

# Junos OS

---

## Multinode High Availability

Published  
2022-09-20



Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

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

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

*Junos OS Multinode High Availability*

Copyright © 2022 Juniper Networks, Inc. All rights reserved.

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

## YEAR 2000 NOTICE

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

## END USER LICENSE AGREEMENT

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



# Table of Contents

About This Guide | vi

1

## Overview

Multinode High Availability | 2

Overview | 2

How Multinode High Availability Works | 14

Multinode High Availability Monitoring | 30

Prepare Your Environment for Multinode High Availability Deployment | 40

Multinode High Availability Services | 43

2

## Multinode High Availability Configuration

Example: Configure Multinode High Availability in a Layer 3 Network | 48

Overview | 48

Requirements | 48

Topology | 50

Configuration | 52

Verification | 79

Example: Configure Multinode High Availability in a Default Gateway Deployment | 93

Overview | 93

Requirements | 93

Topology | 95

Configuration | 97

Verification | 122

Example: Configure Multinode High Availability in a Hybrid Deployment | 134

Overview | 135

Requirements | 135



Topology | 136

Configuration | 138

Verification | 164

3

## Hardware and Software Upgrades

Software Upgrade in Multinode High Availability | 179

Insert Additional SRX5K-SPC3 in a Multinode High Availability Setup | 198

Insert SRX5K-SPC3 in a Multinode High Availability Setup | 199

4

## Multinode High Availability Support for vSRX

Multinode High Availability Support for vSRX Instances | 203

Overview | 203

Multinode High Availability in AWS | 203

Configuring Multinode High Availability In Amazon Web Services (AWS) Deployment | 206

5

## Configuration Statements

activeness-probe | 230

hardware-upgrade | 232

high-availability (Chassis) | 233

high-availability (security cloud) | 236

liveness-detection (high availability) | 238

local-id | 241

peer-id | 243

monitor (Multinode High Availability) | 245

services-redundancy-group | 247

software-upgrade | 252

traceoptions | 253

virtual-ip | 256

6

## Operational Commands



clear chassis high-availability data-plane statistics | 261

clear chassis high-availability information | 262

clear security pki node-local certificate-request | 264

clear security pki node-local local-certificate | 266

clear security pki node-local key-pair | 268

show chassis high-availability data-plane statistics | 269

show chassis high-availability information | 275

request chassis high-availability failover services-redundancy-group | 281

show chassis high-availability peer-info | 283

show chassis high-availability services-redundancy-group | 285

show security pki node-local local-certificate | 290

show security pki node-local certificate-request | 296

request security pki node-local local-certificate verify | 300

request security pki node-local local-certificate re-enroll | 302

request security pki node-local local-certificate load | 304

request security pki node-local local-certificate export | 306

request security pki node-local local-certificate enroll | 308

request security pki node-local key-pair export | 312

request security pki node-local generate-key-pair | 314

request security pki node-local generate-certificate-request | 316



# About This Guide

Use this guide to learn about Multinode High Availability support on SRX Series devices to address high availability requirements for modern data centers. In this guide, you'll learn how to configure the solution in simple and reliable deployment models.



# 1

CHAPTER

## Overview

---

Multinode High Availability | 2

Prepare Your Environment for Multinode High Availability Deployment | 40

Multinode High Availability Services | 43

---



# Multinode High Availability

## SUMMARY

Learn about the Multinode High Availability solution and how you can use it in simple and reliable deployment models. Currently, we support two nodes in any Multinode High Availability deployment.

## IN THIS SECTION

- [Overview | 2](#)
- [How Multinode High Availability Works | 14](#)
- [Multinode High Availability Monitoring | 30](#)

## Overview

Business continuity is an important requirement of the modern network. Downtime of even a few seconds might cause disruption and inconvenience apart from affecting the OpEx and CapEx. Modern networks also have data centers spread across multiple geographical areas. In such scenarios, achieving high availability can be very challenging.

Juniper Networks® SRX Series Firewalls support a new solution, Multinode High Availability, to address high availability requirements for modern data centers. In this solution, both the control plane and the data plane of the participating devices (nodes) are active at the same time. Thus, the solution provides interchassis resiliency.

The participating devices could be co-located or physically separated across geographical areas or other locations such as different rooms or buildings. Having nodes with high availability across geographical locations ensures resilient service. If a disaster affects one physical location, Multinode High Availability can fail over to a node in another physical location, thereby ensuring continuity.

