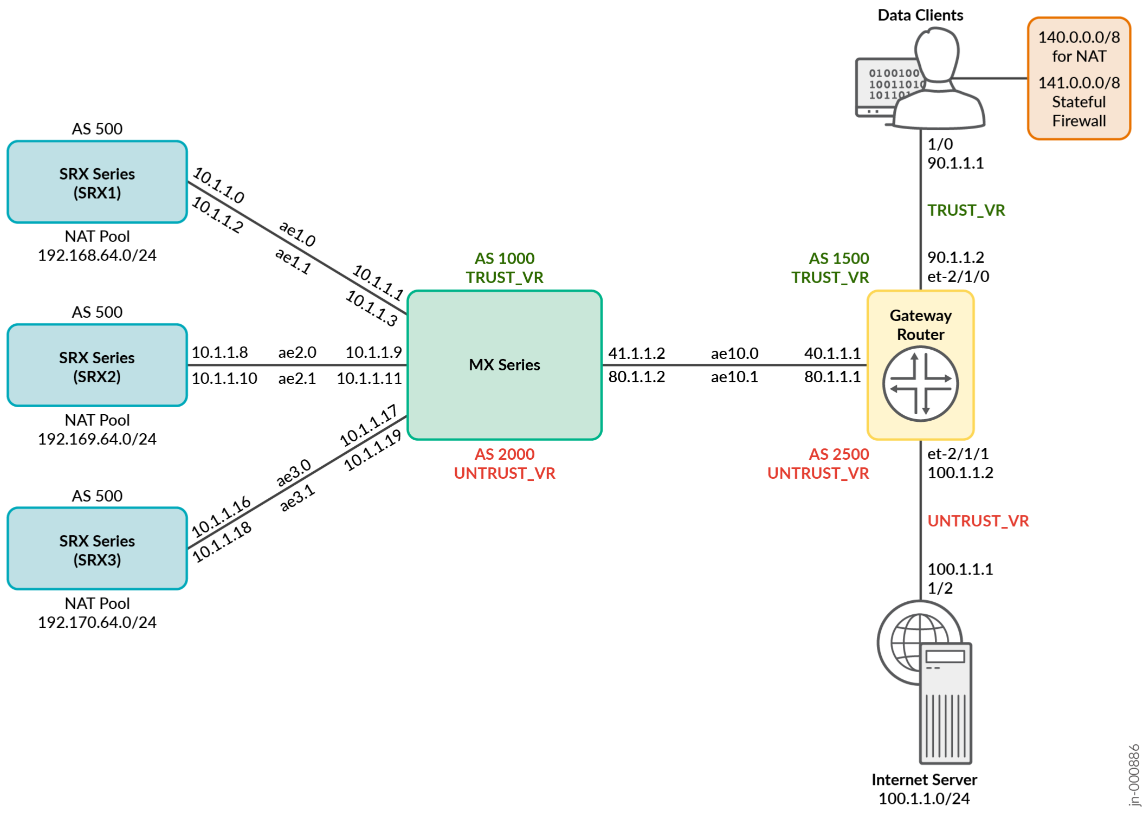


**Figure 24: Route Advertisements for Forward Flow for NAPT44 and Stateful Firewall Services**

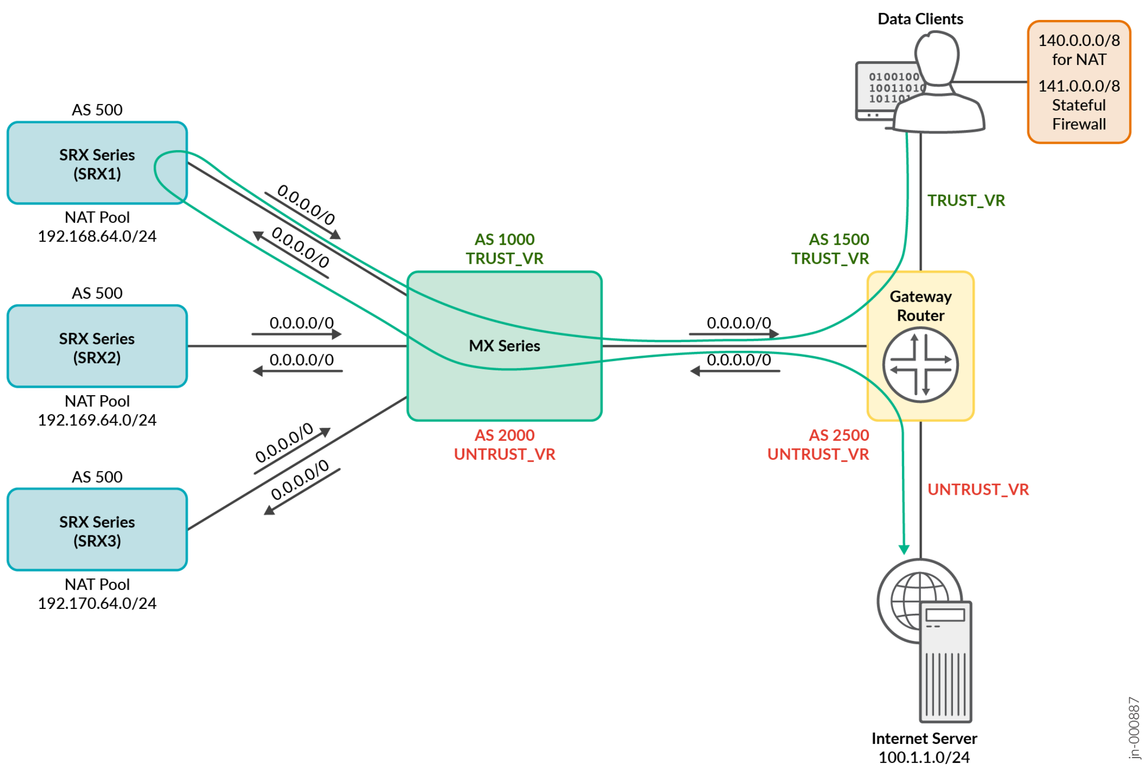


**Figure 25: Route Advertisements for Reverse Flow for Stateful Firewall Services**

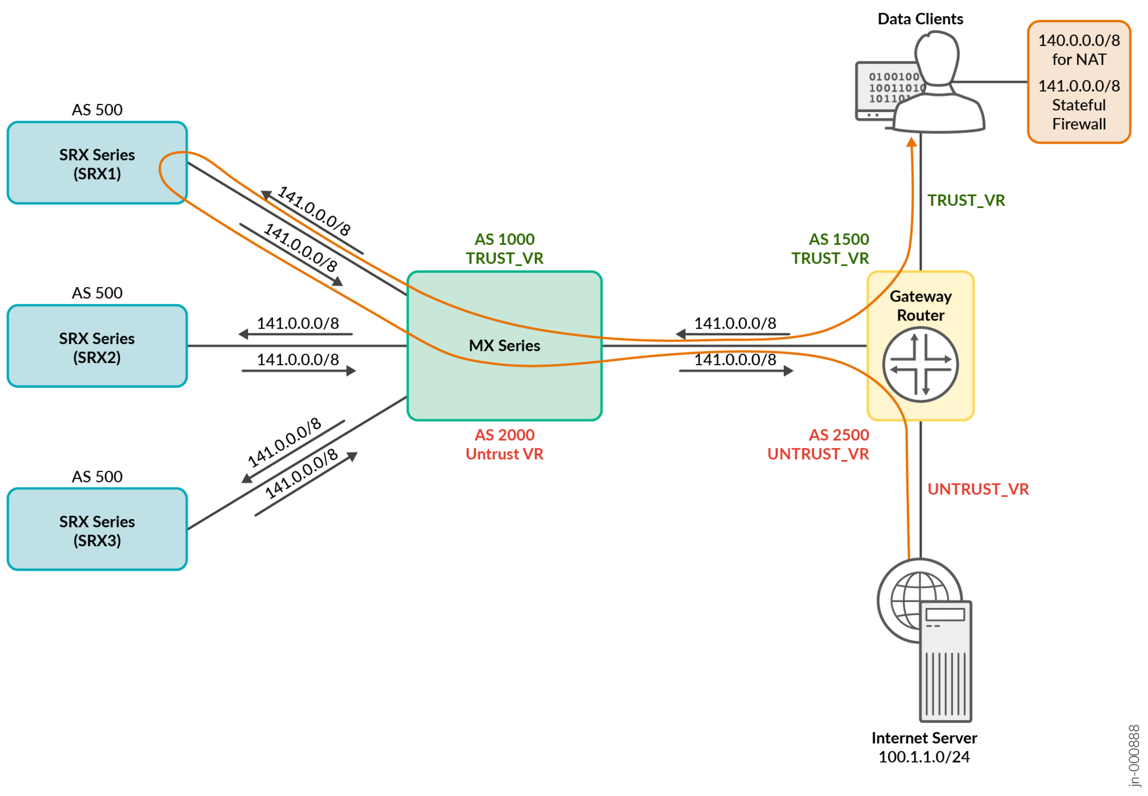
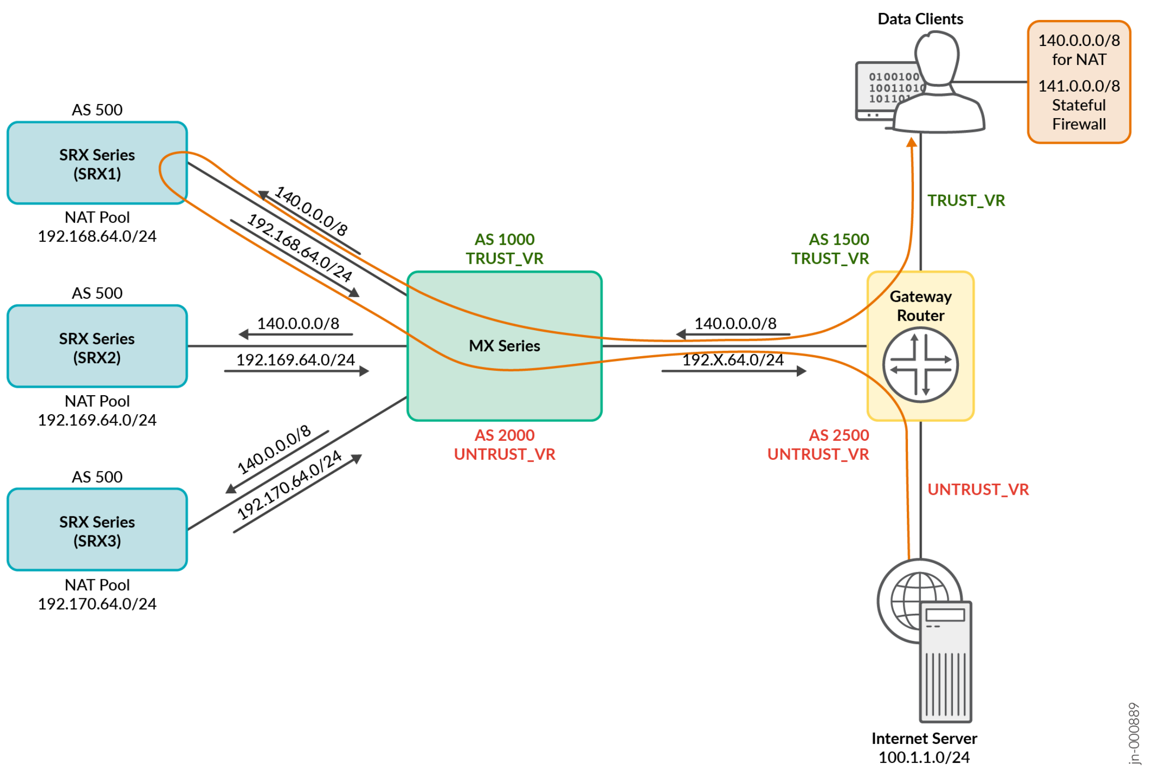


**Figure 26: Route Advertisements for Reverse Flow for NAT44 Services**



## Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

These configurations are captured from a lab environment and are provided for reference only. Actual configurations might vary based on the specific requirements of your environment.

The following items show a list of configuration components for this example:

- Configure MX Series
- Configure the Gateway router
- Configure SRX1
- Configure SRX2

- Configure SRX3

### Configure MX Series

```
[edit]
set interfaces et-0/0/0 gigether-options 802.3ad ae1
set interfaces et-0/0/1 gigether-options 802.3ad ae2
set interfaces et-0/0/2 gigether-options 802.3ad ae3
set interfaces et-0/0/7 gigether-options 802.3ad ae1
set interfaces et-0/0/8 gigether-options 802.3ad ae2
set interfaces et-0/0/9 gigether-options 802.3ad ae3
set interfaces et-0/0/10 gigether-options 802.3ad ae10
set interfaces et-0/0/11 gigether-options 802.3ad ae10
set interfaces et-0/1/0 gigether-options 802.3ad ae10
set interfaces et-0/1/1 gigether-options 802.3ad ae10
set interfaces et-0/1/2 gigether-options 802.3ad ae10
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 1
set interfaces ae1 unit 0 family inet address 10.1.1.1/31
set interfaces ae1 unit 0 family inet6 address 10:1:1::1/127
set interfaces ae1 unit 1 vlan-id 2
set interfaces ae1 unit 1 family inet address 10.1.1.3/31
set interfaces ae1 unit 1 family inet6 address 10:1:1::3/127
set interfaces ae2 vlan-tagging
set interfaces ae2 aggregated-ether-options minimum-links 1
set interfaces ae2 aggregated-ether-options lacp active
set interfaces ae2 aggregated-ether-options lacp periodic fast
set interfaces ae2 unit 0 vlan-id 9
set interfaces ae2 unit 0 family inet address 10.1.1.9/31
set interfaces ae2 unit 0 family inet6 address 10:2:2::1/127
set interfaces ae2 unit 1 vlan-id 10
set interfaces ae2 unit 1 family inet address 10.1.1.11/31
set interfaces ae2 unit 1 family inet6 address 10:2:2::3/127
set interfaces ae3 vlan-tagging
set interfaces ae3 aggregated-ether-options minimum-links 1
set interfaces ae3 aggregated-ether-options lacp active
set interfaces ae3 aggregated-ether-options lacp periodic fast
set interfaces ae3 unit 0 vlan-id 9
set interfaces ae3 unit 0 family inet address 10.1.1.17/31
set interfaces ae3 unit 0 family inet6 address 10:3:3::1/127
set interfaces ae3 unit 1 vlan-id 10
set interfaces ae3 unit 1 family inet address 10.1.1.19/31
set interfaces ae3 unit 1 family inet6 address 10:3:3::3/127
set interfaces ae10 flexible-vlan-tagging
set interfaces ae10 encapsulation flexible-ethernet-services
set interfaces ae10 aggregated-ether-options minimum-links 1
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae10 unit 40 vlan-id 40
set interfaces ae10 unit 40 family inet address 40.1.1.2/30
set interfaces ae10 unit 40 family inet6 address 40:1:1::2/124
set interfaces ae10 unit 80 vlan-id 80
set interfaces ae10 unit 80 family inet address 80.1.1.2/30
set interfaces ae10 unit 80 family inet6 address 80:1:1::2/124
set routing-instances TRUST_VR instance-type virtual-router
set routing-instances TRUST_VR routing-options autonomous-system 1000
set routing-instances TRUST_VR routing-options autonomous-system independent-domain no-attrset
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router type external
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router export
def_route_for_client-2-server
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router peer-as 1500
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router neighbor 40.1.1.1
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 neighbor 10.1.1.0
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 neighbor 10.1.1.8
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx3 neighbor 10.1.1.16
set routing-instances TRUST_VR interface ae1.0
set routing-instances TRUST_VR interface ae2.0
set routing-instances TRUST_VR interface ae3.0
set routing-instances TRUST_VR interface ae10.40
set policy-options policy-statement def_route_for_client-2-server term 1 from protocol bgp
set policy-options policy-statement def_route_for_client-2-server term 1 from route-filter
0.0.0.0/0 exact
set policy-options policy-statement def_route_for_client-2-server term 1 then next-hop self
set policy-options policy-statement def_route_for_client-2-server term 1 then accept
set policy-options policy-statement def_route_for_client-2-server term 2 then reject
set policy-options policy-statement pfe_consistent_hash from route-filter 0.0.0.0/0 exact
set policy-options policy-statement pfe_consistent_hash then load-balance consistent-hash
set policy-options policy-statement pfe_consistent_hash then accept
set policy-options policy-statement trust-to-untrust-export term 1 from protocol bgp
set policy-options policy-statement trust-to-untrust-export term 1 from protocol static
set policy-options policy-statement trust-to-untrust-export term 1 then next-hop self
set policy-options policy-statement trust-to-untrust-export term 1 then accept
set policy-options policy-statement trust-to-untrust-export term 2 then reject
set routing-instances UNTRUST_VR instance-type virtual-router
set routing-instances UNTRUST_VR routing-options autonomous-system 2000
set routing-instances UNTRUST_VR routing-options autonomous-system independent-domain no-attrset
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router export
client_sfw_and_nat_pool_prefix_export
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router peer-as 2500
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection minimum-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection minimum-receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router neighbor 80.1.1.1
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 import
pfe_sfw_return_consistent_hash
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 neighbor 10.1.1.2
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 import
pfe_sfw_return_consistent_hash
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 neighbor 10.1.1.10
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 import
pfe_sfw_return_consistent_hash
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx3 neighbor 10.1.1.18
set routing-instances UNTRUST_VR interface ae1.1
set routing-instances UNTRUST_VR interface ae2.1
set routing-instances UNTRUST_VR interface ae3.1
set routing-instances UNTRUST_VR interface ae10.80
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 from protocol
bgp
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 from route-
filter 140.0.0.0/8 exact
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 from route-
filter 141.0.0.0/8 exact
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 from route-
filter 192.168.64.0/24 exact
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 from route-
filter 192.169.64.0/24 exact
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 from route-
filter 192.170.64.0/24 exact
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 then next-hop
self
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 1 then accept
set policy-options policy-statement client_sfw_and_nat_pool_prefix_export term 2 then reject
set policy-options policy-statement pfe_sfw_return_consistent_hash from route-filter 140.0.0.0/8
exact
set policy-options policy-statement pfe_sfw_return_consistent_hash from route-filter 141.0.0.0/8
exact
set policy-options policy-statement pfe_sfw_return_consistent_hash then load-balance consistent-
hash
set policy-options policy-statement pfe_sfw_return_consistent_hash then accept
set policy-options policy-statement untrust-to-trust-export term 1 from protocol bgp
set policy-options policy-statement untrust-to-trust-export term 1 from protocol static
set policy-options policy-statement untrust-to-trust-export term 1 then next-hop self
set policy-options policy-statement untrust-to-trust-export term 1 then accept
set policy-options policy-statement untrust-to-trust-export term 2 then reject
set policy-options policy-statement pfe_lb_hash term source_hash from route-filter 0.0.0.0/0
exact
set policy-options policy-statement pfe_lb_hash term source_hash then load-balance source-ip-only
set policy-options policy-statement pfe_lb_hash term source_hash then accept
set policy-options policy-statement pfe_lb_hash term dest_hash from route-filter 140.0.0.0/8
exact
set policy-options policy-statement pfe_lb_hash term dest_hash from route-filter 141.0.0.0/8
exact
set policy-options policy-statement pfe_lb_hash term dest_hash then load-balance destination-ip-
only
set policy-options policy-statement pfe_lb_hash term dest_hash then accept
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then load-balance per-packet
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then accept
set routing-options forwarding-table export pfe_lb_hash
```

### Configure Gateway Router

```
[edit]
set interfaces ae10 flexible-vlan-tagging
set interfaces ae10 encapsulation flexible-ethernet-services
set interfaces ae10 aggregated-ether-options minimum-links 1
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae10 unit 40 vlan-id 40
set interfaces ae10 unit 40 family inet address 40.1.1.1/30
set interfaces ae10 unit 80 vlan-id 80
set interfaces ae10 unit 80 family inet address 80.1.1.1/30
set interfaces ae11 flexible-vlan-tagging
set interfaces ae11 encapsulation flexible-ethernet-services
set interfaces ae11 aggregated-ether-options minimum-links 1
set interfaces ae11 aggregated-ether-options lacp active
set interfaces ae11 aggregated-ether-options lacp periodic fast
set interfaces ae11 unit 41 vlan-id 41
set interfaces ae11 unit 41 family inet address 41.1.1.1/30
set interfaces ae11 unit 81 vlan-id 81
set interfaces ae11 unit 81 family inet address 81.1.1.1/30
set interfaces et-2/1/0 unit 0 family inet address 90.1.1.2/30
set routing-instances TRUST_VR instance-type virtual-router
set routing-instances TRUST_VR routing-options autonomous-system 1500
set routing-instances TRUST_VR routing-options static route 140.0.0.0/8 next-hop 90.1.1.1
set routing-instances TRUST_VR routing-options static route 141.0.0.0/8 next-hop 90.1.1.1
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust type external
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust export
client_to_server_export
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust peer-as 1000
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust local-as 1500
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust neighbor 40.1.1.2
set routing-instances TRUST_VR protocols bgp multipath
set routing-instances TRUST_VR interface et-2/1/0.0
set routing-instances TRUST_VR interface ae10.40
set policy-options policy-statement client_to_server_export term 1 from protocol static
set policy-options policy-statement client_to_server_export term 1 from route-filter 141.0.0.0/8
orlonger
set policy-options policy-statement client_to_server_export term 1 from route-filter 140.0.0.0/8
orlonger
set policy-options policy-statement client_to_server_export term 1 then accept
set policy-options policy-statement client_to_server_export term 2 then reject
set routing-instances UNTRUST_VR instance-type virtual-router
set routing-instances UNTRUST_VR routing-options autonomous-system 2500
set routing-instances UNTRUST_VR routing-options static route 0.0.0.0/0 discard
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust type external
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust export
server_to_client_export
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust peer-as 2000
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust local-as 2500
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection minimum-interval 300
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection minimum-receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection multiplier 3
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust neighbor 80.1.1.2
set routing-instances UNTRUST_VR protocols bgp multipath
set routing-instances UNTRUST_VR interface et-2/1/1.0
set routing-instances UNTRUST_VR interface ae10.80
set policy-options policy-statement server_to_client_export term t1 from protocol static
set policy-options policy-statement server_to_client_export term t1 from route-filter 0.0.0.0/0
exact
set policy-options policy-statement server_to_client_export term t1 then accept
set policy-options policy-statement server_to_client_export term t2 then reject
set policy-options policy-statement ECMP_POLICY-LB then load-balance per-packet
set routing-options forwarding-table export ECMP_POLICY-LB
```