## Benefits of Multinode High Availability

- **Reduced CapEx and OpEx**—Eliminates the need for a switched network surrounding the firewall complex and the need for a direct L2 connectivity between nodes
- **Network flexibility**—Provides greater network flexibility by supporting high availability across Layer 3 (L3) and switched network segments.
- **Stateful resilient solution**—Supports active control plane and data plane at the same time on both nodes.



- **Business continuity and disaster recovery**—Maximizes availability, increasing redundancy within and across data centers and geographies.
- **Smooth upgrades**—Supports different versions of Junos OS on two nodes to ensure smooth upgrades between the Junos OS releases.

## Multinode High Availability Support

We support Multinode High Availability on:

- SRX5800, SRX5600, SRX5400 with SPC3, IOC3, SCB3, SCB4, and RE3 (in Junos OS Release 20.4R1)
- SRX4600, SRX4200, SRX4100, and SRX1500 (in Junos OS Release 22.3R1)
- vSRX3.0 virtual firewalls (in Junos OS Release 22.3R1) for the following private and public cloud platforms:
  - KVM (kernel-based virtual machine)
  - VMWare ESXi
  - Amazon Web Services (AWS)

We support two nodes in Multinode High Availability solution in Junos OS Release 22.3R1.

Multinode High Availability does not support transparent mode high availability (HA)

## Supported Features

SRX Series devices with Multinode High Availability support the firewall and advanced security services—such as application security, unified threat management (UTM), intrusion prevention system (IPS), firewall user authentication, NAT, ALG.

For the complete list of features supported with Multinode High Availability, see [Feature Explorer](#).

## Deployment Scenarios

**NOTE:** In Junos OS Release 22.3R1, we support a two-node configuration for the Multinode High Availability solution.

Multinode High Availability supports two SRX Series devices presenting themselves as independent nodes to the rest of the network. The nodes are connected to adjacent infrastructure belonging to

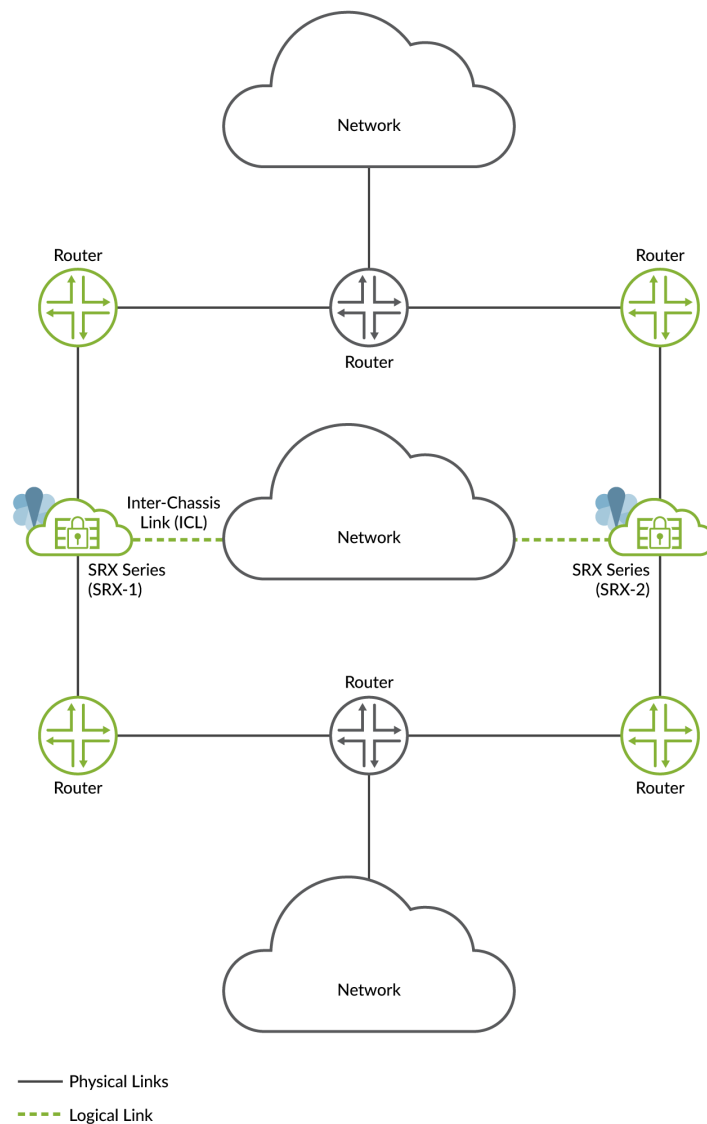


different networks. These nodes can either be collocated or separated across geographies. Participating nodes back up each other to ensure a fast synchronized failover in case of system or hardware failure.

We support the following types of network deployment models for Multinode High Availability:

- Layer 3 network mode ( fully routed environments where routers connected at both ends)

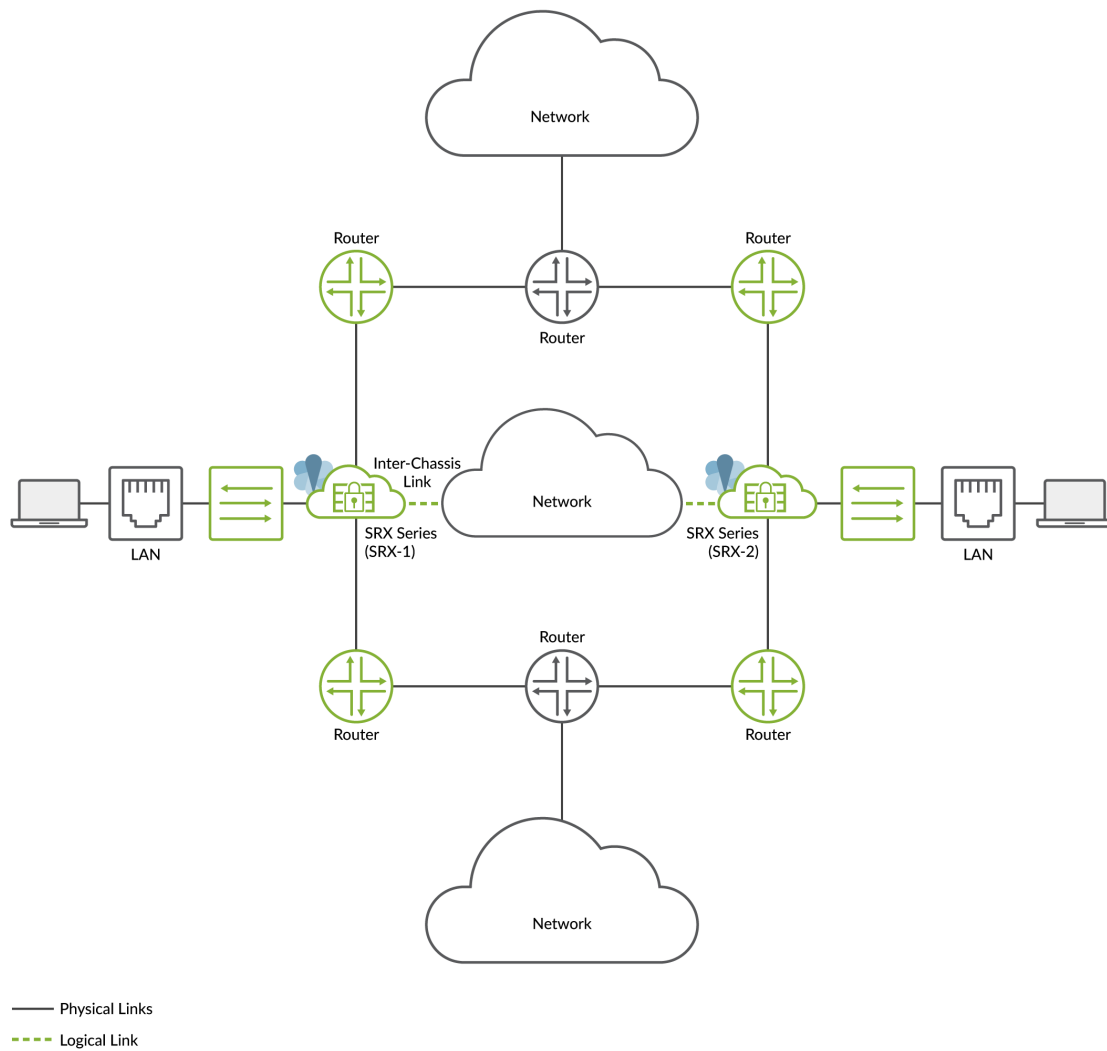
**Figure 1: Layer 3 Mode**





- Default gateway mode ( Layer 2 switches connected at both ends). Common deployment of DMZ networks where the firewall devices act as the default gateway for the hosts and applications on the same segment