### Configure SRX1

```
[edit]
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match source-address sfw_source_prefix_140.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match source-address sfw_source_prefix_141.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY then
permit
set security address-book global address sfw_source_prefix_140.0.0.0/8 140.0.0.0/8
set security address-book global address sfw_source_prefix_141.0.0.0/8 141.0.0.0/8
set security nat source pool vsrx1_nat_pool address 192.168.64.0/24
set security nat source pool vsrx1_nat_pool address-pooling paired
set security nat source rule-set vsrx1_nat_rule-set from zone vr-1_trust_zone
set security nat source rule-set vsrx1_nat_rule-set to zone vr-1_untrust_zone
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match source-address
140.0.0.0/8
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match destination-
address 0.0.0.0/0
```

```
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match application any
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 then source-nat pool
vsrx1_nat_pool
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 1
set interfaces ae1 unit 0 family inet address 10.1.1.0/31
set interfaces ae1 unit 0 family inet6 address 10:1:1::0/127
set interfaces ae1 unit 1 vlan-id 2
set interfaces ae1 unit 1 family inet address 10.1.1.2/31
set interfaces ae1 unit 1 family inet6 address 10:1:1::2/127
set routing-instances VR-1 instance-type virtual-router
set routing-instances VR-1 routing-options static route 192.168.64.0/24 discard
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST export trust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.1 peer-as 1000
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST export untrust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.3 peer-as 2000
set routing-instances VR-1 interface ae1.0
set routing-instances VR-1 interface ae1.1
set policy-options policy-statement trust_export_policy term 1 from protocol bgp
set policy-options policy-statement trust_export_policy term 1 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement trust_export_policy term 1 then next-hop self
set policy-options policy-statement trust_export_policy term 1 then accept
set policy-options policy-statement trust_export_policy term 2 then reject
set policy-options policy-statement untrust_export_policy term 1 from protocol bgp
set policy-options policy-statement untrust_export_policy term 1 from route-filter 141.0.0.0/8
orlonger
set policy-options policy-statement untrust_export_policy term 1 then accept
set policy-options policy-statement untrust_export_policy term 2 from protocol static
set policy-options policy-statement untrust_export_policy term 2 from route-filter
192.168.64.0/24 orlonger
set policy-options policy-statement untrust_export_policy term 2 then accept
set policy-options policy-statement untrust_export_policy term 3 then reject
set policy-options policy-statement ecmp_policy_lab then load-balance per-packet
set routing-options forwarding-table export ecmp_policy_lab
```

Configure SRX2

```
[edit]
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match source-address sfw_source_prefix_140.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match source-address sfw_source_prefix_141.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY then
permit
set security address-book global address sfw_source_prefix_140.0.0.0/8 140.0.0.0/8
set security address-book global address sfw_source_prefix_141.0.0.0/8 141.0.0.0/8
set security nat source pool vsrx1_nat_pool address 192.169.64.0/24
set security nat source pool vsrx1_nat_pool address-pooling paired
set security nat source rule-set vsrx1_nat_rule-set from zone vr-1_trust_zone
set security nat source rule-set vsrx1_nat_rule-set to zone vr-1_untrust_zone
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match source-address
140.0.0.0/8
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match destination-
address 0.0.0.0/0
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match application any
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 then source-nat pool
vsrx1_nat_pool
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 9
set interfaces ae1 unit 0 family inet address 10.1.1.8/31
set interfaces ae1 unit 0 family inet6 address 10:2:2::0/127
set interfaces ae1 unit 1 vlan-id 10
set interfaces ae1 unit 1 family inet address 10.1.1.10/31
set interfaces ae1 unit 1 family inet6 address 10:2:2::2/127
set routing-instances VR-1 instance-type virtual-router
set routing-instances VR-1 routing-options static route 192.169.64.0/24 discard
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST export trust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.9 peer-as 1000
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST export untrust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.11 peer-as 2000
set routing-instances VR-1 interface ae1.0
set routing-instances VR-1 interface ae1.1
set policy-options policy-statement trust_export_policy term 1 from protocol bgp
set policy-options policy-statement trust_export_policy term 1 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement trust_export_policy term 1 then next-hop self
set policy-options policy-statement trust_export_policy term 1 then accept
set policy-options policy-statement trust_export_policy term 2 then reject
set policy-options policy-statement untrust_export_policy term 1 from protocol bgp
set policy-options policy-statement untrust_export_policy term 1 from route-filter 141.0.0.0/8
orlonger
set policy-options policy-statement untrust_export_policy term 1 then accept
set policy-options policy-statement untrust_export_policy term 2 from protocol static
set policy-options policy-statement untrust_export_policy term 2 from route-filter
192.169.64.0/24 orlonger
set policy-options policy-statement untrust_export_policy term 2 then accept
set policy-options policy-statement untrust_export_policy term 3 then reject
set policy-options policy-statement ecmp_policy_lab then load-balance per-packet
set routing-options forwarding-table export ecmp_policy_lab
```

Configure SRX3

```
[edit]
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match source-address sfw_source_prefix_140.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match source-address sfw_source_prefix_141.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match application any
```

```
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY then
permit
set security address-book global address sfw_source_prefix_140.0.0.0/8 140.0.0.0/8
set security address-book global address sfw_source_prefix_141.0.0.0/8 141.0.0.0/8
set security nat source pool vsrx1_nat_pool address 192.170.64.0/24
set security nat source pool vsrx1_nat_pool address-pooling paired
set security nat source rule-set vsrx1_nat_rule-set from zone vr-1_trust_zone
set security nat source rule-set vsrx1_nat_rule-set to zone vr-1_untrust_zone
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match source-address
140.0.0.0/8
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match destination-
address 0.0.0.0/0
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match application any
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 then source-nat pool
vsrx1_nat_pool
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 9
set interfaces ae1 unit 0 family inet address 10.1.1.16/31
set interfaces ae1 unit 0 family inet6 address 10:3:3::0/127
set interfaces ae1 unit 1 vlan-id 10
set interfaces ae1 unit 1 family inet address 10.1.1.18/31
set interfaces ae1 unit 1 family inet6 address 10:3:3::2/127
set routing-instances VR-1 instance-type virtual-router
set routing-instances VR-1 routing-options static route 192.170.64.0/24 discard
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST export trust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.17 peer-as 1000
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST export untrust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
```

```
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.19 peer-as 2000
set routing-instances VR-1 interface ae1.0
set routing-instances VR-1 interface ae1.1
set policy-options policy-statement trust_export_policy term 1 from protocol bgp
set policy-options policy-statement trust_export_policy term 1 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement trust_export_policy term 1 then next-hop self
set policy-options policy-statement trust_export_policy term 1 then accept
set policy-options policy-statement trust_export_policy term 2 then reject
set policy-options policy-statement untrust_export_policy term 1 from protocol bgp
set policy-options policy-statement untrust_export_policy term 1 from route-filter 141.0.0.0/8
orlonger
set policy-options policy-statement untrust_export_policy term 1 then accept
set policy-options policy-statement untrust_export_policy term 2 from protocol static
set policy-options policy-statement untrust_export_policy term 2 from route-filter
192.170.64.0/24 orlonger
set policy-options policy-statement untrust_export_policy term 2 then accept
set policy-options policy-statement untrust_export_policy term 3 then reject
set policy-options policy-statement ecmp_policy_lab then load-balance per-packet
set routing-options forwarding-table export ecmp_policy_lab
```

## Verification

The following items highlight a list of show commands used to verify the feature in this example.

- Verify MX Series configuration
- Verify SRX1 configuration
- Verify SRX2 configuration
- Verify SRX3 configuration

### Verify MX Series Configuration

```
user@MX304> show route table TRUST_VR.inet 0.0.0.0/0 exact active-path
              TRUST_VR.inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 04:17:13, localpref 100
                                    AS path: 500 2000 2500 I, validation-state: unverified
                                 >  to 10.1.1.0 via ae1.0
                                    to 10.1.1.8 via ae2.0
                                    to 10.1.1.16 via ae3.0
```

```
user@MX304> show route table TRUST_VR.inet 0.0.0.0/0 exact active-path extensive
              TRUST_VR.inet.0: 11 destinations, 13 routes (11 active, 0 holddown, 0 hidden)
              0.0.0.0/0 (3 entries, 1 announced)
              TSI:
              KRT in-kernel 0.0.0.0/0 -> {list:10.1.1.0, 10.1.1.8, 10.1.1.16 Flags source ip 
load-balance}
              Page 0 idx 1, (group MX-to-TRUST_GW_Router type External) Type 1 val 0x12b05d48 
(adv_entry)
                Advertised metrics:
                  Flags: Nexthop Change
                  Nexthop: Self
                  AS path: [1000] 500 2000 2500 I
                  Communities:
                 Advertise: 00000001
              Path 0.0.0.0
              from 10.1.1.0
              Vector len 4.  Val: 1
                     *BGP    Preference: 170/-101
                             Next hop type: Router, Next hop index: 0
                             Address: 0xf91865c
                             Next-hop reference count: 2, Next-hop session id: 0
                             Kernel Table Id: 0
                             Source: 10.1.1.0
                             Next hop: 10.1.1.0 via ae1.0, selected
                             Session Id: 0
                             Next hop: 10.1.1.8 via ae2.0
                             Session Id: 0
                             Next hop: 10.1.1.16 via ae3.0
                             Session Id: 0
                             State: <Active Ext LoadBalConsistentHash>
                             Local AS:  1000 Peer AS:   500
                             Age: 4:17:17
                             Validation State: unverified
                             Task: BGP_500_1000.10.1.1.0
```

```
                                Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
                                AS path: 500 2000 2500 I
                                Accepted Multipath
                                Localpref: 100
                                Router ID: 10.1.1.0
                                Thread: junos-main
```

```
user@MX304> show route table UNTRUST_VR.inet 141/8 active-path
              UNTRUST_VR.inet.0: 13 destinations, 15 routes (13 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 04:18:15, localpref 100, from 10.1.1.2
                                   AS path: 500 1000 1500 I, validation-state: unverified
                                    to 10.1.1.2 via ae1.1
                                    to 10.1.1.10 via ae2.1
                                 >  to 10.1.1.18 via ae3.1
```

```
user@MX304> show route table UNTRUST_VR.inet 141/8 active-path extensive
              UNTRUST_VR.inet.0: 13 destinations, 15 routes (13 active, 0 holddown, 0 hidden)
              141.0.0.0/8 (3 entries, 1 announced)
              TSI:
              KRT in-kernel 141.0.0.0/8 -> {list:10.1.1.2, 10.1.1.10, 10.1.1.18 Flags
destination ip load-balance}
              Page 0 idx 1, (group MX-to-UNTRUST_GW_Router type External) Type 1 val 0x12b073d0
(adv_entry)
                Advertised metrics:
                  Flags: Nexthop Change
                  Nexthop: Self
                  AS path: [2000] 500 1000 1500 I
                  Communities:
                 Advertise: 00000001
              Path 141.0.0.0
              from 10.1.1.2
              Vector len 4.  Val: 1
                     *BGP    Preference: 170/-101
                             Next hop type: Router, Next hop index: 0
                             Address: 0xf918b24
                             Next-hop reference count: 2, Next-hop session id: 0
                             Kernel Table Id: 0
                             Source: 10.1.1.2
                             Next hop: 10.1.1.2 via ae1.1
```

```
                              Session Id: 0
                              Next hop: 10.1.1.10 via ae2.1
                              Session Id: 0
                              Next hop: 10.1.1.18 via ae3.1, selected
                              Session Id: 0
                              State: <Active Ext LoadBalConsistentHash>
                              Local AS:  2000 Peer AS:   500
                              Age: 4:18:17
                              Validation State: unverified
                              Task: BGP_500_2000.10.1.1.2
                              Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
                              AS path: 500 1000 1500 I
                              Accepted Multipath
                              Localpref: 100
                              Router ID: 10.1.1.0
                              Thread: junos-main
```

```
user@MX304> show route table UNTRUST_VR.inet 192/8
              UNTRUST_VR.inet.0: 13 destinations, 15 routes (13 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              192.168.64.0/24    *[BGP/170] 03:31:16, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.2 via ae1.1
              192.169.64.0/24    *[BGP/170] 03:31:21, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.10 via ae2.1
              192.170.64.0/24    *[BGP/170] 03:31:27, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.18 via ae3.1
```

```
user@MX304> show bgp summary
              Warning: License key missing; requires 'bgp' license
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 8 Peers: 8 Down peers: 0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              10.1.1.0                500        606        593       0       1     4:31:25
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.2                500        606        593        0        1      4:31:25
Establ
               UNTRUST_VR.inet.0: 2/2/2/0
              10.1.1.8                500        588        576        0        1      4:23:27
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.10               500        589        576        0        1      4:23:27
Establ
               UNTRUST_VR.inet.0: 2/2/2/0
              10.1.1.16               500        578        566        0        1      4:18:56
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.18               500        579        566        0        1      4:18:56
Establ
               UNTRUST_VR.inet.0: 2/2/2/0
              40.1.1.1               1500      12472      12247        0        1 3d 23:18:01
Establ
               TRUST_VR.inet.0: 2/2/2/0
              80.1.1.1               2500      12471      12257        0        1 3d 23:18:01
Establ
               UNTRUST_VR.inet.0: 1/1/1/0
```

```
user@MX304> show bfd session
              Warning: License key missing; requires 'bfd-liveness-detection' license
                                                           Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              10.1.1.0                 Up        ae1.0          0.900     0.300        3
              10.1.1.2                 Up        ae1.1          0.900     0.300        3
              10.1.1.8                 Up        ae2.0          0.900     0.300        3
              10.1.1.10                Up        ae2.1          0.900     0.300        3
              10.1.1.16                Up        ae3.0          0.900     0.300        3
              10.1.1.18                Up        ae3.1          0.900     0.300        3
              40.1.1.1                 Up        ae10.40        0.900     0.300        3
              80.1.1.1                 Up        ae10.80        0.900     0.300        3
              8 sessions, 8 clients
              Cumulative transmit rate 26.7 pps, cumulative receive rate 26.7 pps
```

### Verify SRX1 Configuration

```
user@SRX1> show bgp summary
Threading mode: BGP I/O
```

```
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.1.1.1               1000        596        606       0       0     4:32:18 Establ
  VR-1.inet.0: 2/2/2/0
10.1.1.3               2000        596        606       0       0     4:32:18 Establ
  VR-1.inet.0: 1/1/1/0
```

```
user@SRX1> **show bfd session**
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.1.1.1                 Up        ae1.0          0.900     0.300        3
10.1.1.3                 Up        ae1.1          0.900     0.300        3
2 sessions, 2 clients
Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
```

```
user@SRX1> **show route 0.0.0.0/0 exact**
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[BGP/170] 04:33:13, localpref 100
                      AS path: 2000 2500 I, validation-state: unverified
                    >  to 10.1.1.3 via ae1.1
```

```
user@SRX1> **show route 140/8 exact**
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
140.0.0.0/8        *[BGP/170] 04:33:20, localpref 100
                      AS path: 1000 1500 I, validation-state: unverified
                    >  to 10.1.1.1 via ae1.0
```

```
user@SRX1> **show route 141/8 exact**
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
141.0.0.0/8        *[BGP/170] 04:33:23, localpref 100
                      AS path: 1000 1500 I, validation-state: unverified
                    >  to 10.1.1.1 via ae1.0
```

### Verify SRX2 Configuration

```
user@SRX2> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.1.1.9               1000        601        611       0       0     4:34:20 Establ
  VR-1.inet.0: 2/2/2/0
10.1.1.11              2000        600        611       0       0     4:34:20 Establ
  VR-1.inet.0: 1/1/1/0
```

```
user@SRX2> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.1.1.9                 Up        ae1.0          0.900     0.300        3
10.1.1.11                Up        ae1.1          0.900     0.300        3
2 sessions, 2 clients
Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
```

```
user@SRX2> show route 0.0.0.0/0 exact
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[BGP/170] 04:34:30, localpref 100
                      AS path: 2000 2500 I, validation-state: unverified
                    >  to 10.1.1.11 via ae1.1
```

```
user@SRX2> show route 140/8 exact
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
140.0.0.0/8        *[BGP/170] 04:36:20, localpref 100
                      AS path: 1000 1500 I, validation-state: unverified
                    >  to 10.1.1.9 via ae1.0
```

```
user@SRX2> show route 141/8 exact
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
141.0.0.0/8        *[BGP/170] 04:36:24, localpref 100
                      AS path: 1000 1500 I, validation-state: unverified
                    >  to 10.1.1.9 via ae1.0
```

### Verify SRX3 Configuration

```
user@SRX3> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.1.1.17              1000        622        632       0       0     4:44:11 Establ
  VR-1.inet.0: 2/2/2/0
10.1.1.19              2000        622        634       0       0     4:44:11 Establ
  VR-1.inet.0: 1/1/1/0
```

```
user@SRX3> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.1.1.17                Up        ae1.0          0.900     0.300        3
10.1.1.19                Up        ae1.1          0.900     0.300        3
2 sessions, 2 clients
Cumulative transmit rate 6.7 pps, cumulative receive rate 6.7 pps
```

```
user@SRX3> show route 0.0.0.0/0 exact
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          *[BGP/170] 04:44:23, localpref 100
                      AS path: 2000 2500 I, validation-state: unverified
                    >  to 10.1.1.19 via ae1.1
```

```
user@SRX3> show route 140/8 exact
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
140.0.0.0/8        *[BGP/170] 04:44:51, localpref 100
```

```
                      AS path: 1000 1500 I, validation-state: unverified
                    >  to 10.1.1.17 via ae1.0
```

```
user@SRX3> show route 141/8 exact
VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
141.0.0.0/8        *[BGP/170] 04:44:55, localpref 100
                      AS path: 1000 1500 I, validation-state: unverified
                    >  to 10.1.1.17 via ae1.0
```

# Example: Dual MX Series (ECMP Based Consistent Hashing) and Scaled-Out SRX Series Firewalls (Multinode HA) for NAT and Stateful Firewall

**SUMMARY**

In this configuration, you'll learn to set up a dual MX Series with scaled-out SRX Series Firewalls in Multinode HA mode for NAT and stateful firewall services.

**IN THIS SECTION**

## Overview

Table 23 on page 135 shows the deployment components used in the example.

**Table 23: Deployment Details**

| CSDS Components | Details |
|---|---|
| Forwarding Layer | MX304 with Junos OS Release 23.4R1 or later |

**Table 23: Deployment Details *(Continued)***

| CSDS Components | Details |
|---|---|
| Services Layer | vSRX 3.0 with Junos OS Release 23.4R1 or later |
| Redundancy | Dual MX Series in Active/Standby (SRD) for redundancy and ECMP based Consistent Hashing for load balancing.<br><br>SRX Series Firewalls in Multinode HA (Active/Backup) and session synchronization. |
| Features | NAPT44 and stateful firewall (IPv4 Support) |
| Additional Component | Gateway router for TRUST and UNTRUST networks. The example uses MX Series. You can use any device. |

See Table 24 on page 136 and Table 25 on page 137 for traffic flow in Multinode HA pairs.

**Table 24: Traffic Flows on MNHA Pair for NAT**

| Feature | Traffic Flow Component | IP Address and Port Number |
|---|---|---|
| NAPT44 on SRX Series Firewalls for MNHA Pair 1<br><br>(SRX1-ACT1, SRX1-ACT2) | Original source data client | 140.0.0.0/8 and port 22279 |
| | Original destination Internet server | 100.1.1.0/24 and port 70 |
| | After NAT source | 192.168.64.0/24 and port 2480 |
| | After NAT destination | 100.1.1.0/24 and port 70 |
| NAPT44 on SRX Series Firewalls for MNHA Pair 2<br><br>(SRX1-STA1, SRX1-STA2) | Original source data client | 140.0.0.0/8 and port 22279 |
| | Original destination Internet server | 100.1.1.0/24 and port 70 |
| | After NAT source | 192.169.64.0/24 and port 2480 |
| | After NAT destination | 100.1.1.0/24 and port 70 |

**Table 25: Traffic Flows on MNHA Pair for Stateful Firewall Services**

| Feature | Traffic Flow Component | IP Address |
|---|---|---|
| Stateful firewall services on SRX Series Firewalls for MNHA Pair 1<br><br>(SRX1-ACT1, SRX1-ACT2) | Source data client | 141.0.0.0/8 |
| | Destination Internet server | 100.1.1.0/24 |
| | SRX Series with stateful firewall - Source | 141.0.0.0/8 |
| | SRX Series with stateful firewall - Destination | 100.1.1.0/24 |
| Stateful firewall services on SRX Series Firewalls for MNHA Pair 2<br><br>(SRX1-STA1, SRX1-STA2) | Source data client | 141.0.0.0/8 |
| | Destination Internet server | 100.1.1.0/24 |
| | SRX Series with stateful firewall - Source | 141.0.0.0/8 |
| | SRX Series with stateful firewall - Destination | 100.1.1.0/24 |

## Topology Illustration

Figure 27: Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (MNHA) for NAT and Stateful Firewall Services

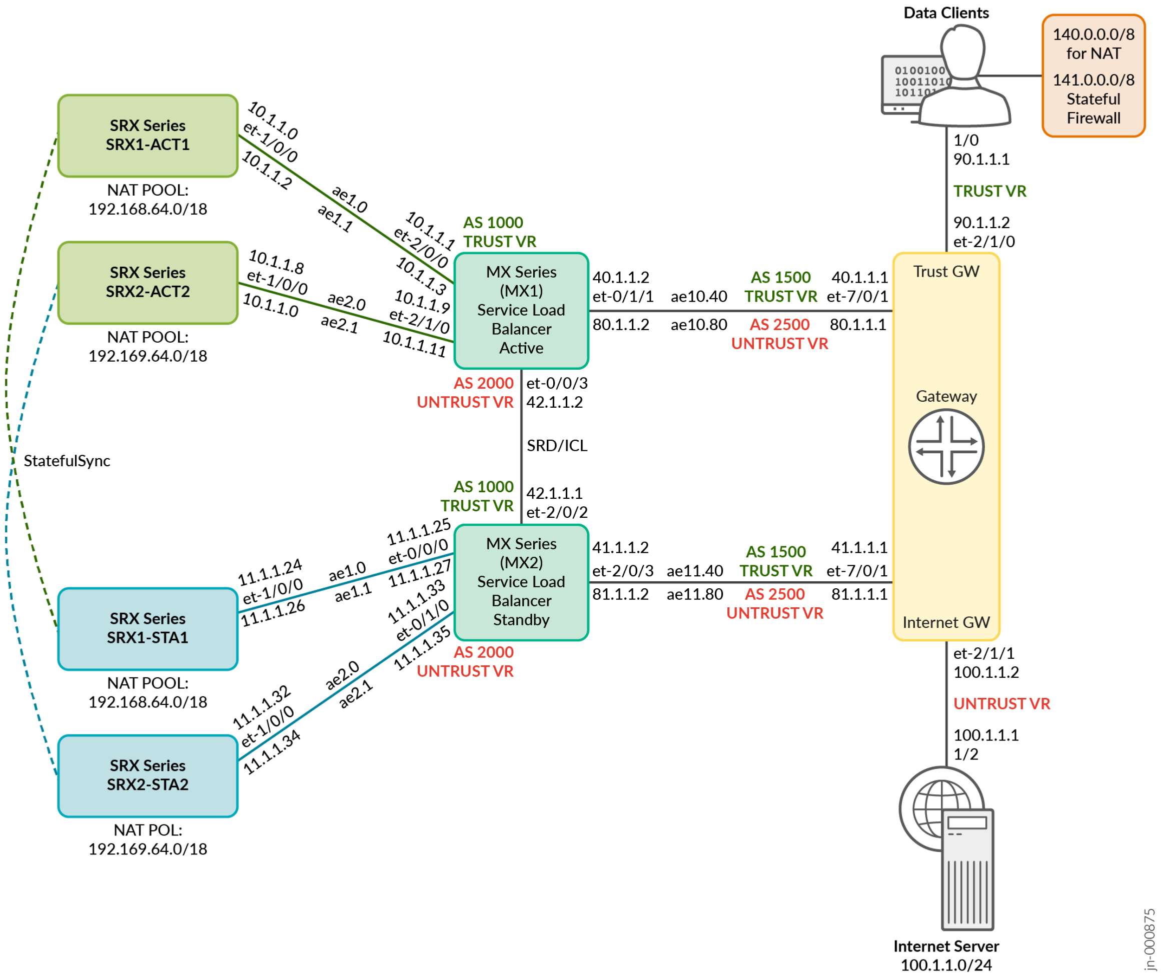


Figure 28: Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (MNHA) Stateful Synchronization Flow

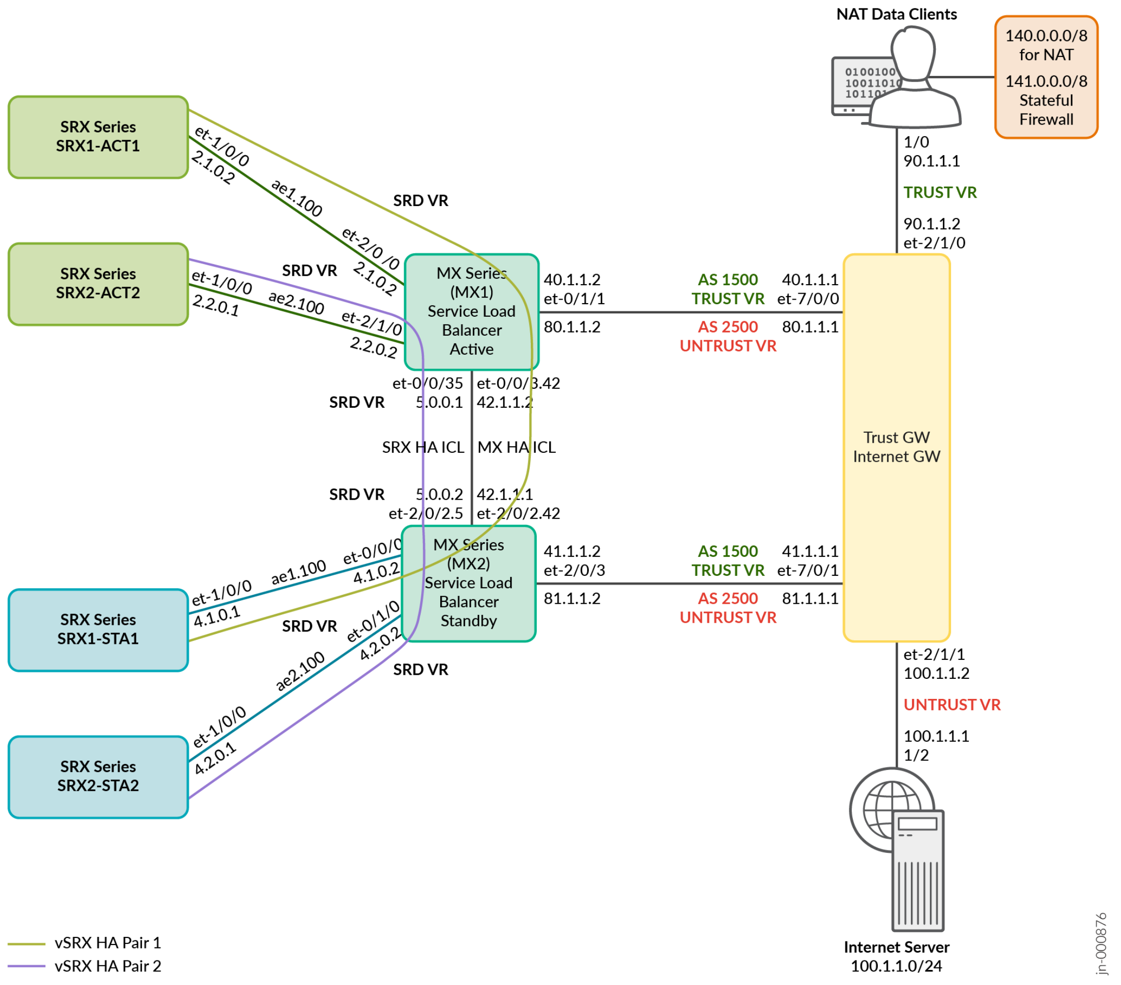


**Figure 29: Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (MNHA) NAT Traffic Flow**

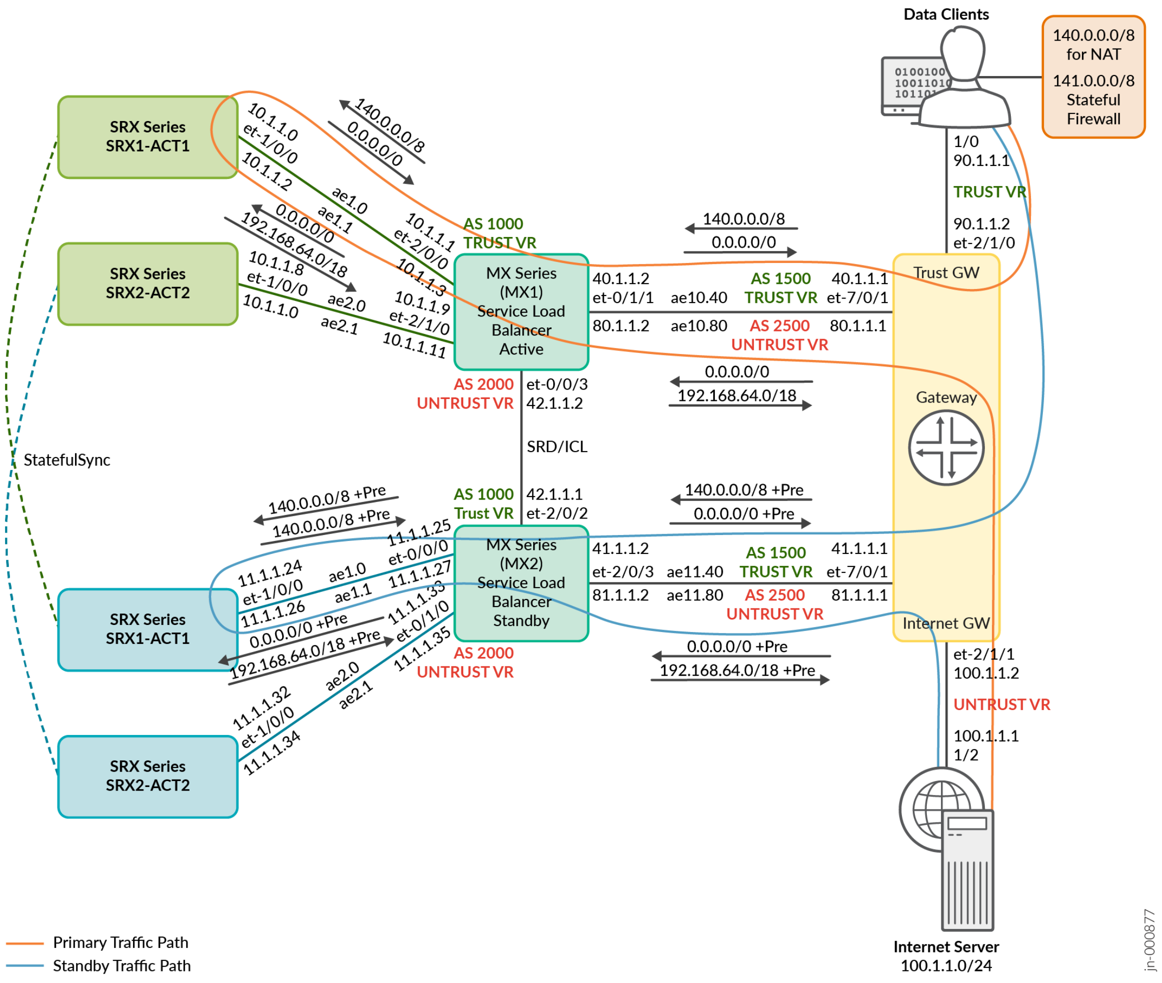
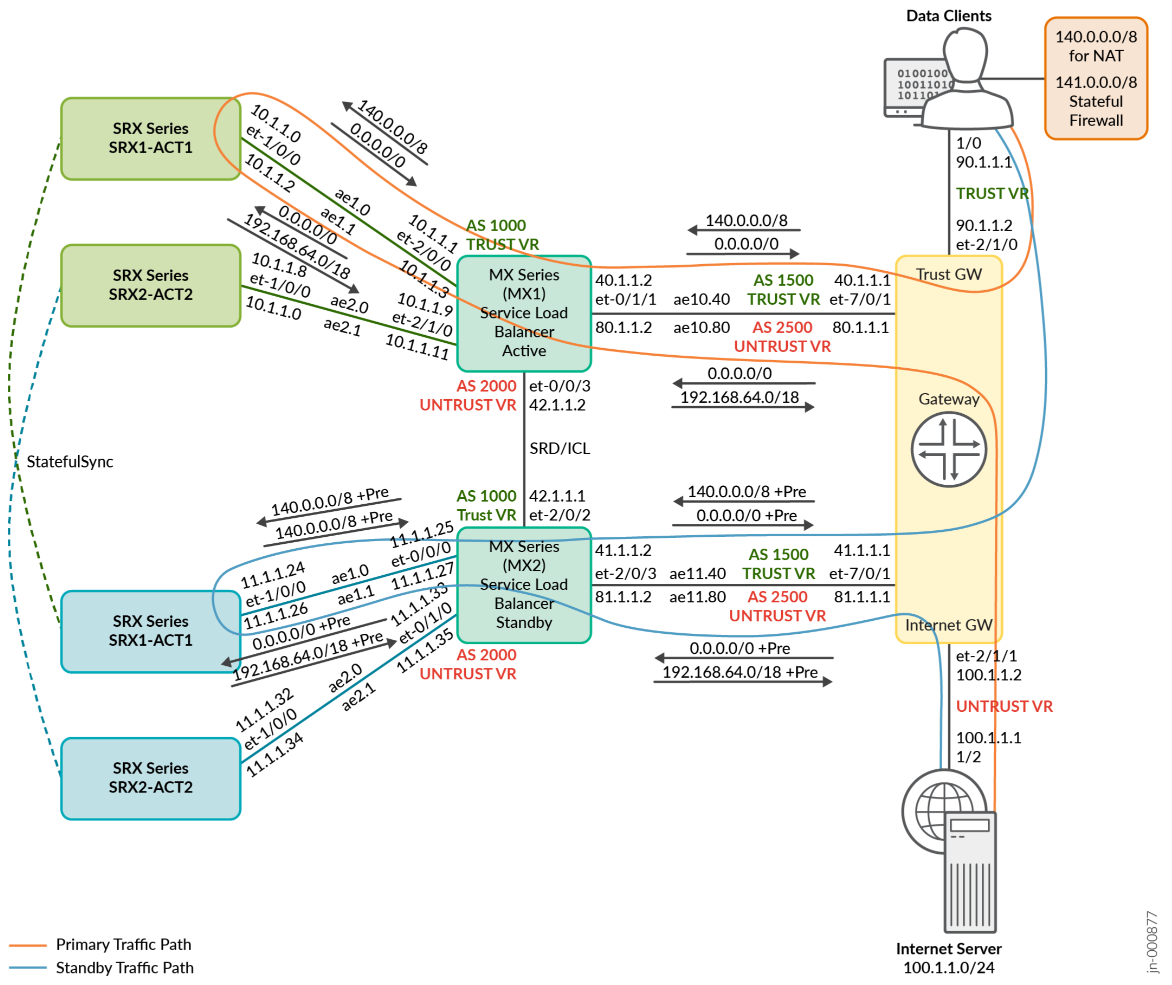


**Figure 30: Dual MX Series (ECMP based Consistent Hashing) and Scaled-Out SRX Series Firewalls (MNHA) Stateful Firewall Traffic Flow**



## Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter commit from configuration mode.

These configurations are captured from a lab environment and are provided for reference only. Actual configurations might vary based on the specific requirements of your environment.

The following items show a list of configuration components for this example:

- Configure MX Series (active node)
- Configure MX Series (standby node)
- Configure the Gateway router
- Configure MNHA Pair 1 (active node)
- Configure MNHA Pair 1 (backup node)
- Configure MNHA Pair 2 (active node)
- Configure MNHA Pair 2 (backup node)

### Configure MX Series (Active Node)

```
[edit]
set routing-instances TRUST_VR instance-type virtual-router
set routing-instances TRUST_VR routing-options autonomous-system 1000
set routing-instances TRUST_VR routing-options autonomous-system independent-domain no-attrset
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router type external
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router export
def_route_for_client-2-server
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router peer-as 1500
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router neighbor 40.1.1.1
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 neighbor 10.1.1.0
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 neighbor 10.1.1.8
set routing-instances TRUST_VR interface ae1.0
set routing-instances TRUST_VR interface ae2.0
set routing-instances TRUST_VR interface ae10.40
set interfaces et-0/0/0 gigether-options 802.3ad ae1
set interfaces et-0/0/7 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 1
set interfaces ae1 unit 0 family inet address 10.1.1.1/31
set interfaces et-0/0/1 gigether-options 802.3ad ae2
set interfaces et-0/0/8 gigether-options 802.3ad ae2
set interfaces ae2 vlan-tagging
set interfaces ae2 aggregated-ether-options minimum-links 1
set interfaces ae2 aggregated-ether-options lacp active
set interfaces ae2 aggregated-ether-options lacp periodic fast
set interfaces ae2 unit 0 vlan-id 9
set interfaces ae2 unit 0 family inet address 10.1.1.9/31
set interfaces et-0/1/1 gigether-options 802.3ad ae10
set interfaces et-0/1/2 gigether-options 802.3ad ae10
set interfaces ae10 flexible-vlan-tagging
set interfaces ae10 encapsulation flexible-ethernet-services
set interfaces ae10 aggregated-ether-options minimum-links 1
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae10 unit 40 vlan-id 40
set interfaces ae10 unit 40 family inet address 40.1.1.2/30
set policy-options policy-statement def_route_for_client-2-server term 1 from protocol bgp
set policy-options policy-statement def_route_for_client-2-server term 1 from route-filter
```

```
0.0.0.0/0 exact
set policy-options policy-statement def_route_for_client-2-server term 1 from condition
1_ROUTE_EXISTS
set policy-options policy-statement def_route_for_client-2-server term 1 then next-hop self
set policy-options policy-statement def_route_for_client-2-server term 1 then accept
set policy-options policy-statement def_route_for_client-2-server term 2 from protocol bgp
set policy-options policy-statement def_route_for_client-2-server term 2 from route-filter
0.0.0.0/0 exact
set policy-options policy-statement def_route_for_client-2-server term 2 then as-path-prepend
"1000 1000 1000 1000"
set policy-options policy-statement def_route_for_client-2-server term 2 then next-hop self
set policy-options policy-statement def_route_for_client-2-server term 2 then accept
set policy-options policy-statement def_route_for_client-2-server term 3 then reject
set policy-options policy-statement pfe_consistent_hash from route-filter 0.0.0.0/0 exact
set policy-options policy-statement pfe_consistent_hash then load-balance consistent-hash
set policy-options policy-statement pfe_consistent_hash then accept
set policy-options policy-statement trust-to-untrust-export term 1 from protocol bgp
set policy-options policy-statement trust-to-untrust-export term 1 from protocol static
set policy-options policy-statement trust-to-untrust-export term 1 then next-hop self
set policy-options policy-statement trust-to-untrust-export term 1 then accept
set policy-options policy-statement trust-to-untrust-export term 2 then reject
set routing-instances UNTRUST_VR instance-type virtual-router
set routing-instances UNTRUST_VR routing-options autonomous-system 2000
set routing-instances UNTRUST_VR routing-options autonomous-system independent-domain no-attrset
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router export
client_sfw_prefix_export
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router peer-as 2500
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection minimum-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection minimum-receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router bfd-liveness-
detection multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-UNTRUST_GW_Router neighbor 80.1.1.1
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 import
pfe_sfw_return_consistent_hash
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx1 neighbor 10.1.1.2
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 type external
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 import
pfe_sfw_return_consistent_hash
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 export untrust-to-trust-export
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 peer-as 500
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 local-as 2000
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 multipath
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection
multiplier 3
set routing-instances UNTRUST_VR protocols bgp group MX-to-vsrx2 neighbor 10.1.1.10
set routing-instances UNTRUST_VR interface ae1.1
set routing-instances UNTRUST_VR interface ae2.1
set routing-instances UNTRUST_VR interface ae10.80
set interfaces ae1 unit 1 vlan-id 2
set interfaces ae1 unit 1 family inet address 10.1.1.3/31
set interfaces ae2 unit 1 vlan-id 10
set interfaces ae2 unit 1 family inet address 10.1.1.11/31
set interfaces ae10 unit 80 vlan-id 80
set interfaces ae10 unit 80 family inet address 80.1.1.2/30
set policy-options policy-statement client_sfw_prefix_export term 1 from protocol bgp
set policy-options policy-statement client_sfw_prefix_export term 1 from route-filter
140.0.0.0/8 exact
set policy-options policy-statement client_sfw_prefix_export term 1 from route-filter
141.0.0.0/8 exact
set policy-options policy-statement client_sfw_prefix_export term 1 from condition 1_ROUTE_EXISTS
set policy-options policy-statement client_sfw_prefix_export term 1 then next-hop self
set policy-options policy-statement client_sfw_prefix_export term 1 then accept
set policy-options policy-statement client_sfw_prefix_export term 2 from protocol bgp
set policy-options policy-statement client_sfw_prefix_export term 2 from route-filter
140.0.0.0/8 exact
set policy-options policy-statement client_sfw_prefix_export term 2 from route-filter
141.0.0.0/8 exact
```

```
set policy-options policy-statement client_sfw_prefix_export term 2 then as-path-prepend "2000
2000 2000 2000"
set policy-options policy-statement client_sfw_prefix_export term 2 then next-hop self
set policy-options policy-statement client_sfw_prefix_export term 2 then accept
set policy-options policy-statement client_sfw_prefix_export term 3 then reject
set policy-options policy-statement pfe_sfw_return_consistent_hash from route-filter 140.0.0.0/8
exact
set policy-options policy-statement pfe_sfw_return_consistent_hash from route-filter 141.0.0.0/8
exact
set policy-options policy-statement pfe_sfw_return_consistent_hash then load-balance consistent-
hash
set policy-options policy-statement pfe_sfw_return_consistent_hash then accept
set policy-options policy-statement untrust-to-trust-export term 1 from protocol bgp
set policy-options policy-statement untrust-to-trust-export term 1 from protocol static
set policy-options policy-statement untrust-to-trust-export term 1 then next-hop self
set policy-options policy-statement untrust-to-trust-export term 1 then accept
set policy-options policy-statement untrust-to-trust-export term 2 then reject
set forwarding-options enhanced-hash-key symmetric
set routing-options forwarding-table export pfe_lb_hash
set policy-options policy-statement pfe_lb_hash term source_hash from route-filter 0.0.0.0/0
exact
set policy-options policy-statement pfe_lb_hash term source_hash then load-balance source-ip-only
set policy-options policy-statement pfe_lb_hash term source_hash then accept
set policy-options policy-statement pfe_lb_hash term dest_hash from route-filter 140.0.0.0/8
exact
set policy-options policy-statement pfe_lb_hash term dest_hash from route-filter 141.0.0.0/8
exact
set policy-options policy-statement pfe_lb_hash term dest_hash then load-balance destination-ip-
only
set policy-options policy-statement pfe_lb_hash term dest_hash then accept
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then load-balance per-packet
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then accept
set interfaces et-0/0/3 vlan-tagging
set interfaces et-0/0/3 unit 42 vlan-id 42
set interfaces et-0/0/3 unit 42 family inet address 42.1.1.1/30
set protocols iccp local-ip-addr 42.1.1.1
set protocols iccp session-establishment-hold-time 45
set protocols iccp peer 42.1.1.2 redundancy-group-id-list 1
set protocols iccp peer 42.1.1.2 liveness-detection minimum-interval 2500
set services redundancy-set 1 redundancy-group 1
set services redundancy-set 1 redundancy-policy 1_ACQU_MSHIP_POL
set services redundancy-set 1 redundancy-policy 1_RELS_MSHIP_POL
set policy-options redundancy-policy 1_ACQU_MSHIP_POL redundancy-events 1_MSHIP_ACQUIRE_EVENT
set policy-options redundancy-policy 1_ACQU_MSHIP_POL then acquire-mastership
set policy-options redundancy-policy 1_ACQU_MSHIP_POL then add-static-route 3.0.0.3/32 receive
set policy-options redundancy-policy 1_ACQU_MSHIP_POL then add-static-route 3.0.0.3/32 routing-
instance SRD
set policy-options redundancy-policy 1_RELS_MSHIP_POL redundancy-events 1_MSHIP_RELEASE_EVENT
set policy-options redundancy-policy 1_RELS_MSHIP_POL then release-mastership
set policy-options redundancy-policy 1_RELS_MSHIP_POL then delete-static-route 3.0.0.3/32
routing-instance SRD
set event-options redundancy-event 1_MSHIP_ACQUIRE_EVENT monitor peer mastership-release
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae1.0
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae1.1
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae2.0
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae2.1
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae3.0
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae3.1
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor process routing restart
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor peer mastership-acquire
set policy-options condition 1_ROUTE_EXISTS if-route-exists 3.0.0.3/32
set policy-options condition 1_ROUTE_EXISTS if-route-exists table SRD.inet.0
set routing-instances SRD instance-type virtual-router
set routing-instances MNHA-VR instance-type virtual-router
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 type external
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 export mnha-mx-to-srx-export
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 peer-as 5000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 local-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 neighbor 2.1.0.1 local-address
2.1.0.2
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 type external
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 export mnha-mx-to-srx-export
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 peer-as 5000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 local-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 neighbor 2.2.0.1 local-address
2.2.0.2
```

```
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp type internal
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp export mnha-mx-to-mx-export
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp local-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp bfd-liveness-detection
minimum-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp bfd-liveness-detection
minimum-receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp bfd-liveness-detection
multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp neighbor 5.0.0.2
set routing-instances MNHA-VR interface et-0/0/3.5
set routing-instances MNHA-VR interface ae1.100
set routing-instances MNHA-VR interface ae2.100
set interfaces et-0/0/3 unit 5 vlan-id 5
set interfaces et-0/0/3 unit 5 family inet address 5.0.0.1/30
set interfaces ae1 unit 100 vlan-id 100
set interfaces ae1 unit 100 family inet address 2.1.0.2/30
set interfaces ae2 unit 100 vlan-id 100
set interfaces ae2 unit 100 family inet address 2.2.0.2/30
```