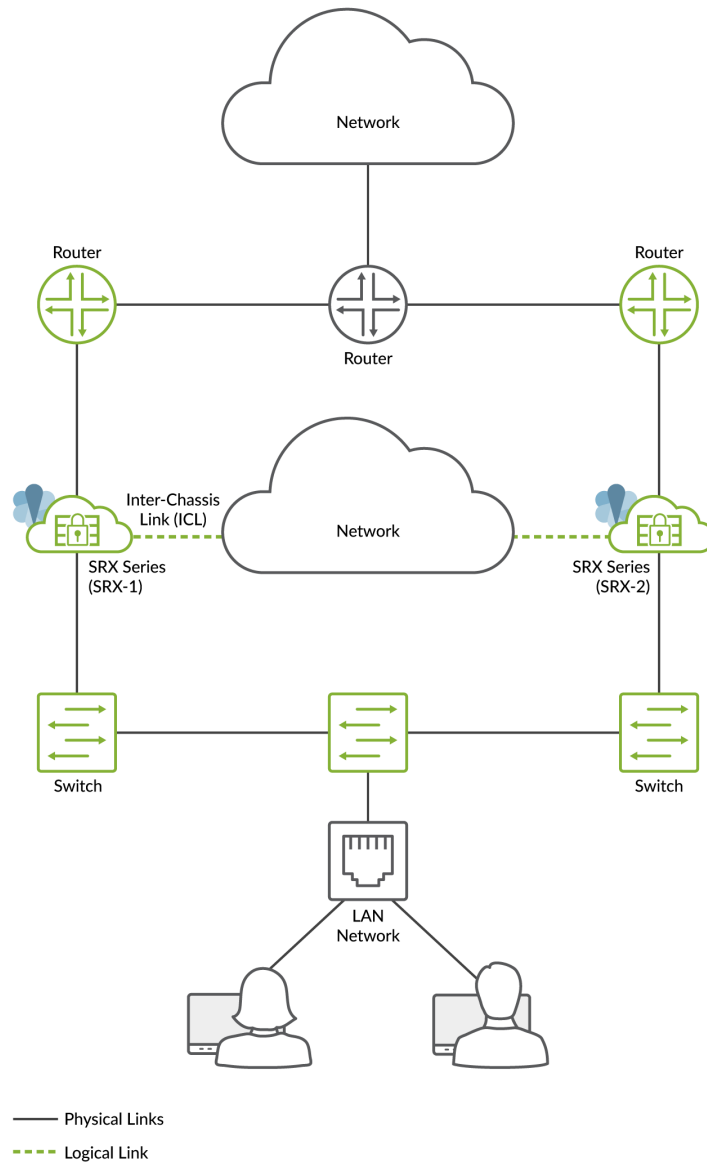
**Figure 2: Default Gateway Mode**





- Hybrid network mode (mixed mode of routed networks on one side and locally connected networks on the other side)

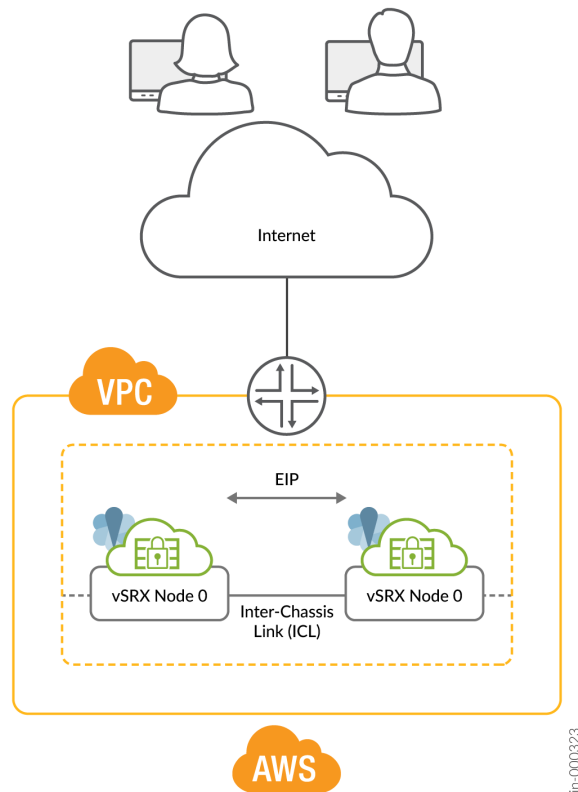
Figure 3: Hybrid Mode





- Public cloud deployment (for example, AWS)