Configure MX Series (Standby Node)

```
[edit]
set routing-instances TRUST_VR instance-type virtual-router
set routing-instances TRUST_VR routing-options autonomous-system 1000
set routing-instances TRUST_VR routing-options autonomous-system independent-domain no-attrset
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router type external
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router export
def_route_for_client-2-server
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router peer-as 1500
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group MX-to-TRUST_GW_Router neighbor 41.1.1.1
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx1 neighbor 11.1.1.24
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 type external
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 import pfe_consistent_hash
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 export trust-to-untrust-export
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 peer-as 500
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 local-as 1000
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 multipath
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 bfd-liveness-detection multiplier
3
set routing-instances TRUST_VR protocols bgp group MX-to-vsrx2 neighbor 11.1.1.32
set routing-instances TRUST_VR interface ae1.0
set routing-instances TRUST_VR interface ae2.0
set routing-instances TRUST_VR interface ae11.41
set interfaces et-2/1/1 gigether-options 802.3ad ae1
set interfaces et-2/2/0 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 21
set interfaces ae1 unit 0 family inet address 11.1.1.25/31
set interfaces et-2/1/2 gigether-options 802.3ad ae2
set interfaces et-2/2/1 gigether-options 802.3ad ae2
set interfaces ae2 vlan-tagging
set interfaces ae2 aggregated-ether-options minimum-links 1
set interfaces ae2 aggregated-ether-options lacp active
set interfaces ae2 aggregated-ether-options lacp periodic fast
set interfaces ae2 unit 0 vlan-id 29
set interfaces ae2 unit 0 family inet address 11.1.1.33/31
set interfaces ae2 unit 0 family inet6 address 11:2:2::1/127
set interfaces et-2/0/3 gigether-options 802.3ad ae11
set interfaces et-4/0/1 gigether-options 802.3ad ae11
set interfaces ae11 flexible-vlan-tagging
set interfaces ae11 encapsulation flexible-ethernet-services
set interfaces ae11 aggregated-ether-options minimum-links 1
set interfaces ae11 aggregated-ether-options lacp active
set interfaces ae11 aggregated-ether-options lacp periodic fast
set interfaces ae11 unit 41 vlan-id 41
set interfaces ae11 unit 41 family inet address 41.1.1.2/30
set policy-options policy-statement def_route_for_client-2-server term 1 from protocol bgp
set policy-options policy-statement def_route_for_client-2-server term 1 from route-filter
0.0.0.0/0 exact
set policy-options policy-statement def_route_for_client-2-server term 1 from condition
1_ROUTE_EXISTS
set policy-options policy-statement def_route_for_client-2-server term 1 then next-hop self
set policy-options policy-statement def_route_for_client-2-server term 1 then accept
set policy-options policy-statement def_route_for_client-2-server term 2 from protocol bgp
set policy-options policy-statement def_route_for_client-2-server term 2 from route-filter
0.0.0.0/0 exact
set policy-options policy-statement def_route_for_client-2-server term 2 then as-path-prepend
"1000 1000 1000 1000"
set policy-options policy-statement def_route_for_client-2-server term 2 then next-hop self
set policy-options policy-statement def_route_for_client-2-server term 2 then accept
set policy-options policy-statement def_route_for_client-2-server term 3 then reject
set policy-options policy-statement pfe_consistent_hash from route-filter 0.0.0.0/0 exact
set policy-options policy-statement pfe_consistent_hash then load-balance consistent-hash
set policy-options policy-statement pfe_consistent_hash then accept
set policy-options policy-statement trust-to-untrust-export term 1 from protocol bgp
set policy-options policy-statement trust-to-untrust-export term 1 from protocol static
set policy-options policy-statement trust-to-untrust-export term 1 then next-hop self
set policy-options policy-statement trust-to-untrust-export term 1 then accept
set policy-options policy-statement trust-to-untrust-export term 2 then reject
set forwarding-options enhanced-hash-key symmetric
set routing-options forwarding-table export pfe_lb_hash
set policy-options policy-statement pfe_lb_hash term source_hash from route-filter 0.0.0.0/0
exact
set policy-options policy-statement pfe_lb_hash term source_hash then load-balance source-ip-only
set policy-options policy-statement pfe_lb_hash term source_hash then accept
set policy-options policy-statement pfe_lb_hash term dest_hash from route-filter 140.0.0.0/8
exact
set policy-options policy-statement pfe_lb_hash term dest_hash from route-filter 141.0.0.0/8
exact
set policy-options policy-statement pfe_lb_hash term dest_hash then load-balance destination-ip-
only
```

```
set policy-options policy-statement pfe_lb_hash term dest_hash then accept
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then load-balance per-packet
set policy-options policy-statement pfe_lb_hash term ALL-ELSE then accept
set interfaces et-2/0/2 vlan-tagging
set interfaces et-2/0/2 encapsulation flexible-ethernet-services
set interfaces et-2/0/2 unit 42 vlan-id 42
set interfaces et-2/0/2 unit 42 family inet address 42.1.1.2/30
set protocols iccp local-ip-addr 42.1.1.2
set protocols iccp session-establishment-hold-time 45
set protocols iccp peer 42.1.1.1 redundancy-group-id-list 1
set protocols iccp peer 42.1.1.1 liveness-detection minimum-interval 2500
set services redundancy-set 1 redundancy-group 1
set services redundancy-set 1 redundancy-policy 1_ACQU_MSHIP_POL
set services redundancy-set 1 redundancy-policy 1_RELS_MSHIP_POL
set policy-options redundancy-policy 1_ACQU_MSHIP_POL redundancy-events 1_MSHIP_ACQUIRE_EVENT
set policy-options redundancy-policy 1_ACQU_MSHIP_POL then acquire-mastership
set policy-options redundancy-policy 1_ACQU_MSHIP_POL then add-static-route 3.0.0.3/32 receive
set policy-options redundancy-policy 1_ACQU_MSHIP_POL then add-static-route 3.0.0.3/32 routing-
instance SRD
set policy-options redundancy-policy 1_RELS_MSHIP_POL redundancy-events 1_MSHIP_RELEASE_EVENT
set policy-options redundancy-policy 1_RELS_MSHIP_POL then release-mastership
set policy-options redundancy-policy 1_RELS_MSHIP_POL then delete-static-route 3.0.0.3/32
routing-instance SRD
set event-options redundancy-event 1_MSHIP_ACQUIRE_EVENT monitor peer mastership-release
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae1.0
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae1.1
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae2.0
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae2.1
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae3.0
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor link-down ae3.1
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor process routing restart
set event-options redundancy-event 1_MSHIP_RELEASE_EVENT monitor peer mastership-acquire
set policy-options condition 1_ROUTE_EXISTS if-route-exists 3.0.0.3/32
set policy-options condition 1_ROUTE_EXISTS if-route-exists table SRD.inet.0
set routing-instances SRD instance-type virtual-router
set routing-instances MNHA-VR instance-type virtual-router
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 type external
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 export mnha-mx-to-srx-export
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 peer-as 6000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 local-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 bfd-liveness-detection minimum-
```

```
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_1 neighbor 4.1.0.1 local-address
4.1.0.2
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 type external
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 export mnha-mx-to-srx-export
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 peer-as 6000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 local-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 bfd-liveness-detection minimum-
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ebgp_2 neighbor 4.2.0.1 local-address
4.2.0.2
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp type internal
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp export mnha-mx-to-mx-export
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp local-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp bfd-liveness-detection
minimum-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp bfd-liveness-detection
minimum-receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp bfd-liveness-detection
multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-mx-to-mx-ibgp neighbor 5.0.0.1
set routing-instances MNHA-VR interface et-2/0/2.5
set routing-instances MNHA-VR interface ae1.100
set routing-instances MNHA-VR interface ae2.100
set interfaces et-2/0/2 unit 5 vlan-id 5
set interfaces et-2/0/2 unit 5 family inet address 5.0.0.2/30
set interfaces ae1 unit 100 vlan-id 100
set interfaces ae1 unit 100 family inet address 4.1.0.2/30
set interfaces ae2 unit 100 vlan-id 100
set interfaces ae2 unit 100 family inet address 4.2.0.2/30
```

### Configuration on Gateway Router

```
[edit]
set routing-instances TRUST_VR instance-type virtual-router
set routing-instances TRUST_VR routing-options autonomous-system 1500
set routing-instances TRUST_VR routing-options static route 140.0.0.0/8 next-hop 90.1.1.1
set routing-instances TRUST_VR routing-options static route 141.0.0.0/8 next-hop 90.1.1.1
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust type external
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust export
client_to_server_export
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust peer-as 1000
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust local-as 1500
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX1_trust neighbor 40.1.1.2
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust type external
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust export
client_to_server_export_mx2
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust peer-as 1000
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust local-as 1500
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust bfd-liveness-detection
minimum-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust bfd-liveness-detection
minimum-receive-interval 300
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust bfd-liveness-detection
multiplier 3
set routing-instances TRUST_VR protocols bgp group trust_GW-to-MX2_trust neighbor 41.1.1.2
set routing-instances TRUST_VR protocols bgp multipath
set routing-instances TRUST_VR protocols ospf area 0.0.0.0 interface et-8/1/0.0
set routing-instances TRUST_VR protocols ospf export CLIENT_DEF_export
set routing-instances TRUST_VR interface et-2/1/0.0
set routing-instances TRUST_VR interface ae10.40
set routing-instances TRUST_VR interface ae11.41
set interfaces et-2/1/0 unit 0 family inet address 90.1.1.2/30
set interfaces et-2/1/0 unit 0 family inet6
set interfaces et-7/0/0 gigether-options 802.3ad ae10
set interfaces et-7/0/3 gigether-options 802.3ad ae10
set interfaces ae10 flexible-vlan-tagging
set interfaces ae10 encapsulation flexible-ethernet-services
set interfaces ae10 aggregated-ether-options minimum-links 1
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae10 unit 40 vlan-id 40
set interfaces ae10 unit 40 family inet address 40.1.1.1/30
set interfaces et-7/0/1 gigether-options 802.3ad ae11
set interfaces et-7/0/2 gigether-options 802.3ad ae11
set interfaces ae11 flexible-vlan-tagging
set interfaces ae11 encapsulation flexible-ethernet-services
set interfaces ae11 aggregated-ether-options minimum-links 1
set interfaces ae11 aggregated-ether-options lacp active
set interfaces ae11 aggregated-ether-options lacp periodic fast
set interfaces ae11 unit 41 vlan-id 41
set interfaces ae11 unit 41 family inet address 41.1.1.1/30
set policy-options policy-statement client_to_server_export term 1 from protocol static
set policy-options policy-statement client_to_server_export term 1 from route-filter 141.0.0.0/8
orlonger
set policy-options policy-statement client_to_server_export term 1 from route-filter 140.0.0.0/8
orlonger
set policy-options policy-statement client_to_server_export term 1 then accept
set policy-options policy-statement client_to_server_export term 2 then reject
set policy-options policy-statement client_to_server_export_mx2 term 1 from protocol static
set policy-options policy-statement client_to_server_export_mx2 term 1 from route-filter
141.0.0.0/8 orlonger
set policy-options policy-statement client_to_server_export_mx2 term 1 from route-filter
140.0.0.0/8 orlonger
set policy-options policy-statement client_to_server_export_mx2 term 1 then accept
set policy-options policy-statement client_to_server_export_mx2 term 2 then reject
set routing-instances UNTRUST_VR instance-type virtual-router
set routing-instances UNTRUST_VR routing-options autonomous-system 2500
set routing-instances UNTRUST_VR routing-options static route 0.0.0.0/0 discard
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust type external
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust export
server_to_client_export
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust peer-as 2000
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust local-as 2500
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection minimum-interval 300
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection minimum-receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust bfd-liveness-
detection multiplier 3
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX1_Untrust neighbor 80.1.1.2
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust type external
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust export
server_to_client_export_mx2
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust peer-as 2000
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust local-as 2500
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust bfd-liveness-
detection minimum-interval 300
```

```
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust bfd-liveness-
detection minimum-receive-interval 300
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust bfd-liveness-
detection multiplier 3
set routing-instances UNTRUST_VR protocols bgp group Untrust_GW-to-MX2_Untrust neighbor 81.1.1.2
set routing-instances UNTRUST_VR protocols bgp multipath
set routing-instances UNTRUST_VR interface et-2/1/1.0
set routing-instances UNTRUST_VR interface ae10.80
set routing-instances UNTRUST_VR interface ae11.81
set interfaces et-2/1/1 unit 0 family inet address 100.1.1.2/24
set interfaces ae10 unit 80 vlan-id 80
set interfaces ae10 unit 80 family inet address 80.1.1.1/30
set interfaces ae11 unit 81 vlan-id 81
set interfaces ae11 unit 81 family inet address 81.1.1.1/30
set policy-options policy-statement server_to_client_export term t1 from protocol static
set policy-options policy-statement server_to_client_export term t1 from route-filter 0.0.0.0/0
exact
set policy-options policy-statement server_to_client_export term t1 then accept
set policy-options policy-statement server_to_client_export term t2 then reject
set policy-options policy-statement server_to_client_export_mx2 term t1 from protocol static
set policy-options policy-statement server_to_client_export_mx2 term t1 from route-filter
0.0.0.0/0 exact
set policy-options policy-statement server_to_client_export_mx2 term t1 then accept
set policy-options policy-statement server_to_client_export_mx2 term t2 then reject
```

Configure MNHA Pair 1 (Active Node)

```
[edit]
set routing-instances VR-1 instance-type virtual-router
set routing-instances VR-1 routing-options static route 192.168.64.0/18 discard
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST export trust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.1 peer-as 1000
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST export untrust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.3 peer-as 2000
set routing-instances VR-1 interface ae1.0
set routing-instances VR-1 interface ae1.1
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 1
set interfaces ae1 unit 0 family inet address 10.1.1.0/31
set interfaces ae1 unit 0 family inet6 address 10:1:1::0/127
set interfaces ae1 unit 1 vlan-id 2
set interfaces ae1 unit 1 family inet address 10.1.1.2/31
set interfaces ae1 unit 1 family inet6 address 10:1:1::2/127
set policy-options policy-statement trust_export_policy term 1 from protocol bgp
set policy-options policy-statement trust_export_policy term 1 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement trust_export_policy term 1 then next-hop self
set policy-options policy-statement trust_export_policy term 1 then accept
set policy-options policy-statement trust_export_policy term 2 then reject
set policy-options policy-statement untrust_export_policy term 1 from protocol bgp
set policy-options policy-statement untrust_export_policy term 1 from route-filter 141.0.0.0/8
orlonger
set policy-options policy-statement untrust_export_policy term 1 then accept
set policy-options policy-statement untrust_export_policy term 2 from protocol static
set policy-options policy-statement untrust_export_policy term 2 from route-filter
192.168.64.0/18 orlonger
set policy-options policy-statement untrust_export_policy term 2 then accept
set policy-options policy-statement untrust_export_policy term 3 then reject
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security zones security-zone trust_zone_mnha host-inbound-traffic system-services all
set security zones security-zone trust_zone_mnha host-inbound-traffic protocols all
set security zones security-zone trust_zone_mnha interfaces ae1.100
set security zones security-zone trust_zone_mnha interfaces lo0.0
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match source-address sfw_source_prefix_140.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match source-address sfw_source_prefix_141.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY then
permit
set security address-book global address sfw_source_prefix_140.0.0.0/8 140.0.0.0/8
set security address-book global address sfw_source_prefix_141.0.0.0/8 141.0.0.0/8
set security nat source pool vsrx1_nat_pool address 192.168.64.0/18
set security nat source pool vsrx1_nat_pool address-pooling paired
set security nat source rule-set vsrx1_nat_rule-set from zone vr-1_trust_zone
set security nat source rule-set vsrx1_nat_rule-set to zone vr-1_untrust_zone
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match source-address
140.0.0.0/8
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match destination-
address 0.0.0.0/0
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match application any
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 then source-nat pool
vsrx1_nat_pool
set routing-instances MNHA-VR instance-type virtual-router
set routing-instances MNHA-VR protocols bgp group mnha-ibgp type external
set routing-instances MNHA-VR protocols bgp group mnha-ibgp export mnha_ip
set routing-instances MNHA-VR protocols bgp group mnha-ibgp peer-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp local-as 5000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp multipath
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ibgp neighbor 2.1.0.2 local-address
2.1.0.1
set routing-instances MNHA-VR interface ae1.100
set routing-instances MNHA-VR interface lo0.0
set policy-options policy-statement mnha_ip term 1 from route-filter 2.0.0.1/32 exact
set policy-options policy-statement mnha_ip term 1 then next-hop self
set policy-options policy-statement mnha_ip term 1 then accept
set policy-options policy-statement mnha_ip term 2 then reject
set interfaces ae1 unit 100 vlan-id 100
set interfaces ae1 unit 100 family inet address 2.1.0.1/30
set interfaces lo0 unit 0 family inet address 2.0.0.1/32
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 2.0.0.1
set chassis high-availability peer-id 2 peer-ip 4.0.0.1
set chassis high-availability peer-id 2 interface lo0.0
set chassis high-availability peer-id 2 routing-instance MNHA-VR
set chassis high-availability peer-id 2 liveness-detection minimum-interval 1000
set chassis high-availability peer-id 2 liveness-detection multiplier 3
set chassis high-availability services-redundancy-group 0 peer-id 2
```

Configure MNHA Pair 1 (Backup Node)

```
[edit]
set routing-instances VR-1 instance-type virtual-router
set routing-instances VR-1 routing-options static route 192.168.64.0/18 discard
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST export trust_export_policy_mx2
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST neighbor 11.1.1.25 peer-as 1000
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST export
untrust_export_policy_mx2
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST bfd-liveness-detection
minimum-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST bfd-liveness-detection
```

```
minimum-receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST neighbor 11.1.1.27 peer-as
2000
set routing-instances VR-1 interface ae1.0
set routing-instances VR-1 interface ae1.1
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 21
set interfaces ae1 unit 0 family inet address 11.1.1.24/31
set interfaces ae1 unit 0 family inet6 address 11:1:1::0/127
set interfaces ae1 unit 1 vlan-id 22
set interfaces ae1 unit 1 family inet address 11.1.1.26/31
set interfaces ae1 unit 1 family inet6 address 11:1:1::2/127
set interfaces ae1 unit 2 vlan-id 23
set interfaces ae1 unit 2 family inet address 11.1.1.28/31
set interfaces ae1 unit 3 vlan-id 24
set interfaces ae1 unit 3 family inet address 11.1.1.30/31
set policy-options policy-statement trust_export_policy_mx2 term 1 from route-filter 0.0.0.0/0
exact
set policy-options policy-statement trust_export_policy_mx2 term 1 then next-hop self
set policy-options policy-statement trust_export_policy_mx2 term 1 then accept
set policy-options policy-statement trust_export_policy_mx2 term 2 then reject
set policy-options policy-statement untrust_export_policy_mx2 term 1 from protocol bgp
set policy-options policy-statement untrust_export_policy_mx2 term 1 from route-filter
141.0.0.0/8 orlonger
set policy-options policy-statement untrust_export_policy_mx2 term 2 from protocol static
set policy-options policy-statement untrust_export_policy_mx2 term 2 from route-filter
192.168.64.0/18 orlonger
set policy-options policy-statement untrust_export_policy_mx2 term 2 then accept
set policy-options policy-statement untrust_export_policy_mx2 term 3 then reject
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security zones security-zone trust_zone_mnha host-inbound-traffic system-services all
set security zones security-zone trust_zone_mnha host-inbound-traffic protocols all
set security zones security-zone trust_zone_mnha interfaces ae1.100
set security zones security-zone trust_zone_mnha interfaces lo0.0
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match source-address sfw_source_prefix_140.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match source-address sfw_source_prefix_141.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY then
permit
set security address-book global address sfw_source_prefix_140.0.0.0/8 140.0.0.0/8
set security address-book global address sfw_source_prefix_141.0.0.0/8 141.0.0.0/8
set security nat source pool vsrx1_nat_pool address 192.168.64.0/18
set security nat source pool vsrx1_nat_pool address-pooling paired
set security nat source rule-set vsrx1_nat_rule-set from zone vr-1_trust_zone
set security nat source rule-set vsrx1_nat_rule-set to zone vr-1_untrust_zone
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match source-address
140.0.0.0/8
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match destination-
address 0.0.0.0/0
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match application any
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 then source-nat pool
vsrx1_nat_pool
set routing-instances MNHA-VR instance-type virtual-router
set routing-instances MNHA-VR protocols bgp group mnha-ibgp type external
set routing-instances MNHA-VR protocols bgp group mnha-ibgp export mnha_ip
set routing-instances MNHA-VR protocols bgp group mnha-ibgp peer-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp local-as 6000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp multipath
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ibgp neighbor 4.1.0.2 local-address
4.1.0.1
set routing-instances MNHA-VR interface ae1.100
set routing-instances MNHA-VR interface lo0.0
set policy-options policy-statement mnha_ip term 1 from route-filter 4.0.0.1/32 exact
set policy-options policy-statement mnha_ip term 1 then next-hop self
set policy-options policy-statement mnha_ip term 1 then accept
set policy-options policy-statement mnha_ip term 2 then reject
set interfaces ae1 unit 100 vlan-id 100
set interfaces ae1 unit 100 family inet address 4.1.0.1/30
set interfaces lo0 unit 0 family inet address 4.0.0.1/32
set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 4.0.0.1
set chassis high-availability peer-id 1 peer-ip 2.0.0.1
set chassis high-availability peer-id 1 interface lo0.0
set chassis high-availability peer-id 1 routing-instance MNHA-VR
set chassis high-availability peer-id 1 liveness-detection minimum-interval 1000
set chassis high-availability peer-id 1 liveness-detection multiplier 3
set chassis high-availability services-redundancy-group 0 peer-id 1
```