**Figure 4: Public Cloud Deployment**



### How Is Multinode High Availability Different from Chassis Cluster?

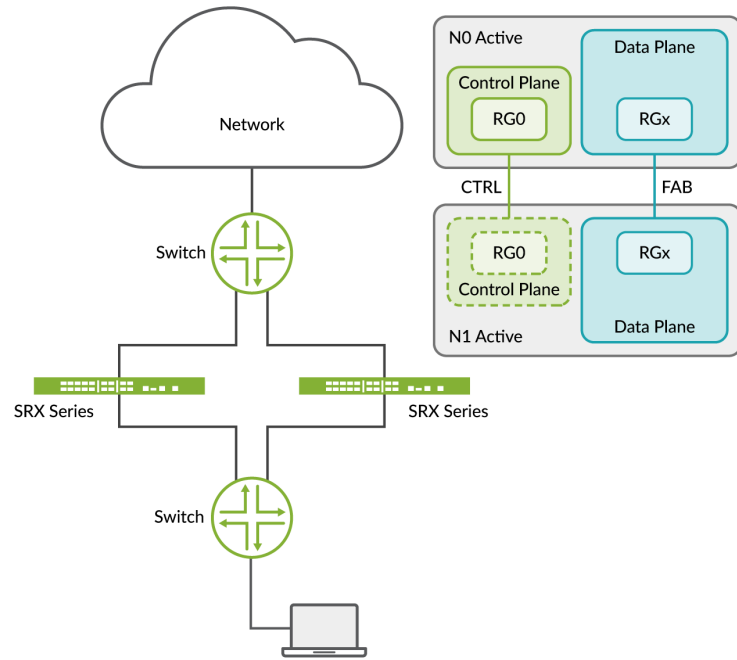
A chassis cluster operates in Layer 2 network environment and requires two links—the control link and the fabric link). These links connect both nodes over dedicated VLANs using back-to-back cabling or over dark (unlit) fiber connections. Control links and fabric links use dedicated physical ports on the SRX Series device.

Multinode High Availability uses an encrypted logical interchassis link (ICL). The ICL connects the nodes over a routed network instead of a dedicated Layer 2 network. You can use revenue ports on the SRX Series devices to setup ICL connection and configure a routing instance for the ICL path to ensure maximum segmentation.



Figure 5 on page 8 and Figure 6 on page 9 show two architectures.