Configure MNHA Pair 2 (Active Node)

```
[edit]
set routing-instances VR-1 instance-type virtual-router
set routing-instances VR-1 routing-options static route 192.169.64.0/18 discard
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST export trust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_TRUST neighbor 10.1.1.9 peer-as 1000
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST export untrust_export_policy
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection minimum-
receive-interval 300
```

```
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX_UNTRUST neighbor 10.1.1.11 peer-as 2000
set routing-instances VR-1 interface ae1.0
set routing-instances VR-1 interface ae1.1
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 9
set interfaces ae1 unit 0 family inet address 10.1.1.8/31
set interfaces ae1 unit 1 vlan-id 10
set interfaces ae1 unit 1 family inet address 10.1.1.10/31
set policy-options policy-statement trust_export_policy term 1 from protocol bgp
set policy-options policy-statement trust_export_policy term 1 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement trust_export_policy term 1 then next-hop self
set policy-options policy-statement trust_export_policy term 1 then accept
set policy-options policy-statement trust_export_policy term 2 then reject
set policy-options policy-statement untrust_export_policy term 1 from protocol bgp
set policy-options policy-statement untrust_export_policy term 1 from route-filter 141.0.0.0/8
orlonger
set policy-options policy-statement untrust_export_policy term 1 then accept
set policy-options policy-statement untrust_export_policy term 2 from protocol static
set policy-options policy-statement untrust_export_policy term 2 from route-filter
192.169.64.0/18 orlonger
set policy-options policy-statement untrust_export_policy term 2 then accept
set policy-options policy-statement untrust_export_policy term 3 then reject
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security zones security-zone trust_zone_mnha host-inbound-traffic system-services all
set security zones security-zone trust_zone_mnha host-inbound-traffic protocols all
set security zones security-zone trust_zone_mnha interfaces ae1.100
set security zones security-zone trust_zone_mnha interfaces lo0.0
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match source-address sfw_source_prefix_140.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match destination-address any
```

```
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match source-address sfw_source_prefix_141.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY then
permit
set security address-book global address sfw_source_prefix_140.0.0.0/8 140.0.0.0/8
set security address-book global address sfw_source_prefix_141.0.0.0/8 141.0.0.0/8
set security nat source pool vsrx1_nat_pool address 192.169.64.0/18
set security nat source pool vsrx1_nat_pool address-pooling paired
set security nat source rule-set vsrx1_nat_rule-set from zone vr-1_trust_zone
set security nat source rule-set vsrx1_nat_rule-set to zone vr-1_untrust_zone
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match source-address
140.0.0.0/8
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match destination-
address 0.0.0.0/0
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match application any
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 then source-nat pool
vsrx1_nat_pool
set routing-instances MNHA-VR instance-type virtual-router
set routing-instances MNHA-VR protocols bgp group mnha-ibgp type external
set routing-instances MNHA-VR protocols bgp group mnha-ibgp export mnha_ip
set routing-instances MNHA-VR protocols bgp group mnha-ibgp peer-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp local-as 5000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp multipath
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ibgp neighbor 2.2.0.2 local-address
2.2.0.1
set routing-instances MNHA-VR interface ae1.100
set routing-instances MNHA-VR interface lo0.0
set policy-options policy-statement mnha_ip term 1 from route-filter 2.0.0.2/32 exact
set policy-options policy-statement mnha_ip term 1 then next-hop self
set policy-options policy-statement mnha_ip term 1 then accept
set policy-options policy-statement mnha_ip term 2 then reject
set interfaces ae1 unit 100 vlan-id 100
set interfaces ae1 unit 100 family inet address 2.2.0.1/30
set interfaces lo0 unit 0 family inet address 2.0.0.2/32
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 2.0.0.2
set chassis high-availability peer-id 2 peer-ip 4.0.0.2
set chassis high-availability peer-id 2 interface lo0.0
set chassis high-availability peer-id 2 routing-instance MNHA-VR
set chassis high-availability peer-id 2 liveness-detection minimum-interval 1000
set chassis high-availability peer-id 2 liveness-detection multiplier 3
set chassis high-availability services-redundancy-group 0 peer-id 2
```

Configure MNHA Pair 2 (Backup Node)

```
[edit]
set routing-instances VR-1 instance-type virtual-router
set routing-instances VR-1 routing-options static route 192.169.64.0/18 discard
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST export trust_export_policy_mx2
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST bfd-liveness-detection minimum-
interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST bfd-liveness-detection minimum-
receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_TRUST neighbor 11.1.1.33 peer-as 1000
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST type external
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST export
untrust_export_policy_mx2
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST local-as 500
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST bfd-liveness-detection
minimum-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST bfd-liveness-detection
minimum-receive-interval 300
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST bfd-liveness-detection
multiplier 3
set routing-instances VR-1 protocols bgp group Vsrx-to-MX2_UNTRUST neighbor 11.1.1.35 peer-as
2000
set routing-instances VR-1 interface et-1/0/3.0
set routing-instances VR-1 interface ae1.0
set routing-instances VR-1 interface ae1.1
set interfaces et-1/0/0 gigether-options 802.3ad ae1
set interfaces et-1/0/1 gigether-options 802.3ad ae1
set interfaces ae1 vlan-tagging
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 unit 0 vlan-id 29
set interfaces ae1 unit 0 family inet address 11.1.1.32/31
set interfaces ae1 unit 0 family inet6 address 11:2:2::0/127
set interfaces ae1 unit 1 vlan-id 30
set interfaces ae1 unit 1 family inet address 11.1.1.34/31
set interfaces ae1 unit 1 family inet6 address 11:2:2::2/127
set policy-options policy-statement trust_export_policy_mx2 term 1 from protocol bgp
set policy-options policy-statement trust_export_policy_mx2 term 1 from route-filter 0.0.0.0/0
exact
set policy-options policy-statement trust_export_policy_mx2 term 1 then next-hop self
set policy-options policy-statement trust_export_policy_mx2 term 1 then accept
set policy-options policy-statement trust_export_policy_mx2 term 2 then reject
set policy-options policy-statement untrust_export_policy_mx2 term 1 from protocol bgp
set policy-options policy-statement untrust_export_policy_mx2 term 1 from route-filter
141.0.0.0/8 orlonger
set policy-options policy-statement untrust_export_policy_mx2 term 1 then accept
set policy-options policy-statement untrust_export_policy_mx2 term 2 from protocol static
set policy-options policy-statement untrust_export_policy_mx2 term 2 from route-filter
192.169.64.0/18 orlonger
set policy-options policy-statement untrust_export_policy_mx2 term 2 then accept
set policy-options policy-statement untrust_export_policy_mx2 term 3 then reject
set security zones security-zone vr-1_trust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_trust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_trust_zone interfaces ae1.0
set security zones security-zone vr-1_untrust_zone host-inbound-traffic system-services all
set security zones security-zone vr-1_untrust_zone host-inbound-traffic protocols all
set security zones security-zone vr-1_untrust_zone interfaces ae1.1
set security zones security-zone trust_zone_mnha host-inbound-traffic system-services all
set security zones security-zone trust_zone_mnha host-inbound-traffic protocols all
set security zones security-zone trust_zone_mnha interfaces ae1.100
set security zones security-zone trust_zone_mnha interfaces lo0.0
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match source-address sfw_source_prefix_140.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
```

```
SNAT_NAPT44_POLICY match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy
SNAT_NAPT44_POLICY then permit
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match source-address sfw_source_prefix_141.0.0.0/8
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match destination-address any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY
match application any
set security policies from-zone vr-1_trust_zone to-zone vr-1_untrust_zone policy SFW_POLICY then
permit
set security address-book global address sfw_source_prefix_140.0.0.0/8 140.0.0.0/8
set security address-book global address sfw_source_prefix_141.0.0.0/8 141.0.0.0/8
set security nat source pool vsrx1_nat_pool address 192.169.64.0/18
set security nat source pool vsrx1_nat_pool address-pooling paired
set security nat source rule-set vsrx1_nat_rule-set from zone vr-1_trust_zone
set security nat source rule-set vsrx1_nat_rule-set to zone vr-1_untrust_zone
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match source-address
140.0.0.0/8
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match destination-
address 0.0.0.0/0
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 match application any
set security nat source rule-set vsrx1_nat_rule-set rule vsrx1_nat_rule1 then source-nat pool
vsrx1_nat_pool
set routing-instances MNHA-VR instance-type virtual-router
set routing-instances MNHA-VR protocols bgp group mnha-ibgp type external
set routing-instances MNHA-VR protocols bgp group mnha-ibgp export mnha_ip
set routing-instances MNHA-VR protocols bgp group mnha-ibgp peer-as 4000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp local-as 6000
set routing-instances MNHA-VR protocols bgp group mnha-ibgp multipath
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection minimum-
receive-interval 300
set routing-instances MNHA-VR protocols bgp group mnha-ibgp bfd-liveness-detection multiplier 3
set routing-instances MNHA-VR protocols bgp group mnha-ibgp neighbor 4.2.0.2 local-address
4.2.0.1
set routing-instances MNHA-VR interface ae1.100
set routing-instances MNHA-VR interface lo0.0
set policy-options policy-statement mnha_ip term 1 from route-filter 4.0.0.2/32 exact
set policy-options policy-statement mnha_ip term 1 then next-hop self
set policy-options policy-statement mnha_ip term 1 then accept
set policy-options policy-statement mnha_ip term 2 then reject
set interfaces ae1 unit 100 vlan-id 100
set interfaces ae1 unit 100 family inet address 4.2.0.1/30
set interfaces lo0 unit 0 family inet address 4.0.0.2/32
```

## Verification

The following items highlight a list of show commands used to verify the feature in this example.

- Verify MX Series (active node) configuration
- Verify MX Series (standby node) configuration
- Verify the Gateway router configuration
- Verify MNHA Pair 1 (active node) configuration
- Verify MNHA Pair 1 (backup node) configuration
- Verify MNHA Pair 2 (active node) configuration
- Verify MNHA Pair 2 (backup node) configuration

### Verify MX Series (Active Node) Configuration

```
user@MX1# run show services redundancy-group
ICCP process connection               : Connected
              Redundancy Group ID: 1
               Number of peer RG connections  : 1
               Local RG IP                    : 42.1.1.1
               RS ID     Local RS state     Peer RS state     Peer RG IP        Status
               1         MASTER             STANDBY           42.1.1.2
Connecteduser@MX1# run show bgp summary instance TRUST_VR
              Threading mode: BGP I/O
              Groups: 8 Peers: 8 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              TRUST_VR.inet.0
                                    5          5          0          0          0          0
              TRUST_VR.mdt.0
                                    0          0          0          0          0          0
              Peer                       AS       InPkt      OutPkt     OutQ    Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              10.1.1.0                 500        1191       1159        0        0     9:00:31
```

```
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.8                500       1188       1159       0       0     9:00:03
Establ
               TRUST_VR.inet.0: 1/1/1/0
              10.1.1.16               500       1357       1323       0       0    10:16:00
Establ
               TRUST_VR.inet.0: 1/1/1/0
              40.1.1.1               1500        402        396       0       1     3:03:12
Establ
               TRUST_VR.inet.0: 2/2/2/0user@MX1# run show bgp summary instance UNTRUST_VR
              Threading mode: BGP I/O
              Groups: 8 Peers: 8 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              UNTRUST_VR.inet.0
                                    10         10          0          0          0          0
              UNTRUST_VR.mdt.0
                                     0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              10.1.1.2                500       1194       1161       0       0     9:01:03
Establ
               UNTRUST_VR.inet.0: 3/3/3/0
              10.1.1.10               500       1191       1159       0       0     9:00:28
Establ
               UNTRUST_VR.inet.0: 3/3/3/0
              10.1.1.18               500       1359       1325       0       0    10:16:30
Establ
               UNTRUST_VR.inet.0: 3/3/3/0
              80.1.1.1               2500        402        401       0       1     3:03:41
Establ
               UNTRUST_VR.inet.0: 1/1/1/0
```

```
user@MX1# run show bgp summary instance MNHA-VR
                            Threading mode: BGP I/O
              Groups: 4 Peers: 4 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              MNHA-VR.inet.0
                                     6          6          0          0          0          0
              MNHA-VR.mdt.0
                                     0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              2.1.0.1                5000       1193       1161       0       0     9:01:43
Establ
               MNHA-VR.inet.0: 1/1/1/0
              2.2.0.1                5000       1189       1160       0       0     9:01:12
Establ
               MNHA-VR.inet.0: 1/1/1/0
              2.3.0.1                5000       1358       1327       0       0    10:17:22
Establ
               MNHA-VR.inet.0: 1/1/1/0
              5.0.0.2                4000       1330       1328       0       0    10:17:06
Establ
               MNHA-VR.inet.0: 3/3/3/0
```

```
user@MX1# run show bfd session
                                         Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              2.1.0.1                  Up        ae1.100        0.900     0.300        3
              2.2.0.1                  Up        ae2.100        0.900     0.300        3
              2.3.0.1                  Up        ae3.100        0.900     0.300        3
              5.0.0.2                  Up        et-0/0/3.5     0.900     0.300        3
              10.1.1.0                 Up        ae1.0          0.900     0.300        3
              10.1.1.2                 Up        ae1.1          0.900     0.300        3
              10.1.1.8                 Up        ae2.0          0.900     0.300        3
              10.1.1.10                Up        ae2.1          0.900     0.300        3
              10.1.1.16                Up        ae3.0          0.900     0.300        3
              10.1.1.18                Up        ae3.1          0.900     0.300        3
              40.1.1.1                 Up        ae10.40        0.900     0.300        3
              42.1.1.2                 Up                       7.500     2.500        3
              80.1.1.1                 Up        ae10.80        0.900     0.300        3
              13 sessions, 13 clients
              Cumulative transmit rate 67.1 pps, cumulative receive rate 67.1 pps
```

```
user@MX1# run show route table TRUST_VR.inet 100.1.1.4 active-path
TRUST_VR.inet.0: 13 destinations, 15 routes (13 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 04:00:35, localpref 100
                                    AS path: 500 2000 2500 I, validation-state: unverified
```

```
                                  >  to 10.1.1.0 via ae1.0
                                     to 10.1.1.8 via ae2.0
```

```
user@MX1# run show route table TRUST_VR.inet 100.1.1.4 extensive active-path
TRUST_VR.inet.0: 13 destinations, 15 routes (13 active, 0 holddown, 0 hidden)
              0.0.0.0/0 (3 entries, 1 announced)
              TSI:
              KRT in-kernel 0.0.0.0/0 -> {list:10.1.1.0, 10.1.1.8, 10.1.1.16 Flags source ip
load-balance}
              Page 0 idx 1, (group MX-to-TRUST_GW_Router type External) Type 1 val 0x12b05d48
(adv_entry)
                Advertised metrics:
                  Flags: Nexthop Change
                  Nexthop: Self
                  AS path: [1000] 500 2000 2500 I
                  Communities:
                 Advertise: 00000001
              Path 0.0.0.0
              from 10.1.1.0
              Vector len 4.  Val: 1
              Associated with 1 conditions: 1_ROUTE_EXISTS (static)
                     *BGP    Preference: 170/-101
                             Next hop type: Router, Next hop index: 0
                             Address: 0xf918fec
                             Next-hop reference count: 2, Next-hop session id: 0
                             Kernel Table Id: 0
                             Source: 10.1.1.0
                             Next hop: 10.1.1.0 via ae1.0, selected
                             Session Id: 0
                             Next hop: 10.1.1.8 via ae2.0
                             Session Id: 0
                             State: <Active Ext LoadBalConsistentHash>
                             Local AS:  1000 Peer AS:   500
                             Age: 4:00:00
                             Validation State: unverified
                             Task: BGP_500_1000.10.1.1.0
                             Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
                             AS path: 500 2000 2500 I
                             Accepted Multipath
                             Localpref: 100
```

```
                               Router ID: 10.1.1.0
                               Thread: junos-main
```

```
user@MX1# run show route table UNTRUST_VR.inet 141/8 active-path
                              Mar 14 21:02:22
              UNTRUST_VR.inet.0: 16 destinations, 20 routes (16 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 04:01:42, localpref 100, from 10.1.1.2
                                   AS path: 500 1000 1500 I, validation-state: unverified
                                    to 10.1.1.2 via ae1.1
                                    to 10.1.1.10 via ae2.1
```

```
user@MX1# run show route table UNTRUST_VR.inet 141/8 active-path extensive
                              Mar 14 21:02:25
              UNTRUST_VR.inet.0: 16 destinations, 20 routes (16 active, 0 holddown, 0 hidden)
              141.0.0.0/8 (3 entries, 1 announced)
              TSI:
              KRT in-kernel 141.0.0.0/8 -> {list:10.1.1.2, 10.1.1.10, 10.1.1.18 Flags
destination ip load-balance}
              Page 0 idx 1, (group MX-to-UNTRUST_GW_Router type External) Type 1 val 0x12b073d0
(adv_entry)
                Advertised metrics:
                  Flags: Nexthop Change
                  Nexthop: Self
                  AS path: [2000] 500 1000 1500 I
                  Communities:
                 Advertise: 00000001
              Path 141.0.0.0
              from 10.1.1.2
              Vector len 4.  Val: 1
              Associated with 1 conditions: 1_ROUTE_EXISTS (static)
                     *BGP    Preference: 170/-101
                             Next hop type: Router, Next hop index: 0
                             Address: 0xf918b24
                             Next-hop reference count: 3, Next-hop session id: 0
                             Kernel Table Id: 0
                             Source: 10.1.1.2
                             Next hop: 10.1.1.2 via ae1.1
                             Session Id: 0
                             Next hop: 10.1.1.10 via ae2.1
```

```
                              Session Id: 0
                              State: <Active Ext LoadBalConsistentHash>
                              Local AS:  2000 Peer AS:   500
                              Age: 4:01:44
                              Validation State: unverified
                              Task: BGP_500_2000.10.1.1.2
                              Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
                              AS path: 500 1000 1500 I
                              Accepted Multipath
                              Localpref: 100
                              Router ID: 10.1.1.0
                              Thread: junos-main
```

```
user@MX1# run show route table UNTRUST_VR.inet 192/8
                             UNTRUST_VR.inet.0: 16 destinations, 20 routes (16 active, 0 
holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              192.168.64.0/18    *[BGP/170] 09:59:58, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.2 via ae1.1
              192.169.64.0/18    *[BGP/170] 09:59:24, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 10.1.1.10 via ae2.1
```