**Figure 5: Chassis Cluster Topology in a Layer 2 Network**



jn-000324



Figure 6: Multinode High Availability in a Layer 3 Network

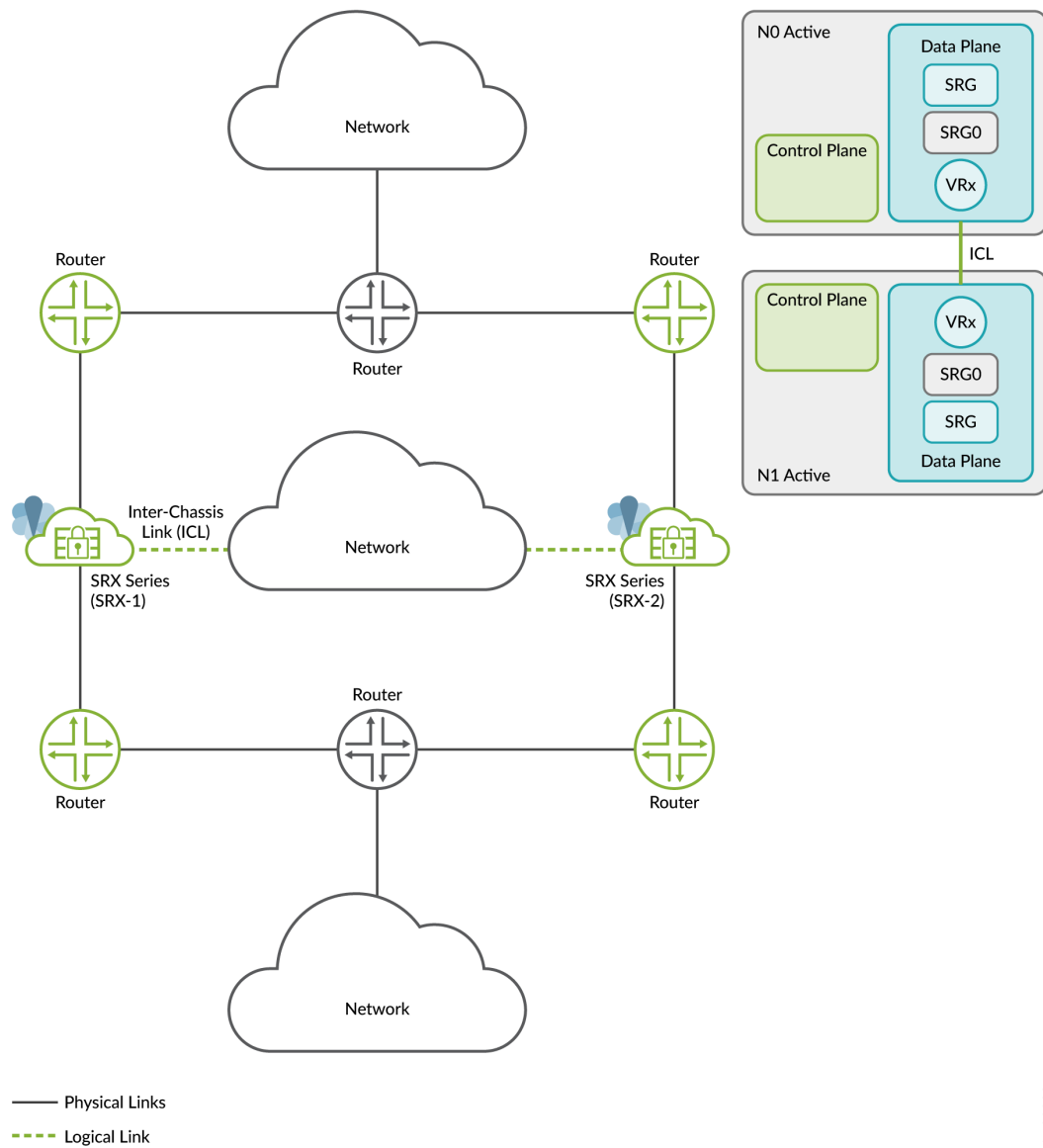


Table 1 on page 10 lists the differences between the two architectures



**Table 1: Comparing Chassis Cluster and Multinode High Availability**

Parameters	Chassis Cluster	Multinode High Availability
Network topology	Nodes connect to a broadcast domain and move the IP address during failover by sending Gratuitous Address Resolution Protocol (GARP) messages to the switch.	Nodes connect to a router, a broadcast domain, or a combination of both.
Network environment	Layer 2	Layer 3, Layer 2, a combination of Layer 3 and Layer 2 (hybrid mode), and public cloud (AWS) deployments
Traffic switchover approach	Switchover using Layer 2 GARP from an SRX Series device to a peer Layer 2 switch	Switchover using IP path selection by a peer Layer 3 router or Layer 2 GARP from an SRX Series device to a peer Layer 2 switch
Public cloud	Not supported	Supported
Dynamic routing function	Active only on an SRX Series device in the primary RG0 state	Active on each SRX Series device
Deployment	Requires a dedicated L2 stretch between nodes to offer geo-redundancy	Offers geo-redundancy without any switched/broadcast domain
Connection between SRX Series devices	Control link and fabric link (cables)	ICL (Layer 3-capable, IP-based link)
IP monitoring to detect network failure	Supports IPv4 traffic	Supports both IPv4 and IPv6 traffic

## Multinode High Availability Glossary

Let's begin by getting familiar with Multinode High Availability terms used in this documentation.