## Verify MX Series (Standby Node) Configuration

```
user@MX1# run show services redundancy-group
                             ICCP process connection                : Connected
              Redundancy Group ID: 1
               Number of peer RG connections  : 1
               Local RG IP                    : 42.1.1.2
               RS ID     Local RS state     Peer RS state     Peer RG IP        Status
               1         STANDBY            MASTER            42.1.1.1          Connected
```

```
user@MX1# run show bgp summary instance TRUST_VR
                             Threading mode: BGP I/O
              Groups: 8 Peers: 8 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              TRUST_VR.inet.0
```

```
                      5          5          0          0          0          0
              TRUST_VR.inet6.0
                      3          3          0          0          0          0
              TRUST_VR.mdt.0
                      0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              11.1.1.24               500       1317       1283       0       0     9:57:45
Establ
               TRUST_VR.inet.0: 1/1/1/0
              11.1.1.32               500       1315       1284       0       0     9:58:11
Establ
               TRUST_VR.inet.0: 1/1/1/0
              11.1.1.40               500       1315       1284       0       0     9:58:15
Establ
               TRUST_VR.inet.0: 1/1/1/0
              41.1.1.1               1500        535        526       0    4767     4:04:16
Establ
               TRUST_VR.inet.0: 2/2/2/0
```

```
user@MX1# run show bgp summary instance UNTRUST_VR
                            Threading mode: BGP I/O
              Groups: 8 Peers: 8 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              UNTRUST_VR.inet.0
                     10         10          0          0          0          0
              UNTRUST_VR.mdt.0
                      0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              11.1.1.26               500       1318       1282       0       0     9:57:45
Establ
               UNTRUST_VR.inet.0: 3/3/3/0
              11.1.1.34               500       1317       1283       0       0     9:58:11
Establ
               UNTRUST_VR.inet.0: 3/3/3/0
              11.1.1.42               500       1318       1283       0       0     9:58:12
Establ
               UNTRUST_VR.inet.0: 3/3/3/0
              81.1.1.1               2500        535        531       0    5211     4:04:24
```

```
Establ
               UNTRUST_VR.inet.0: 1/1/1/0
```

```
user@MX1# run show bgp summary instance MNHA-VR
                            Threading mode: BGP I/O
              Groups: 4 Peers: 4 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              MNHA-VR.inet.0
                                    6          6          0          0          0          0
              MNHA-VR.mdt.0
                                    0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              4.1.0.1                6000       1316       1282       0       0     9:57:53
Establ
               MNHA-VR.inet.0: 1/1/1/0
              4.2.0.1                6000       1315       1282       0       0     9:58:18
Establ
               MNHA-VR.inet.0: 1/1/1/0
              4.3.0.1                6000       1315       1283       0       0     9:58:27
Establ
               MNHA-VR.inet.0: 1/1/1/0
              5.0.0.1                4000       1457       1457       0       0    11:17:04
Establ
               MNHA-VR.inet.0: 3/3/3/0
```

```
user@MX1# run show bfd session
                                                                             Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              4.1.0.1                  Up        ae1.100        0.900     0.300        3
              4.2.0.1                  Up        ae2.100        0.900     0.300        3
              4.3.0.1                  Up        ae3.100        0.900     0.300        3
              5.0.0.1                  Up        et-2/0/2.5     0.900     0.300        3
              11.1.1.24                Up        ae1.0          0.900     0.300        3
              11.1.1.26                Up        ae1.1          0.900     0.300        3
              11.1.1.32                Up        ae2.0          0.900     0.300        3
              11.1.1.34                Up        ae2.1          0.900     0.300        3
              11.1.1.40                Up        ae3.0          0.900     0.300        3
              11.1.1.42                Up        ae3.1          0.900     0.300        3
              41.1.1.1                 Up        ae11.41        0.900     0.300        3
              42.1.1.1                 Up                      7.500     2.500        3
              81.1.1.1                 Up         ae11.81      0.900     0.300        3
```

```
user@MX1# run show route table TRUST_VR.inet 100.1.1.4 active-path
                            TRUST_VR.inet.0: 13 destinations, 15 routes (13 active, 0 holddown,
0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 04:06:50, localpref 100
                                   AS path: 500 2000 2500 I, validation-state: unverified
                                 >  to 11.1.1.24 via ae1.0
                                    to 11.1.1.32 via ae2.0
```

```
user@MX1# run show route table TRUST_VR.inet 100.1.1.4 active-path extensive
                            TRUST_VR.inet.0: 13 destinations, 15 routes (13 active, 0 holddown,
0 hidden)
              0.0.0.0/0 (3 entries, 1 announced)
              TSI:
              Page 0 idx 0, (group MX-to-TRUST_GW_Router type External) Type 1 val 0x10f68c00
(adv_entry)
                Advertised metrics:
                  Flags: Nexthop Change
                  Nexthop: Self
                  AS path: 1000 1000 1000 1000 [1000] 500 2000 2500 I
                  Communities:
                 Advertise: 00000001
              Path 0.0.0.0
              from 11.1.1.24
              Vector len 4.  Val: 0
              KRT in-kernel 0.0.0.0/0 -> {list:11.1.1.24, 11.1.1.32 Flags source ip load-balance}
              Associated with 1 conditions: 1_ROUTE_EXISTS (static)
                     *BGP    Preference: 170/-101
                             Next hop type: Router, Next hop index: 0
                             Address: 0x1724a194
                             Next-hop reference count: 2, Next-hop session id: 0
                             Kernel Table Id: 0
                             Source: 11.1.1.24
                             Next hop: 11.1.1.24 via ae1.0, selected
                             Session Id: 0
                             Next hop: 11.1.1.32 via ae2.0
                             Session Id: 0
```

```
                               State: <Active Ext LoadBalConsistentHash>
                               Local AS:  1000 Peer AS:   500
                               Age: 4:06:51
                               Validation State: unverified
                               Task: BGP_500_1000.11.1.1.24
                               Announcement bits (3): 0-BGP_RT_Background 1-KRT 2-BGP_Multi_Path
                               AS path: 500 2000 2500 I
                               Accepted Multipath
                               Localpref: 100
                               Router ID: 11.1.1.24
                               Thread: junos-main
```

```
user@MX1# run show route table UNTRUST_VR.inet 141/8 active-path
                          UNTRUST_VR.inet.0: 16 destinations, 20 routes (16 active, 0 holddown,
0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 04:07:32, localpref 100, from 11.1.1.26
                                   AS path: 500 1000 1500 I, validation-state: unverified
                                    to 11.1.1.26 via ae1.1
                                    to 11.1.1.34 via ae2.1
```

```
user@MX1# run show route table UNTRUST_VR.inet 141/8 active-path extensive
                            UNTRUST_VR.inet.0: 16 destinations, 20 routes (16 active, 0
holddown, 0 hidden)
              141.0.0.0/8 (3 entries, 1 announced)
              TSI:
              KRT in-kernel 141.0.0.0/8 -> {list:11.1.1.26, 11.1.1.34 Flags destination ip load-
balance}
              Page 0 idx 0, (group MX-to-UNTRUST_GW_Router type External) Type 1 val 0x10f6b2b8
(adv_entry)
                Advertised metrics:
                  Flags: Nexthop Change
                  Nexthop: Self
                  AS path: 2000 2000 2000 2000 [2000] 500 1000 1500 I
                  Communities:
                 Advertise: 00000001
              Path 141.0.0.0
              from 11.1.1.26
              Vector len 4.  Val: 0
              Associated with 1 conditions: 1_ROUTE_EXISTS (static)
```

```
                       *BGP    Preference: 170/-101
                               Next hop type: Router, Next hop index: 0
                               Address: 0x10c0e194
                               Next-hop reference count: 3, Next-hop session id: 0
                               Kernel Table Id: 0
                               Source: 11.1.1.26
                               Next hop: 11.1.1.26 via ae1.1
                               Session Id: 0
                               Next hop: 11.1.1.34 via ae2.1
                               Session Id: 0
                               State: <Active Ext LoadBalConsistentHash>
                               Local AS:  2000 Peer AS:   500
                               Age: 4:07:36
                               Validation State: unverified
                               Task: BGP_500_2000.11.1.1.26
                               Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
                               AS path: 500 1000 1500 I
                               Accepted Multipath
                               Localpref: 100
                               Router ID: 11.1.1.24
                               Thread: junos-main
```

```
user@MX1# run show route table UNTRUST_VR.inet 192/8
                            UNTRUST_VR.inet.0: 16 destinations, 20 routes (16 active, 0
holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              192.168.64.0/18    *[BGP/170] 10:01:32, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 11.1.1.26 via ae1.1
              192.169.64.0/18    *[BGP/170] 10:01:57, localpref 100
                                   AS path: 500 I, validation-state: unverified
                                 >  to 11.1.1.34 via ae2.1
```

## Verify Gateway Router Configuration

```
user@gw# run show route table TRUST_VR.inet 100.1.1.4
                            TRUST_VR.inet.0: 20 destinations, 21 routes (20 active, 0 holddown,
0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 01:09:44, localpref 100
                                   AS path: 1000 500 2000 2500 I, validation-state: unverified
```

```
                                  >  to 40.1.1.2 via ae10.40
                                  [BGP/170] 01:09:44, localpref 100
                                    AS path: 1000 1000 1000 1000 1000 500 2000 2500 I, validation-
state: unverified
                                  >  to 41.1.1.2 via ae11.41
```

```
user@gw# run show route table UNTRUST_VR.inet 141/8
                             UNTRUST_VR.inet.0: 23 destinations, 28 routes (23 active, 0
holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 01:10:00, localpref 100
                                    AS path: 2000 500 1000 1500 I, validation-state: unverified
                                  >  to 80.1.1.2 via ae10.80
                                  [BGP/170] 01:10:00, localpref 100
                                    AS path: 2000 2000 2000 2000 2000 500 1000 1500 I, validation-
state: unverified
                                  >  to 81.1.1.2 via ae11.81
```

```
user@gw# run show route table UNTRUST_VR.inet 192/8
                             UNTRUST_VR.inet.0: 23 destinations, 28 routes (23 active, 0
holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              192.168.64.0/18    *[BGP/170] 01:10:07, localpref 100
                                    AS path: 2000 500 I, validation-state: unverified
                                  >  to 80.1.1.2 via ae10.80
                                  [BGP/170] 01:10:07, localpref 100
                                    AS path: 2000 2000 2000 2000 2000 500 I, validation-state:
unverified
                                  >  to 81.1.1.2 via ae11.81
              192.169.64.0/18    *[BGP/170] 01:10:07, localpref 100
                                    AS path: 2000 500 I, validation-state: unverified
                                  >  to 80.1.1.2 via ae10.80
                                  [BGP/170] 01:10:07, localpref 100
                                    AS path: 2000 2000 2000 2000 2000 500 I, validation-state:
unverified
                                  >  to 81.1.1.2 via ae11.81
              run show services redundancy-group
```

### Verify MNHA Pair 1 (Active Node) Configuration

```
user@srxa1# run show chassis high-availability information
                            Node failure codes:
                 HW  Hardware monitoring    LB  Loopback monitoring
                 MB  Mbuf monitoring        SP  SPU monitoring
                 CS  Cold Sync monitoring   SU  Software Upgrade
              Node Status: ONLINE
              Local-id: 1
              Local-IP: 2.0.0.1
              HA Peer Information:
                 Peer Id: 2        IP address: 4.0.0.1       Interface: lo0.0
                 Routing Instance: MNHA-VR
                 Encrypted: NO     Conn State: UP
                 Configured BFD Detection Time: 3 * 1000ms
                 Cold Sync Status: COMPLETE
              Services Redundancy Group: 0
                     Current State: ONLINE
                Peer Information:
                       Peer Id: 2
```

```
user@srxa1# run show bgp summary
                            Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 5 Peers: 5 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              inet.0
                                     0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              2.1.0.2                4000       1325       1358       0       0    10:17:59
Establ
               MNHA-VR.inet.0: 3/3/3/0
              10.1.1.1               1000       1326       1359       0       0    10:18:04
Establ
               VR-1.inet.0: 2/2/2/0
              10.1.1.3               2000       1327       1361       0       0    10:18:07
```

```
Establ
               VR-1.inet.0: 1/1/1/0
```

```
user@srxa1# run show bfd session
                                                                              Detect   Transmit
              Address                  State     Interface       Time     Interval  Multiplier
              2.1.0.2                  Up        ae1.100         0.900     0.300        3
              4.0.0.1                  Up                        3.000     1.000        3
              10.1.1.1                 Up        ae1.0           0.900     0.300        3
              10.1.1.3                 Up        ae1.1           0.900     0.300        3
```

```
user@srxa1# run show route 100.1.1.4
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 04:21:10, localpref 100
                                   AS path: 2000 2500 I, validation-state: unverified
                                 >  to 10.1.1.3 via ae1.1
```

```
user@srxa1# run show route 140/8
                            Mar 14 21:21:55
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              140.0.0.0/8        *[BGP/170] 04:21:15, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.1 via ae1.0
```

```
user@srxa1# run show route 141/8
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 04:21:17, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.1 via ae1.0
```

```
user@srxa1# run show security flow session source-prefix 140.6.219.193 source-port 22279
               Session ID: 51539643469, Policy name: SNAT_NAPT44_POLICY/6, HA State: Active,
Timeout: 1794, Session State: Valid
```

```
               In: 140.6.219.193/22279 --> 100.1.1.77/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts: 
16428, Bytes: 17507248, HA Wing State: Active,
               Out: 100.1.1.77/70 --> 192.168.65.156/2480;tcp, Conn Tag: 0x0, If: ae1.1, Pkts: 
14829, Bytes: 17366534, HA Wing State: Active,
              Total sessions: 1
```

```
user@srxa1# run show security flow session source-prefix 141.5.93.243 source-port 11462
               Session ID: 51539624377, Policy name: SFW_POLICY/4, HA State: Active, Timeout: 
1800, Session State: Valid
               In: 141.5.93.243/11462 --> 100.1.1.238/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts: 
17929, Bytes: 19082142, HA Wing State: Active,
               Out: 100.1.1.238/70 --> 141.5.93.243/11462;tcp, Conn Tag: 0x0, If: ae1.1, Pkts: 
15104, Bytes: 18908914, HA Wing State: Active,
              Total sessions: 1
```

## Verify MNHA Pair 1 (Backup Node) Configuration

```
user@srxa2# run show chassis high-availability information
                 Node failure codes:
                 HW  Hardware monitoring    LB  Loopback monitoring
                 MB  Mbuf monitoring        SP  SPU monitoring
                 CS  Cold Sync monitoring   SU  Software Upgrade
              Node Status: ONLINE
              Local-id: 2
              Local-IP: 4.0.0.1
              HA Peer Information:
                 Peer Id: 1        IP address: 2.0.0.1       Interface: lo0.0
                 Routing Instance: MNHA-VR
                 Encrypted: NO     Conn State: UP
                 Configured BFD Detection Time: 3 * 1000ms
                 Cold Sync Status: COMPLETE
              Services Redundancy Group: 0
                     Current State: ONLINE
                Peer Information:
                       Peer Id: 1
```

```
user@srxa2# run show bgp summary
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 5 Peers: 5 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              inet.0
                                   0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              4.1.0.2                4000       1333       1365       0       0    10:20:56
Establ
               MNHA-VR.inet.0: 3/3/3/0
              11.1.1.25              1000       1334       1367       0       0    10:20:56
Establ
               VR-1.inet.0: 2/2/2/0
              11.1.1.27              2000       1333       1367       0       0    10:20:52
Establ
               VR-1.inet.0: 1/1/1/0
```

```
user@srxa2# run show bfd session
                                                             Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              2.0.0.1                  Up                       3.000     1.000        3
              4.1.0.2                  Up        ae1.100        0.900     0.300        3
              11.1.1.25                Up        ae1.0          0.900     0.300        3
              11.1.1.27                Up        ae1.1          0.900     0.300        3
```

```
user@srxa2# run show route 100.1.1.4
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 04:27:30, localpref 100
                                   AS path: 2000 2500 I, validation-state: unverified
                                 >  to 11.1.1.27 via ae1.1
```

```
user@srxa2# run show route 140/8
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              140.0.0.0/8        *[BGP/170] 04:27:34, localpref 100
```

```
                                  AS path: 1000 1500 I, validation-state: unverified
                                >  to 11.1.1.25 via ae1.0
```

```
user@srxa2# run show route 141/8
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 04:27:38, localpref 100
                                  AS path: 1000 1500 I, validation-state: unverified
                                >  to 11.1.1.25 via ae1.0
```

```
user@srxa2# run show security flow session source-prefix 140.6.219.193 source-port 22279
              Session ID: 17188740466, Policy name: SNAT_NAPT44_POLICY/6, HA State: Warm,
Timeout: 138, Session State: Valid
               In: 140.6.219.193/22279 --> 100.1.1.77/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts:
3225, Bytes: 3433324, HA Wing State: Warm,
               Out: 100.1.1.77/70 --> 192.168.65.156/2480;tcp, Conn Tag: 0x0, If: ae1.1, Pkts:
2928, Bytes: 3430574, HA Wing State: Warm,
              Total sessions: 1
```

```
user@srxa2# run show security flow session source-prefix 141.5.93.243 source-port 11462
              Session ID: 17234757046, Policy name: SFW_POLICY/4, HA State: Warm, Timeout: 128,
Session State: Valid
               In: 141.5.93.243/11462 --> 100.1.1.238/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts:
3521, Bytes: 3750136, HA Wing State: Warm,
               Out: 100.1.1.238/70 --> 141.5.93.243/11462;tcp, Conn Tag: 0x0, If: ae1.1, Pkts:
3063, Bytes: 3705524, HA Wing State: Warm,
              Total sessions: 1
```

### Verify MNHA Pair 2 (Active Node) Configuration

```
user@srxb1# run show chassis high-availability information
              Node failure codes:
                 HW  Hardware monitoring    LB  Loopback monitoring
                 MB  Mbuf monitoring        SP  SPU monitoring
                 CS  Cold Sync monitoring   SU  Software Upgrade
              Node Status: ONLINE
              Local-id: 1
              Local-IP: 2.0.0.2
              HA Peer Information:
                 Peer Id: 2        IP address: 4.0.0.2       Interface: lo0.0
                 Routing Instance: MNHA-VR
                 Encrypted: NO     Conn State: UP
                 Configured BFD Detection Time: 3 * 1000ms
                 Cold Sync Status: COMPLETE
              Services Redundancy Group: 0
                     Current State: ONLINE
                Peer Information:
                       Peer Id: 2
```

```
user@srxb1# run show bgp summary
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 5 Peers: 5 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              inet.0
                                    0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              2.2.0.2               4000       1617       1654       0       0    12:34:09
Establ
               MNHA-VR.inet.0: 3/3/3/0
              10.1.1.9              1000       1619       1656       0       0    12:34:17
Establ
               VR-1.inet.0: 2/2/2/0
              10.1.1.11             2000       1617       1658       0       0    12:34:13
Establ
               VR-1.inet.0: 1/1/1/0
```

```
user@srxb1# run show bfd session
                                                              Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              2.2.0.2                  Up        ae1.100         0.900     0.300        3
              4.0.0.2                  Up                        3.000     1.000        3
              10.1.1.9                 Up        ae1.0          0.900     0.300        3
              10.1.1.11                Up        ae1.1          0.900     0.300        3
```

```
user@srxb1# run show route 100.1.1.4
              inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[Static/5] 3w2d 02:44:07
                                 >  to 10.102.31.254 via fxp0.0
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 06:37:25, localpref 100
                                   AS path: 2000 2500 I, validation-state: unverified
                                 >  to 10.1.1.11 via ae1.1
```

```
user@srxb1# run show route 140/8
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              140.0.0.0/8        *[BGP/170] 06:37:29, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.9 via ae1.0
```

```
user@srxb1# run show route 141/8
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 06:37:32, localpref 100
                                   AS path: 1000 1500 I, validation-state: unverified
                                 >  to 10.1.1.9 via ae1.0
```

```
user@srxb1# run show security flow session source-prefix 140.6.176.20 source-port 55512
              Session ID: 11163575, Policy name: SNAT_NAPT44_POLICY/5, HA State: Active,
Timeout: 1798, Session State: Valid
               In: 140.6.176.20/55512 --> 100.1.1.73/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts:
45611, Bytes: 48604015, HA Wing State: Active,
               Out: 100.1.1.73/70 --> 192.169.117.14/26405;tcp, Conn Tag: 0x0, If: ae1.1, Pkts:
```

```
39332, Bytes: 48235684, HA Wing State: Active,
              Total sessions: 1
```

```
user@srxb1# run show security flow session source-prefix 141.2.253.107 source-port
28793              Session ID: 11163145, Policy name: SFW_POLICY/4, HA State: Active, Timeout:
1798, Session State: Valid
               In: 141.2.253.107/28793 --> 100.1.1.216/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts:
276097, Bytes: 294266524, HA Wing State: Active,
               Out: 100.1.1.216/70 --> 141.2.253.107/28793;tcp, Conn Tag: 0x0, If: ae1.1, Pkts:
238530, Bytes: 291926528, HA Wing State: Active,
              Total sessions: 1
```

### Verify MNHA Pair 2 (Standby Node) Configuration

```
user@srxb2# run show chassis high-availability information
              Node failure codes:
                 HW  Hardware monitoring    LB  Loopback monitoring
                 MB  Mbuf monitoring        SP  SPU monitoring
                 CS  Cold Sync monitoring   SU  Software Upgrade
              Node Status: ONLINE
              Local-id: 2
              Local-IP: 4.0.0.2
              HA Peer Information:
                 Peer Id: 1        IP address: 2.0.0.2       Interface: lo0.0
                 Routing Instance: MNHA-VR
                 Encrypted: NO     Conn State: UP
                 Configured BFD Detection Time: 3 * 1000ms
                 Cold Sync Status: COMPLETE
              Services Redundancy Group: 0
                     Current State: ONLINE
                Peer Information:
                       Peer Id: 1
```

```
user@srxb2# run show bgp summary
              Threading mode: BGP I/O
              Default eBGP mode: advertise - accept, receive - accept
              Groups: 5 Peers: 5 Down peers: 0
              Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
              inet.0
                                     0          0          0          0          0          0
              Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
              4.2.0.2               4000       1644       1683       0       0    12:46:04
Establ
               MNHA-VR.inet.0: 3/3/3/0
              11.1.1.33             1000       1645       1683       0       0    12:46:05
Establ
               VR-1.inet.0: 2/2/2/0
              11.1.1.35             2000       1644       1685       0       0    12:46:01
Establ
               VR-1.inet.0: 1/1/1/0
              11:2:2::1             1000       1642       1683       0       0    12:45:54
Establ
               VR-1.inet6.0: 0/0/0/0
              11:2:2::3             2000       1644       1681       0       0    12:45:50
Establ
               VR-1.inet6.0: 1/1/1/0
```

```
user@srxb2# run show bfd session
                                                       Detect   Transmit
              Address                  State     Interface      Time     Interval  Multiplier
              2.0.0.2                  Up                       3.000     1.000        3
              4.2.0.2                  Up        ae1.100        0.900     0.300        3
              11.1.1.33                Up        ae1.0          0.900     0.300        3
              11.1.1.35                Up        ae1.1          0.900     0.300        3
              11:2:2::1                Up        ae1.0          0.900     0.300        3
              11:2:2::3                Up        ae1.1          0.900     0.300        3
              6 sessions, 6 clients
              Cumulative transmit rate 17.7 pps, cumulative receive rate 17.7 pps
```

```
user@srxb2# run show route 100.1.1.4
              inet.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[Static/5] 3w2d 02:59:12
                                 >  to 10.102.24.254 via fxp0.0
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              0.0.0.0/0          *[BGP/170] 06:52:14, localpref 100
```

```
                                AS path: 2000 2500 I, validation-state: unverified
                              >  to 11.1.1.35 via ae1.1
```

```
user@srxb2# run show route 140/8
               VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              140.0.0.0/8        *[BGP/170] 06:52:19, localpref 100
                                AS path: 1000 1500 I, validation-state: unverified
                              >  to 11.1.1.33 via ae1.0
```

```
user@srxb2# run show route 141/8
              VR-1.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
              + = Active Route, - = Last Active, * = Both
              141.0.0.0/8        *[BGP/170] 06:52:23, localpref 100
                                AS path: 1000 1500 I, validation-state: unverified
                              >  to 11.1.1.33 via ae1.0
```

```
user@srxb2# run show security flow session source-prefix 140.6.176.20 source-port 55512
              Session ID: 12493056, Policy name: SNAT_NAPT44_POLICY/5, HA State: Warm, Timeout:
414, Session State: Valid
               In: 140.6.176.20/55512 --> 100.1.1.73/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts:
3345, Bytes: 3566309, HA Wing State: Warm,
               Out: 100.1.1.73/70 --> 192.169.117.14/26405;tcp, Conn Tag: 0x0, If: ae1.1, Pkts:
3018, Bytes: 3541634, HA Wing State: Warm,
              Total sessions: 1
```

```
user@srxb2# run show security flow session source-prefix 141.2.253.107 source-port 28793
              Session ID: 12492626, Policy name: SFW_POLICY/4, HA State: Warm, Timeout: 412,
Session State: Valid
               In: 141.2.253.107/28793 --> 100.1.1.216/70;tcp, Conn Tag: 0x0, If: ae1.0, Pkts:
21518, Bytes: 22929255, HA Wing State: Warm,
               Out: 100.1.1.216/70 --> 141.2.253.107/28793;tcp, Conn Tag: 0x0, If: ae1.1, Pkts:
18881, Bytes: 22787128, HA Wing State: Warm,
              Total sessions: 1
```

# Configure Junos Node Unifier for CSDS

SUMMARY

Use this configuration example to configure Junos Node Unifier (JNU) for unified management of network devices in your Connected Security Distributed Services (CSDS) topology.

IN THIS SECTION

Junos Node Unifier (JNU) provides unified command line interface (CLI) view of all the nodes present in Connected Security Distributed Services (CSDS) topology. The JNU controller and the JNU satellite communicate over the management network. In this configuration example, you'll see how to configure JNU with dual controllers. If you've one JNU controller in your JNU topology, we've indicated steps that you can skip. Note that the controllers have dual Routing Engines (RE), *re0* and *re1*, to continue forwarding packets, even if one RE fails.

**TIP**:
**Table 26: Time Estimates**

| | |
|---|---|
| Reading Time | Less than an hour |
| Configuration Time | Less than an hour |

## Example Prerequisites

**Table 27: Requirements**

| | |
|---|---|
| Hardware requirements | • Juniper Networks® MX304 for JNU controllers<br>• Juniper Networks® SRX4600 for JNU satellites |
| Software requirements | • Junos OS Release 24.4R1 |

Ensure you've completed basic configuration of MX Series and SRX Series Firewalls, and the nodes can communicate with each other over the management network.

## Before You Begin

**Table 28: Resources, and Additional Information**

| | |
|---|---|
| Understand JNU for CSDS | Configure JNU to manage the network devices in the Connected Security Distributed Services (CSDS) architecture using a single touch point management solution. You can perform the following tasks using JNU:<br>• Configure and manage the nodes using the Junos OS configuration commands.<br>• Run the Junos OS operational mode commands. |
| Know more | • Connected Security Distributed Services Architecture Deployment Guide<br>• Release Notes: Connected Security Distributed Services Architecture |

## Functional Overview

**Table 29: Junos Node Unifier Functional Overview**

| | |
|---|---|
| JNU controller | JNU controller node is a centralized entity presenting the unified CLI view of multiple network devices that are added as JNU satellites. This node runs *jnud* process to present unified user experience and uses remote procedure calls (RPC) to communicate with JNU satellites. |
| JNU satellites | JNU satellites operate under the control of the JNU controller. These are the nodes that run security services. The *jnud* process also runs in the satellites. |
| Primary verification tasks | Verify the following:<br>1. JNU controller lists the JNU satellites present in the JNU topology. |

## Topology Overview

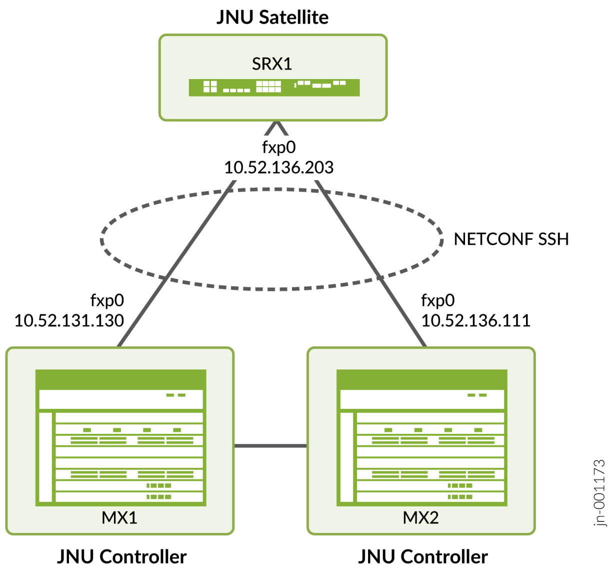
**Table 30: Devices, Role, and Functionality used in this Configuration**

| Hostname | Role | Function |
|---|---|---|
| MX1 | JNU controller | Serves as a CLI touchpoint for all network devices in the JNU topology.<br>• re0 IP address - 10.52.136.131/8<br>• re1 IP address - 10.52.136.132/8<br>• master-only IP address - 10.52.131.130/8<br>The master-only IP address is an additional IP address configured on the management interface of both REs. This address is active only on the primary RE and moves to the new primary RE during a graceful Routing Engine switchover (GRES). |
| MX2 | JNU controller (second controller) | Serves as a second controller for high availability<br>• re0 IP address - 10.52.136.112/8<br>• re1 IP address - 10.52.136.113/8<br>• master-only IP address - 10.52.136.111/8 |
| SRX1 | JNU satellite | Node in JNU topology that you can manage using the JNU controller.<br>• IP address - 10.52.130.203/8<br>For every new satellite use a unique IP address. |

The nodes use `fxp0` management interface for communication between controller and satellites.

## Topology Illustration

**Figure 31: JNU Topology for CSDS**



## Step-By-Step Prerequisite Configuration on MX1, MX2, and SRX1

1. Enable SSH and NETCONF services on MX1, MX2, and SRX1.

```
[edit]
user@mx1# set system services ssh
user@mx1# set system services netconf ssh
```

2. Ensure SSH keys are available on controller and satellites before configuring JNU controller.

   You can either generate SSH keys manually or use the auto-generated keys.

   The command `request jnu role controller invoke-on all-routing-engines` on the controller and the command `request jnu role satellite` on the satellite verify the presence of custom SSH keys in /var/db/jnu/.ssh directory. If the keys are missing, these commands generate a new pair of SSH keys. If the keys already exists, the command refrains from overwriting them.

To generate custom SSH keys on controller and satellites, follow steps a to d. If you are generating custom SSH keys, generate the keys for both the REs, *re0* and *re1* seperately for the controller. On each of the controller REs and satellites run the following steps.

In this example topology run the following steps for MX1 *re0*, MX1 *re1*, MX2 *re0*, MX2 *re1*, and SRX1. If you have a single controller, run these steps for MX1 *re0*, MX1 *re1*, and SRX1:

a. At the shell prompt, create a directory to store SSH key pairs.

```
user@host:~# mkdir -p /var/db/jnu/.ssh
```

b. Create authentication key pair for SSH.

```
user@host:~# echo 'y' | ssh-keygen -t rsa -f /var/db/jnu/.ssh/id_rsa -N "" -b 2048
```

In this configuration, we created the keys with RSA 2048-bit encryption. We support only RSA-based encryption.

c. Get the public key.

```
user@host:~# cat /var/db/jnu/.ssh/id_rsa.pub
$ABC123c1r0
```

d. (Only for controllers) For the controller, to generate SSH keys on the other RE, run the following command in operational mode to login to the other RE, and repeat step 1 and 2. Ignore this step for the satellite:

```
user@host request routing-engine login other-routing-engine
```

Note the public keys that you generate to configure the JNU nodes. In the example topology, following are the public keys for MX1 *re0*, MX1 *re1*, MX2 *re0*, MX2 *re1*, and SRX1.

- `$ABC123c1r0` is the public key of MX1 *re0*.
- `$ABC123c1r1` is the public key of MX1 *re1*.
- `$ABC123c2r0` is the public key of MX2 *re0*.
- `$ABC123c2r1` is the public key of MX2 *re1*.
- `$ABC123` is the public key of SRX1.

## Step-By-Step Controller Configuration on MX1

Ensure SSH key pair on MX1 *re0*, MX1 *re1*, MX2 *re0*, MX2 *re1*, and SRX1 are available and you've noted their public keys.

Configure JNU controller on MX1. Run the following steps.

**NOTE**: For complete sample configurations on the DUT, see:

- "Set Commands on MX1" on page 206

1. Configure JNU controller role on all REs. Run this command on *re0*.

```
user@mx1> request jnu role controller invoke-on all-routing-engines
```

2. Configure controller auto-login into the other JNU nodes for the jnuadmin user. Repeat this step for every satellite that the controller manages.

   a. Set system login on MX1 for SRX1 using SRX1's public key.

   ```
   [edit]
   user@mx1# set system login user jnuadmin class super-user authentication ssh-rsa $ABC123
   ```

   b. Set system login on MX1 for MX2 using MX2 re0's public key.

   ```
   [edit]
   user@mx1# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c2r0
   ```

   Ignore this step if you've a single controller in your JNU topology.

   c. Set system login on MX1 for MX2 using MX2 re1's public key.

   ```
   [edit]
   user@mx1# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c2r1
   user@mx1# commit
   ```

   Ensure to commit the configuration at this step.

Ignore this step if you've a single controller in your JNU topology. Ensure to commit the configuration at **step a** if you've a single controller.

d. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to SRX1 with jnuadmin user.

```
user@mx1:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.130.203 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

e. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX2 *re0* with jnuadmin user.

```
user@mx1:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.112 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

Ignore this step if you have single controller in your JNU topology.

f. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX2 *re1* with jnuadmin user.

```
user@mx1:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.113 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

Ignore this step if you have single controller in your JNU topology.

3. Configure JNU management features on JNU controller.

a. Enable feature-rich mode.

```
[edit]
user@mx1# set chassis jnu-management mode feature-rich
```

b. Associate jnuadmin user for JNU management tasks.

```
[edit]
user@mx1# set chassis jnu-management user jnuadmin
```

c. Associate other controller for JNU management tasks using its master-only IP.

```
[edit]
user@mx1# set chassis jnu-management other-controller 10.52.136.111/8
```

Ignore this step if you have single controller in your JNU topology.

4. Configure dual RE settings.

a. Configure commit synchronization on the REs by default, enable graceful switchover to the other RE in the event of failure of active RE, and enable non-stop routing.

```
[edit]
user@mx1# set system commit synchronize
user@mx1# set chassis redundancy graceful-switchover
user@mx1# set routing-options nonstop-routing
```

b. Configure master-only settings for *re0* and *re1*.

```
[edit]
user@mx1# set groups re0 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-
only
user@mx1# set groups re1 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-
only
user@mx1# commit
```

Ensure to commit the configuration at this step.

## Step-By-Step Controller Configuration on MX2

Ensure SSH key pair on MX1 *re0*, MX1 *re1*, MX2 *re0*, MX2 *re1*, and SRX1 are available and you've noted their public keys.

Configure JNU controller on MX2. Run the following steps.

**NOTE**: For complete sample configurations on the DUT, see:

- "Set Commands on MX2" on page 206

1. Configure JNU controller role on all REs. Run this command on *re0*.

```
user@mx2> request jnu role controller invoke-on all-routing-engines
```

2. Configure controller auto-login into the other JNU nodes for the jnuadmin user. Repeat this step for every satellite that the controller manages.

   a. Set system login on MX2 for SRX1 using SRX1's public key.

   ```
   [edit]
   user@mx2# set system login user jnuadmin class super-user authentication ssh-rsa $ABC123
   ```

   b. Set system login on MX2 for MX1 using MX1 re0's public key.

   ```
   [edit]
   user@mx2# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c1r0
   ```

   Ignore this step if you've single controller in your JNU topology.

   c. Set system login on MX2 for MX1 using MX1 re1's public key.

   ```
   [edit]
   user@mx2# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c1r1
   user@mx2# commit
   ```

   Ensure to commit the configuration at this step.

   Ignore this step if you've single controller in your JNU topology. Ensure to commit the configuration at **step a** if you've a single controller.

d. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to SRX1 with jnuadmin user.

```
user@mx2:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.130.203 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

e. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX1 *re0* with jnuadmin user.

```
user@mx2:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.131 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

Ignore this step if you have a single controller in your JNU topology.

f. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX1 *re1* with jnuadmin user.

```
user@mx2:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.132 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

Ignore this step if you have a single controller in your JNU topology.

3. Configure JNU management features on JNU controller.

a. Enable feature-rich mode.

```
[edit]
user@mx2# set chassis jnu-management mode feature-rich
```

b. Associate jnuadmin user for JNU management tasks.

```
[edit]
user@mx2# set chassis jnu-management user jnuadmin
```

c. Associate other controller for JNU management tasks using its master-only IP.

```
[edit]
user@mx2# set chassis jnu-management other-controller 10.52.131.130/8
```

Ignore this step if you have single controller in your JNU topology.

4. Configure dual RE settings.

   a. Configure commit synchronization on the REs by default, enable graceful switchover to the other RE in the event of failure of active RE, and enable non-stop routing.

```
[edit]
user@mx2# set system commit synchronize
user@mx2# set chassis redundancy graceful-switchover
user@mx2# set routing-options nonstop-routing
```

   b. Configure master-only settings for *re0* and *re1*.

```
[edit]
user@mx2# set groups re0 interfaces fxp0 unit 0 family inet address 10.52.136.111/8 master-
only
user@mx2# set groups re1 interfaces fxp0 unit 0 family inet address 110.52.136.111/8
master-only
user@mx2# commit
```

Ensure to commit the configuration at this step.

## Step-By-Step Satellite Configuration on SRX1

Before configuring the satellites, ensure you've configured the controllers. Ensure SSH key pair on SRX1 is available and you've noted its public key.

Configure JNU satellite on SRX1. Run the following steps. Repeat these steps for every satellite in your topology and adjust the configuration as per your topology.

**NOTE**: For complete sample configurations on the DUT, see:

• "Set Commands on SRX1" on page 207

1. Configure JNU satellite role.

```
user@srx1> request jnu role satellite
```

2. Configure satellite auto-login into the controllers for the jnuadmin user.

   a. Set system login on SRX1 for MX1 using MX1 re0's public key.

   ```
   [edit]
   user@srx1# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c1r0
   ```

   b. Set system login on SRX1 for MX1 using MX1 re1's public key.

   ```
   [edit]
   user@srx1# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c1r1
   ```

   c. Set system login on SRX1 for MX2 using MX2 re0's public key.

   ```
   [edit]
   user@srx1# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c2r0
   ```

   Ignore this step if you've a single controller in your JNU topology.

   d. Set system login on SRX1 for MX2 using MX2 re1's public key.

   ```
   [edit]
   user@srx1# set system login user jnuadmin class super-user authentication ssh-rsa
   $ABC123c2r1
   ```

   Ensure to commit the configuration at this step.

   Ignore this step if you've single controller in your JNU topology. Ensure to commit the configuration at **step b** if you've a single controller.

e. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX1 *re0* with jnuadmin user.

```
user@srx1:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.131 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

f. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX1 *re1* with jnuadmin user.

```
user@srx1:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.132 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

g. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX2 *re0* with jnuadmin user.

```
user@srx1:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.112 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

Ignore this step if you have a single controller in your JNU topology.

h. In the shell prompt, add the SSH keys to the list of known hosts. For the first time SSH login to MX2 *re1* with jnuadmin user.

```
user@srx1:~# ssh -i '/var/db/jnu/.ssh/id_rsa' -o 'StrictHostKeyChecking no'
jnuadmin@10.52.136.113 -s netconf
```

This will prompt for Yes/No. Press Yes, hit enter and quit (press Ctrl + C).

Ignore this step if you have a single controller in your JNU topology.

3. Configure JNU management features on JNU satellite.

   a. Enable feature-rich mode.

```
[edit]
user@srx1# set chassis jnu-management mode feature-rich
```

b. Associate satellite for JNU management tasks.

```
[edit]
user@srx1# set chassis jnu-management satellite-name 10.52.130.203
```

c. Associate jnuadmin user for JNU management tasks.

```
[edit]
user@srx1# set chassis jnu-management user jnuadmin
```

d. Associate controller, MX1, for JNU management tasks.

```
[edit]
user@srx1# set chassis jnu-management controller 10.52.131.130
```

e. Associate controller, MX2, for JNU management tasks.

```
[edit]
user@srx1# set chassis jnu-management controller 10.52.136.111
```

Ensure to commit the configuration in this step.

Ignore this step if you have a single controller in your JNU topology and commit the configuration in **step e**

## Verification

**IN THIS SECTION**

This section provides a list of show commands that you can use to verify the feature in this example.

| Command | Verification Task |
|---|---|
| show chassis jnu satellites | Verify JNU nodes synchronization. |
| show configuration chassis jnu-management | Verify nodes in JNU topology. |

## Verify JNU Nodes Synchronization

### Purpose

Run the command to check that JDM and vSRX Virtual Firewalls are synchronized. Both the controllers, MX1 and MX2, lists the satellite that you add. Satellites push their schema to the controller during the initial synchronization.

### Action

From operational mode, run `show chassis jnu satellites` command on MX1 and MX2 controller to verify that the satellites are added to the controllers.

```
user@mx1> show chassis jnu satellites
Satellite           Alive     Model     Version
-----------------------------------------------------
10.52.130.203       up        vSRX      24.4I-20241106.0.1958
```

```
user@mx2> show chassis jnu satellites
Satellite           Alive     Model     Version
-----------------------------------------------------
10.52.130.203       up        vSRX      24.4I-20241106.0.1958
```

### Meaning

Controller list the satellite's name, status, model, and Junos OS version. It takes approximately 5-6 minutes for synchronization per controller per satellite.

## Verify Nodes in JNU Topology

### Purpose

Run the command on the controllers and satellite to know the details of nodes in JNU topology.

### Action

From operational mode, run `show configuration chassis jnu-management` command on MX1, MX2 and SRX1 to verify the nodes in JNU topology.

```
user@mx1> show configuration chassis jnu-management
mode feature-rich;
satellite 10.52.130.203 {
    model vSRX;
    version 24.4I-20241106.0.1958;
}
user jnuadmin;
other-controller 10.52.136.111;
```

```
user@mx2> show configuration chassis jnu-management
mode feature-rich;
satellite 10.52.130.203 {
    model vSRX;
    version 24.4I-20241106.0.1958;
}
user jnuadmin;
other-controller 10.52.131.130;
```

```
user@srx1> show configuration chassis jnu-management
mode feature-rich;
satellite-name 10.52.130.203;
user jnuadmin;
controller [ 10.52.131.130 10.52.136.111 ];
```

### Meaning

The command shows the details of the controller and the satellite nodes.

# Appendix 1: Set Commands on All Devices

Set command output on all devices.

### Set Commands on MX1

```
set groups re0 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-only
set groups re0 interfaces fxp0 unit 0 family inet address 10.52.136.131/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.132/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-only
set groups global system root-authentication encrypted-password “$ABC123”
set groups global system login user remote uid 2000
set groups global system login user remote class super-user
set groups global system services netconf ssh
set groups global system services ssh root-login allow
set system commit synchronize
set system login user jnuadmin uid 2001
set system login user jnuadmin class super-user
set system login user jnuadmin authentication ssh-rsa "$ABC123c1r0"
set system login user jnuadmin authentication ssh-rsa "$ABC123c1r1"
set system login user jnuadmin authentication ssh-rsa "$ABC123"
set system services netconf ssh
set system ports console log-out-on-disconnect
set chassis redundancy graceful-switchover
set chassis jnu-management mode feature-rich
set chassis jnu-management user jnuadmin
set chassis jnu-management other-controller 10.52.136.111
set routing-options nonstop-routing
```

### Set Commands on MX2

```
set groups re0 interfaces fxp0 unit 0 family inet address 10.52.136.111/8 master-only
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.112/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.113/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.111/8 master-only
set groups global system root-authentication encrypted-password “$ABC123”
set groups global system login user remote uid 2000
set groups global system login user remote class super-user
set groups global system services netconf ssh
set groups global system services ssh root-login allow
set system commit synchronize
set system login user jnuadmin uid 2001
set system login user jnuadmin class super-user
set system login user jnuadmin authentication ssh-rsa "$ABC123"
set system login user jnuadmin authentication ssh-rsa "$ABC123c2r0”
set system login user jnuadmin authentication ssh-rsa "$ABC123c2r1”
set system services netconf ssh
set system ports console log-out-on-disconnect
set chassis redundancy graceful-switchover
set chassis jnu-management mode feature-rich
set chassis jnu-management user jnuadmin
set chassis jnu-management other-controller 10.52.131.130
set routing-options nonstop-routing
```

### Set Commands on SRX1

```
set groups member0 interfaces fxp0 unit 0 family inet address 10.52.130.203/8
set groups global system root-authentication encrypted-password "$ABC123”
set groups global system login user remote uid 2000
set groups global system login user remote class super-user
set groups global system services netconf ssh
set groups global system services ssh root-login allow
set system host-name shanv1r
set system login user jnuadmin uid 2001
set system login user jnuadmin class super-user
set system login user jnuadmin authentication ssh-rsa “$ABC123c1r0”
set system login user jnuadmin authentication ssh-rsa “$ABC123c1r1”
set system login user jnuadmin authentication ssh-rsa “$ABC123c2r0”
set system login user jnuadmin authentication ssh-rsa "“$ABC123c2r1”
set system services netconf ssh
set system ports console log-out-on-disconnect
set chassis jnu-management mode feature-rich
set chassis jnu-management satellite-name 10.52.130.203
set chassis jnu-management user jnuadmin
set chassis jnu-management controller 10.52.136.111
set chassis jnu-management controller 10.52.131.130
```

## Appendix 2: Show Configuration Output on All Devices

Show command output on all devices.

### Show Command on MX1

```
user@mx1# show chassis jnu-management
Nov 07 23:21:42
mode feature-rich;
satellite 10.52.130.203 {
    model vSRX;
    version 24.4I-20241106.0.1958;
}
user jnuadmin;
other-controller 10.52.131.130;
```

### Show Command on MX2

```
user@mx2# show chassis jnu-management
Nov 07 23:21:47
mode feature-rich;
satellite 10.52.130.203 {
    model vSRX;
    version 24.4I-20241106.0.1958;
}
user jnuadmin;
other-controller 10.52.136.111;
```

### Show Command on SRX1

```
user@srx1# show chassis jnu-management
mode feature-rich;
satellite-name 10.52.130.203;
```