Table 2: Multinode High Availability Glossary

Term	Description
active/backup state	Active/backup state is available when you have the SRG1 in active state on one node and in backup state on the other node.
active node	First node that is active in the high availability deployment. The active node accepts connections and manages the session.
backup node	Node that takes over as the new active node if the active node goes down due to any reason.
device priority	Priority value determines whether a node can act as the active node in a Multinode High Availability setup. The node with a lower numerical value has a higher priority and, therefore, acts as the active node while the other node acts as the backup node.
device preemption	Preemptive behavior allows the device with the higher priority (lower numerical value) to resume as active node after it recovers from a failure. If you need to use a specific device in Multinode High Availability as active node, then you must enable the preemptive behavior on both the devices and assign a device priority value for each device.
failover	Event where the backup node in a high availability system takes over the task of the active node when the active node fails. In contrast, failure can occur when a physical link goes down or an ICMP probe fails.
floating IP address or activeness probing IP address	An IP address that moves from an active node to the backup node during failover in a Multinode High Availability setup. This mechanism enables clients to communicate with the nodes using a single IP address. You configure the floating IP address on the interface that connects to participating networks or segments.
high availability/resiliency	Ability of a system to eliminate single points of failure to ensure continuous operations over an extended period of time.



Table 2: Multinode High Availability Glossary *(Continued)*

Term	Description
interchassis link (ICL)	<p>IP-based link (logical link) that connects nodes over a routed network in a Multinode High Availability deployment. The security device uses the ICL to synchronize and maintain state information and to handle device failover scenarios.</p> <p>You can use an ICL to connect the nodes directly. Alternatively, you can use a switch or a set of routers to connect nodes (for geo-redundant deployments). Because this is an IP-based link, you must be able to route local and peer IP addresses in the network.</p>
link encryption	<p>Link encryption provides data privacy for messages traversing over the network. In Multinode High Availability, packets sent over an ICL may traverse a path on the public IP network. So, we secure ICL using IPsec VPN.</p>
monitoring (BFD)	<p>Monitoring of one or more links using Bidirectional Forwarding Detection (BFD). BFD monitoring triggers a routing path change or a system failover, depending on system configuration.</p>
monitoring (IP)	<p>Monitoring of a reliable IP address and system state in case of loss of communication with the peer node.</p>
monitoring (path)	<p>Method that uses ICMP to verify the reachability of the IP address. The default interval for ICMP ping probes is 1 second.</p>
monitoring (system)	<p>Monitoring of key hardware and software resources and infrastructures by triggering failover when a failure is detected on a node.</p>
probing	<p>Mechanism used to exchange messages between active and backup nodes in the high availability setup. The messages determine the status and health of the application on each individual node.</p>



Table 2: Multinode High Availability Glossary (*Continued*)

Term	Description
real-time object (RTO)	Special payload packet that contains the necessary information to synchronize the data from one node to the other node.
split-brain detection (also known as control plane detection or activeness conflict detection)	Event where the ICL between two Multinode High Availability nodes is down, and both nodes initiate an activeness determination probe (split-brain probe). Based on the response to the probe, subsequent failover to a new role is triggered
services redundancy group (SRG)	Failover unit that includes and manages a collection of objects on the participating nodes. The SRG on one node switches over to the other node when a failover is detected.
SRG0	Manages all control plane stateless services such as firewall, NAT, and ALG. SRG0 is active on all participating nodes and handles symmetric security flows.
SRG1	Manages control plane stateful service (IPsec VPN).
synchronization	Process where controls and data plane states are synchronized across the nodes.
virtual IP (VIP) address	(For hybrid and default gateway deployments). Virtual IP address used for activeness determination and enforcement on the switching side in a Multinode High Availability setup.
virtual MAC (VMAC) address	(For hybrid and default gateway deployments). Virtual MAC address dynamically assigned to the interface on active node that faces the switching side.

Now we are that familiar with Multinode High Availability features and terminology, let's proceed to understand how Multinode High Availability works.



## How Multinode High Availability Works

### IN THIS SECTION

- [Services Redundancy Groups | 18](#)
- [Activeness Determination and Enforcement | 19](#)
- [Failover and Resiliency | 23](#)
- [Interchassis Link \(ICL\) Encryption | 23](#)
- [Split-Brain Detection and Prevention | 26](#)

**NOTE:** As of Junos OS Release 22.3R1, we support a two-node configuration for the Multinode High Availability solution.

In a Multinode High Availability setup, you connect two SRX Series devices to adjacent upstream and downstream routers (for Layer 3 deployments), routers and switches (hybrid deployment), or switches (default gateway deployment) using Gigabit Ethernet ports.