```
user jnuadmin;
controller [ 10.52.136.111 10.52.131.130 ];
```

SEE ALSO

JNU Use Cases for CSDS | 50

## Install and Configure Junos Device Manager for CSDS

SUMMARY

Use this configuration example to install and configure Junos Device Manager (JDM) for vSRX orchestartion in Connected Security Distributed Services (CSDS) topology.

IN THIS SECTION

Junos Device Manager (JDM) is a Linux container that orchestrates vSRX Virtual Firewalls for services layer in Connected Security Distributed Services (CSDS) topology. This configuration example demonstrates how to install and configure JDM for vSRX orchestration. Before proceeding with the installation and configuration of JDM, ensure that the Junos Node Unifier (JNU) is already set up for CSDS.

This example also includes setting up JNU with dual node controllers, assuming it hasn't been configured yet. Even when you've already configured JNU, you can still run JNU-related configuration commands as the system ensures that the existing JNU configurations remain unchanged. If you have a single JNU controller in your topology, we've indicated the steps that you can ignore. Once JDM is installed and configured, both JDM and vSRX Virtual Firewalls will be added as JNU satellites.

**TIP**:

**Table 31: Readability Score and Time Estimates**

| | |
|---|---|
| Reading Time | Less than an hour |
| Configuration Time | Less than two hours |

## Example Prerequisites

**Table 32: Requirements for JDM**

| | |
|---|---|
| Hardware requirements | • Intel Xeon Gold 6438N 2GHz with Ubuntu 22.04.4 LTS OS |
| Software requirements | • JDM package: csds-jdm-24.4-R1.x.x86_64.deb<br>• vSRX image: junos-vsrx3-x86-64-24.4.qcow2 |

**Table 33: Requirements for JNU**

| | |
|---|---|
| Hardware requirements | • Juniper Networks® MX304 for JNU controllers |
| Software requirements | • Junos OS Release 24.4R1 |

- Ensure you've installed Ubuntu OS on baremetal host server with the specified software and hardware requirements. See "JDM Components for CSDS" on page 58

- Ensure you understand the modifications that the JDM installation is set to apply on the host server. See "Understand vSRX Orchestration with JDM for CSDS" on page 65.
- Ensure your host server has a management IP address.
- Ensure you've completed basic configuration of MX Series and the nodes can communicate with each other over the management network.

## Before You Begin

**Table 34: Resources, and Additional Information**

| | |
|---|---|
| Understand JDM for CSDS | Configure JDM to orchestrate the life cycle management of the vSRX Virtual Firewalls. Use JDM only when you plan to include vSRX Virtual Firewalls in your CSDS architecture. You require JNU to configure JDM. JDM is the satellite and uses the MX Series controller for its configuration. See "Junos Device Manager for CSDS" on page 57 |
| Know more | • "JDM Components for CSDS" on page 58<br>• "Understand vSRX Orchestration with JDM for CSDS" on page 65 |
| Learn more | • "Configure Junos Node Unifier for CSDS" on page 189<br>• Connected Security Distributed Services Architecture Deployment Guide<br>• Release Notes: Connected Security Distributed Services Architecture |

## Functional Overview

**Table 35: Junos Device Manager Functional Overview**

| | |
|---|---|
| **Ubuntu host server** | Ubutu host server is a baremetal server for hosting JDM container and vSRX Virtual Firewalls spawned by JDM. This node is not a JNU satellite. |

| | |
|---|---|
| **JDM container** | JDM is a Linux Container (LXC) that runs in the host to perform vSRX orchestration. The *jnud* process runs in JDM. JDM is a JNU satellite node. |
| **vSRX Virtual Firewall** | JDM spawns the vSRX Virtual Firewalls for CSDS services plane. The *jnud* process runs in the firewall. vSRX Virtual Firewall is a JNU satellite node. |
| **JNU controller** | Even though the JNU controller doesn't belong to the JDM components, you still require the JNU controller for installing and configuring JDM. The MX Series acts as the controller. |
| **Primary verification tasks** | Verify the following:<br>1. JDM status from the controller.<br>2. JNU nodes and their synchronization status in JNU topology.<br>3. The controller lists the JDM and vSRX Virtual Firewall as the satellites.<br>4. Run the satellite's operational commands from the controller. |

## Topology Overview

**Table 36: Devices, Role, and Functionality used in this Configuration**

| Hostname | Role | Function |
|---|---|---|
| MX1 | JNU controller | This serves as a CLI touchpoint for all network devices in the JNU topology.<br>• re0 IP address - 10.52.136.131/8<br>• re1 IP address - 10.52.136.132/8<br>• master-only IP address- 10.52.131.130/8<br>The master-only IP address is an additional IP address configured on the management interface of both Routing Engines (REs). This address is active only on the primary RE and moves to the new primary RE during a graceful Routing Engine switchover (GRES). |
| MX2 | JNU controller (second controller) | This serves as a second the controller for high availability.<br>• re0 IP address - 10.52.136.112/8<br>• re1 IP address - 10.52.136.113/8<br>• master-only IP address- 10.52.136.111/8 |
| Ubuntu host server | Baremetal server | The baremetal server hosts JDM and vSRX Virtual Firewalls. This host is not part of JNU topology.<br>• IP address - 10.157.79.104/19 |
| JDM | JNU satellite | Node in JNU topology that creates vSRX Virtual Firewall.<br>• IP address - 10.157.80.182/19<br>The system allocates the first IP address within the specified IP prefix range in `ip-prefix-range` option to JDM. |

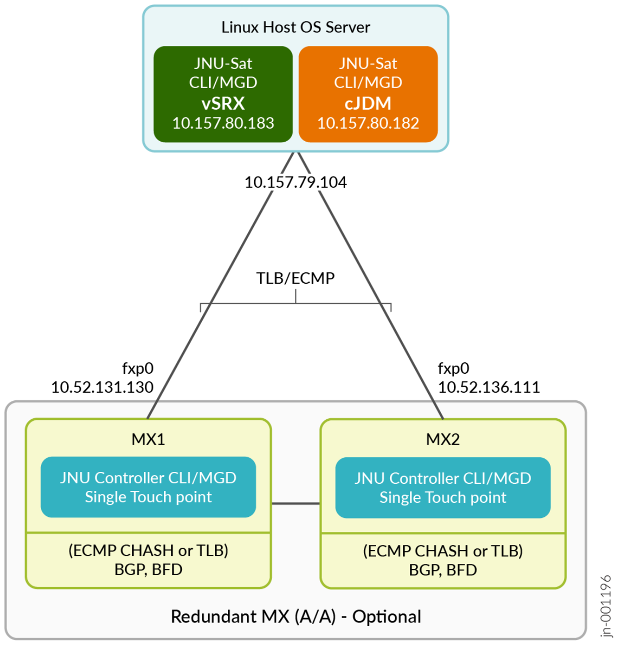
**Table 36: Devices, Role, and Functionality used in this Configuration *(Continued)***

| Hostname | Role | Function |
|---|---|---|
| vSRX1 | JNU satellite | Node in JNU topology that you can manage using the JNU Controller.<br>• IP address - 10.157.80.183/19<br>The system allocates the second and the subsequent IP addresses within the specified IP prefix range in `ip-prefix-range` option to vSRX Virtual Firewalls. |

The nodes use `fxp0` management interface for communication between the controller and the satellites.

## Topology Illustration

**Figure 32: JDM Topology for CSDS**


Linux Host OS Server
JNU-Sat CLI/MGD vSRX 10.157.80.183
JNU-Sat CLI/MGD cJDM 10.157.80.182
10.157.79.104
TLB/ECMP
fxp0 10.52.131.130
fxp0 10.52.136.111
MX1
JNU Controller CLI/MGD Single Touch point
(ECMP CHASH or TLB) BGP, BFD
MX2
JNU Controller CLI/MGD Single Touch point
(ECMP CHASH or TLB) BGP, BFD
Redundant MX (A/A) - Optional
jn-001196

## Step-By-Step Controller Configuration on MX1

Ensure you meet the following prerequisites:

- Download the JDM software package (`csds-jdm-jdm-<release-number>.x86_64.deb`) and the vSRX Virtual Firewall image (`junos-vsrx3-x86-64-24.4.qcow2`) from the Juniper Networks Downloads page at https://support.juniper.net/support/downloads/. Place the images on the controller.
- Enable NETCONF and SSH services.

```
[edit]
user@mx2# set system services netconf ssh
user@mx2# set system services ssh
```

Configure JNU controller on MX1 and install JDM on the host. Run the following steps.

**NOTE**: For complete sample configurations on the DUT, see:

- "Set Commands on MX1" on page 226

1. Configure the controller role on all REs. Run this command on *re0*.

```
user@mx1> request jnu role controller invoke-on all-routing-engines
```

2. Configure the controller auto-login for the jnuadmin user.

```
[edit]
user@mx1# set system login user jnuadmin class super-user
```

3. Configure JNU management features on JNU controller.

   a. Enable feature-rich mode.

```
[edit]
user@mx1# set chassis jnu-management mode feature-rich
```

b. Associate jnuadmin user for JNU management tasks.

```
[edit]
user@mx1# set chassis jnu-management user jnuadmin
```

c. Associate other controller for JNU management tasks using its master-only IP.

```
[edit]
user@mx1# set chassis jnu-management other-controller 10.52.136.111/8
```

Ignore this step if you have single controller in your JNU topology.

4. Configure the host instance ID of the Ubuntu host server.

```
[edit]
user@mx1# set system csds node-instance 0 host-ip 10.157.79.104 gateway-ip 10.157.64.1 ip-
prefix-range 10.157.80.182-184/19
```

The system allocates the first IP address to the JDM and the subsequent IP addresses to the vSRX Virtual Firewalls, within the specified IP prefix range in `ip-prefix-range` option.

5. Configure dual RE settings if your controller has two Routing Engines.

a. Configure commit synchronization on the REs by default, enable graceful switchover to the other RE in the event of failure of active RE, and enable non-stop routing.

```
[edit]
user@mx1# set system commit synchronize
user@mx1# set chassis redundancy graceful-switchover
user@mx1# set routing-options nonstop-routing
```

b. Configure master-only settings for *re0* and *re1*.

```
[edit]
user@mx1# set groups re0 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-
only
user@mx1# set groups re1 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-
only
user@mx1# commit
```

Ensure to commit the configuration at this step.

6. Authenticate host server from the controller.

   ```
   user@mx1> request csds authenticate-host csds-instance-id 0
   ```

   At the prompt, enter `yes` and the host server's password.

   Every host server and its associated JDM and vSRX Virtual Firewall instances are referred in the option `csds-instance-id`. If you have multiple host servers, use a unique instance ID for each of the host servers. The command performs the controller's one-way key exchange with the host. This allows passwordless SSH access to the host from the controller. This enables the controller to remotely run commands for managing JDM in CSDS architecture.
7. Install JDM package on the host server.

   ```
   user@mx1> request csds jdm add csds-instance-id 0 image jdm-image-with-absolute-path
   ```

   You'll notice that the system identifies the server profile based on Intel or AMD configuration and installs the JDM. Installing JDM for the first time on the host server prompts the system to request a reboot with the message `KERNAL CMDLINE ARGUMENTS ARE MODIFIED !!! Host requires reboot Kindly confirm the reboot on host? Confirm (y/n)`. Type yes to reboot the host server.
8. Initiate the creation of vSRX Virtual Firewalls on the host server.

   ```
   user@mx1> request csds add-vsrx csds-instance-id 0 image vsrx-qcow2-image-with-absolute-path
   ```

   Wait for approximately 10 minutes for the vSRX Virtual Firewall installation. You'll notice that the command creates VNFs and restarts the CLI. Wait until you see the message `Satellite has been added, Please restart the CLI session`. Type yes to restart the CLI. All the Junos OS CLI sessions that you open on your terminal displays this message. But the shell prompt doesn't show the message. You can verify the vSRX Virtual Firewall logs using the VNF console logs using the command `request virtual-network-functions console device-list` *`vsrx-ip-address vnf-name`*.
9. Extract the public keys of vSRX Virtual Firewall.

   ```
   user@mx1> request csds extract-vsrx-keys csds-instance-id 0
   ```

   Wait for 5-6 minutes for the vSRX Virtual Firewall to synchronize with the controller. Wait until you see the message `Satellite has been added, Please restart the CLI session`. Type yes to restart the CLI. All the Junos OS CLI sessions that you open on your terminal displays this message. But the shell prompt doesn't show the message.

## Step-By-Step Controller Configuration on MX2

Ensure you meet the following prerequisites:

- Ignore the controller configuration on MX2 if you have a single controller in your JNU topology.
- Ensure that you have configured MX1.
- Enable NETCONF and SSH services.

```
[edit]
user@mx2# set system services netconf ssh
user@mx2# set system services ssh
```

Configure JNU controller on MX2 and synchronize with the other controller. Run the following steps.

**NOTE**: For complete sample configurations on the DUT, see

- "Set Commands on MX2" on page 227

1. Configure JNU controller role on all REs. Run this command on *re0*.

```
user@mx2> request jnu role controller invoke-on all-routing-engines
```

2. Configure controller auto-login for the jnuadmin user.

```
[edit]
user@mx2# set system login user jnuadmin class super-user
```

3. Configure JNU management features on JNU Controller.

   a. Enable feature rich mode.

```
[edit]
user@mx2# set chassis jnu-management mode feature-rich
```

b. Associate jnuadmin user for JNU management tasks.

```
[edit]
user@mx2# set chassis jnu-management user jnuadmin
```

c. Associate other controller for JNU management tasks.

```
[edit]
user@mx1# set chassis jnu-management other-controller 10.52.131.130
```

Ignore this step if you have single controller in your JNU topology.

4. Configure JNU controllers synchronization.

```
user@mx2> request csds sync-controller other-controller-ip 10.52.131.130/8
```

Ignore this step if you have single controller in your JNU topology.

5. Configure dual RE settings if your controller has two Routing Engines.

a. Configure commit synchronization on the REs by default, enable graceful switchover to the other RE in the event of failure of active RE, and enable non-stop routing.

```
[edit]
user@mx2# set system commit synchronize
user@mx2# set chassis redundancy graceful-switchover
user@mx2# set routing-options nonstop-routing
```

b. Configure master only settings for *re0* and *re1*.

```
[edit]
user@mx2# set groups re0 interfaces fxp0 unit 0 family inet address 10.52.136.111/19
master-only
user@mx2# set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.111/19
master-only
user@mx2# commit
```

Ensure to commit the configuration.

## Verification

IN THIS SECTION

This section provides a list of show commands that you can use to verify the feature in this example. Run these commands from the controller in operational mode.

| Command | Verification Task |
|---|---|
| show version device-list *jdm-satellite-ip* | Verify the JDM's version. |
| show system cpu\| memory\| network\| storage device-list *jdm-satellite-ip* | Verify the infrastructure (CPU, memory, network and storage) details of JDM. |
| show virtual-network-functions device-list *jdm-satellite-ip* | Verify the virtual network function (VNF) status on JDM. |
| show chassis jnu satellites | Verify the satellite nodes synchronization in JNU topology. |
| show configuration chassis jnu-management | Verify the nodes in JNU topology. |

| Command | Verification Task |
|---|---|
| show version device-list *satellite-ip* | Verify that you are able to run the satellite's (vSRX Series Virtual Firewall) operational commands from the controller. |
| set chassis satellite *satellite-ip* | Verify that you are able to configure the satellite's (vSRX Series Virtual Firewall) using the configuration commands from the controller. |

## Verify Satellite Version

### Purpose

Run the command to check version of the satellite.

### Action

From operational mode, run `show version device-list` *`jdm-satellite-ip`* command on the controllers, MX1 or MX2.

```
user@mx1> show version device-list device-list 10.157.80.182
Hostname: jdm
Model: csds_jdm
Family: junos
JDM package version : 24.4
Host Software [Ubuntu 22.04.2 LTS]
JDM container Software [Ubuntu 22.04 LTS]
JDM daemon jdmd [Version: 24.4]
```

### Meaning

The controller displays the JDM satellite's version details.

## Verify Satellite Infrastructure Details

### Purpose

Run the commands to fetch CPU, memory, network and storage details of JDM.

## Action

From operational mode, run `show system network storage device-list` *`jdm-satellite-ip`* command on the controllers, MX1 or MX2.

```
user@mx1> show system network device-list 10.157.80.182
Physical Interfaces
-------------------------------------------------------------------------------------------------
--------------------------------------------
Name     Index MTU     Hardware-address  Rcvd Bytes   Rcvd Packets Rcvd Error Rcvd Drop Trxd
Bytes   Trxd Packets Trxd Error Trxd Drop Flags
-------- ----- ------- ----------------- ------------ ------------ ---------- ---------
------------ ------------ ---------- --------- ------
eno1     2             ac:1f:6b:db:9a:98 96917714533  382476745    0          2552
1434989121   13207569     0          0

Per VNF Interface Statistics
-------------------------------------------------------------------------------------------------
---------------------------------------------------------
Interface                 Source        MAC Address        Rcvd Bytes   Rcvd packets Rcvd Error
Rcvd Drop Trxd bytes   Trxd Packets Trxd Error Trxd Drop
------------------------- ------------- -----------------  ------------ ------------ ----------
--------- ------------ ------------ ---------- ---------
VNF name: vnf0
macvtap23                 eno1          52:54:00:48:58:1f  11130730973  95294125     13453
13453     106174583    297660       0          0

JDM Interface Statistics
-------------------------------------------------------------------------------------------------
-----------------------------------------
Name     Index MTU   Hardware-address  Rcvd Bytes   Rcvd Packets Rcvd Error Rcvd Drop Trxd Bytes
  Trxd Packets Trxd Error Trxd Drop Flags
-------- ----- ----- ----------------- ------------ ------------ ---------- ---------
------------ ------------ ---------- --------- ------
bme1     433   1500  52:54:00:a1:f2:32 5222992      95044        0          0         315130
  1876         0          0
jmgmt0   435   1500  52:54:00:d8:19:eb 1126420395   9211539      0          14        55015
  1016         0          0
```

Similarly, you can run the following commands on JDM:

- `show system cpu device-list` *`jdm-satellite-ip`*

- `show system memory device-list` *`jdm-satellite-ip`*
- `show system storage device-list` *`jdm-satellite-ip`*

### Meaning

The controller displays the JDM satellite's CPU, memory, network and storage details.

## Verify Satellite VNF Status

### Purpose

Run the command to check VNF status of the satellite. Though the `device-list` option shows the JDM IP address, you'll not see the output of JDM as it doesn't have VNFs.

### Action

From operational mode, run `show virtual-network-functions device-list` *`satellite-name`* command on the controllers, MX1 or MX2.

```
user@mx1> show virtual-network-functions device-list 10.157.80.182
--------------------------------------------------------------------
ID       Name                                              State      Liveness
--------------------------------------------------------------------------------
113      vnf0                                              Running    alive
```

### Meaning

The controller lists the satellite's running and liveliness status. If the status of the VNF is running and liveliness is unavailable, wait until the the MX controller extracts the vSRX Virtual Firewall public keys.

## Verify JNU Nodes Synchronization

### Purpose

Run the command to check that JDM and vSRX Virtual Firewalls are synchronized with the controller. Both the controllers, MX1 and MX2, lists the satellites that you add. Satellites push their schema to the controller during the initial synchronization. Although JDM is added as a satellite, JDM doesn’t send its configuration to the controller during the initial synchronization, unlike the other satellites.

### Action

From operational mode, run `show chassis jnu satellites` command on the MX1 or MX2 controller to verify that the satellites are added to the controllers.

```
user@mx1> show chassis jnu satellites
Satellite           Alive      Model      Version
-----------------------------------------------------
10.157.80.182       up         jdm        24.4-I.20241027.0.2259
10.157.80.183       up         vSRX       24.4I-20241027.0.2259
```

### Meaning

The controller list the satellite's name, status, model, and Junos OS version. It takes approximately 5-6 minutes for synchronization per controller per satellite.

## Verify Nodes in JNU Topology

### Purpose

Run the command on the controllers to know the details of nodes in JNU topology.

### Action

From operational mode, run `show configuration chassis jnu-management` command on the MX1 or MX2 controllers to verify the nodes in JNU topology.

```
user@mx1> show configuration chassis jnu-management
mode feature-rich;
satellite 10.157.80.182 {
    model jdm;
    version 24.4I-20241106.0.1958;
}
satellite 10.157.80.183 {
    model vSRX;
    version 24.4I-20241106.0.1958;
}
user jnuadmin;
other-controller 10.52.136.111;
```

### Meaning

The command shows the details of the controller and the satellite nodes.

## Verify the Satellite's Operational Commands from the Controller

### Purpose

Run the satellite's operational commands on the controller.

### Action

Run the `show version device-list` *`satellite-name`* satellite's operational command from the controller .

```
user@mx1> show version device-list 10.157.80.183
10.157.80.183
--------------------------------------------------------------------
Hostname: vnf0
Model: vSRX
Family: junos-es
Junos: 24.4I-20241120.0.0829
```

### Meaning

The command shows the details of the vSRX Virtual Firewall's Junos OS version. You can run the satellite's other operational commands from the controller.

## Verify the Satellite's Configuration Commands from the Controller

### Purpose

Run the satellite's configuration commands from the controller. Note that JDM is a non-configurable satellite.

### Action

Run the `show version device-list` *`satellite-name`* satellite's operational command from the controller.

```
[edit]
user@mx1# edit chassis satellite 10.157.80.183
```

```
[edit chassis satellite 10.157.80.183]
user@mx1# set security ipsec vpn hub-vpn vpn-monitor
user@mx1# top commit
```

Perform top commit operation to commit the configuration command on the satellite.

### Meaning

The configuration schema for the satellite is available in this here. The command configures IPsec VPN monitoring feature for the hub-vpn.

## Appendix 1: Set Commands on All Devices

Set command output on all devices.

### Set Commands on MX1

```

set groups re0 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-only
set groups re0 interfaces fxp0 unit 0 family inet address 10.52.136.131/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.132/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.131.130/8 master-onlyset system
commit synchronize
set system login user jnuadmin uid 2001
set system login user jnuadmin class super-user
set system login user jnuadmin authentication ssh-rsa "$ABC123c1r0"
set system login user jnuadmin authentication ssh-rsa "$ABC123c1r1"set system services netconf
ssh
set system csds node-instance 0 host-ip 10.157.79.104
set system csds node-instance 0 ip-prefix-range 10.157.80.182-184/19
set system csds node-instance 0 gateway-ip 10.157.64.1
set chassis redundancy graceful-switchover
set chassis jnu-management mode feature-rich
set chassis jnu-management user jnuadmin
set chassis jnu-management other-controller 10.52.136.111
set routing-options nonstop-routing
```

### Set Commands on MX2

```
set groups re0 interfaces fxp0 unit 0 family inet address 10.52.136.111/8 master-only
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.112/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.113/8
set groups re1 interfaces fxp0 unit 0 family inet address 10.52.136.111/8 master-onlyset system
commit synchronize
set system login user jnuadmin uid 2001
set system login user jnuadmin class super-userset system login user jnuadmin authentication ssh-
rsa "$ABC123c2r0”
set system login user jnuadmin authentication ssh-rsa "$ABC123c2r1”
set system services netconf sshset chassis redundancy graceful-switchover
set chassis jnu-management mode feature-rich
set chassis jnu-management user jnuadmin
set chassis jnu-management other-controller 10.52.131.130
set routing-options nonstop-routing
```

## Appendix 2: Show Configuration Output on All Devices

Show command output on all devices.

### Show Command on MX1

```
user@mx1# show chassis jnu-management
mode feature-rich;
satellite 10.157.80.183 {
    model vSRX;
    version 24.4I-20241120.0.0829;
}
satellite 10.157.80.182 {
    model jdm;
    version 24.4-I.20241120.0.0829;
}
user jnuadmin;
other-controller 10.52.131.130;
```

### Show Command on MX2

```


user@mx2# show chassis jnu-management
mode feature-rich;
satellite 10.157.80.183 {
    model vSRX;
    version 24.4-I.20241120.0.0829;
}
satellite 10.157.80.182 {
    model jdm;
    version 24.4-I.20241120.0.0829;
}
user jnuadmin;
other-controller 10.52.136.111;
```

## Appendix 3: Commands to Delete JNU Satellites

Ensure that you know the CSDS instance ID for the vSRX Series Virtual Firewall and the JDM satellites that you plan to delete.

To delete all the satellites, first remove the vSRX Series Virtual Firewalls, followed by the JDM. You can delete an existing vSRX Series Virtual Firewall and add a new one without deleting JDM. Note that you must delete all firewalls running on the host server before you can delete the JDM.

To delete the satellites, run the following steps:

1. Delete vSRX Series Virtual Firewall.

   ```
   user@mx1> request csds delete-vsrx csds-instance-id 0
   ```

2. Delete JDM.

```
user@mx1> request csds jdm delete csds-instance-id 0
```

## SEE ALSO

Understand vSRX Orchestration with JDM for CSDS | **65**