The nodes communicate with each other using an interchassis link (ICL) connected over a routed network or connected directly. You can establish the ICL by using:

- Indirect connection through a Layer 3 network: Establish a logical IP link connecting both nodes using a loopback (lo0) interface or an aggregated Ethernet interface (ae0) (nodes located across different locations).
- Direct connection: Connect two ports on each node directly using a crossover cable (node co-located).

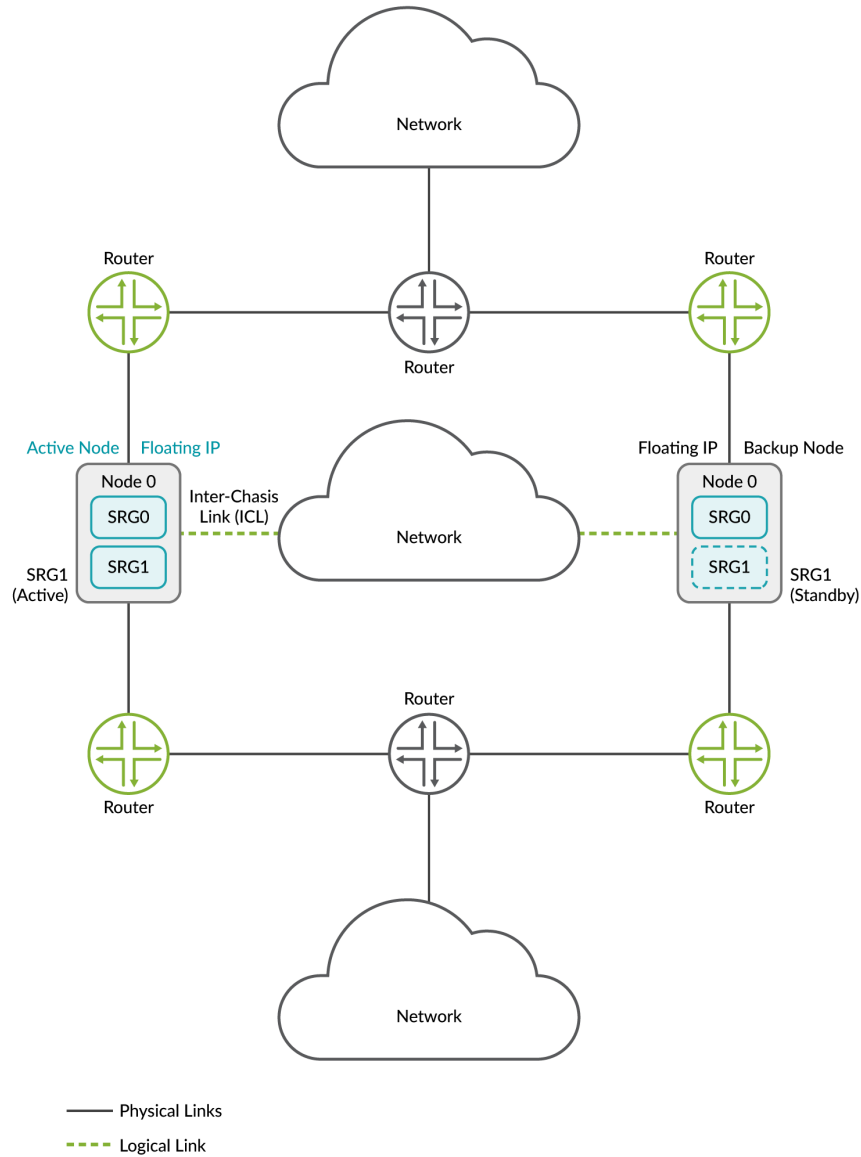
Multinode High Availability operates in active/active mode for data plane and active/backup mode for control plane services. The active SRX Series device hosts the floating IP address and steers traffic towards it using the floating IP address.

During a failover, the floating IP address moves from the old active node to the new active node. This mechanism enables clients to communicate with the nodes using a single IP address. You configure the floating IP address on the interface that connects to participating networks or segments.



Figure 7 on page 15, Figure 8 on page 16, and Figure 9 on page 17 show deployments in Layer 3, hybrid, and default gateway modes.

**Figure 7: Layer 3 Deployment**

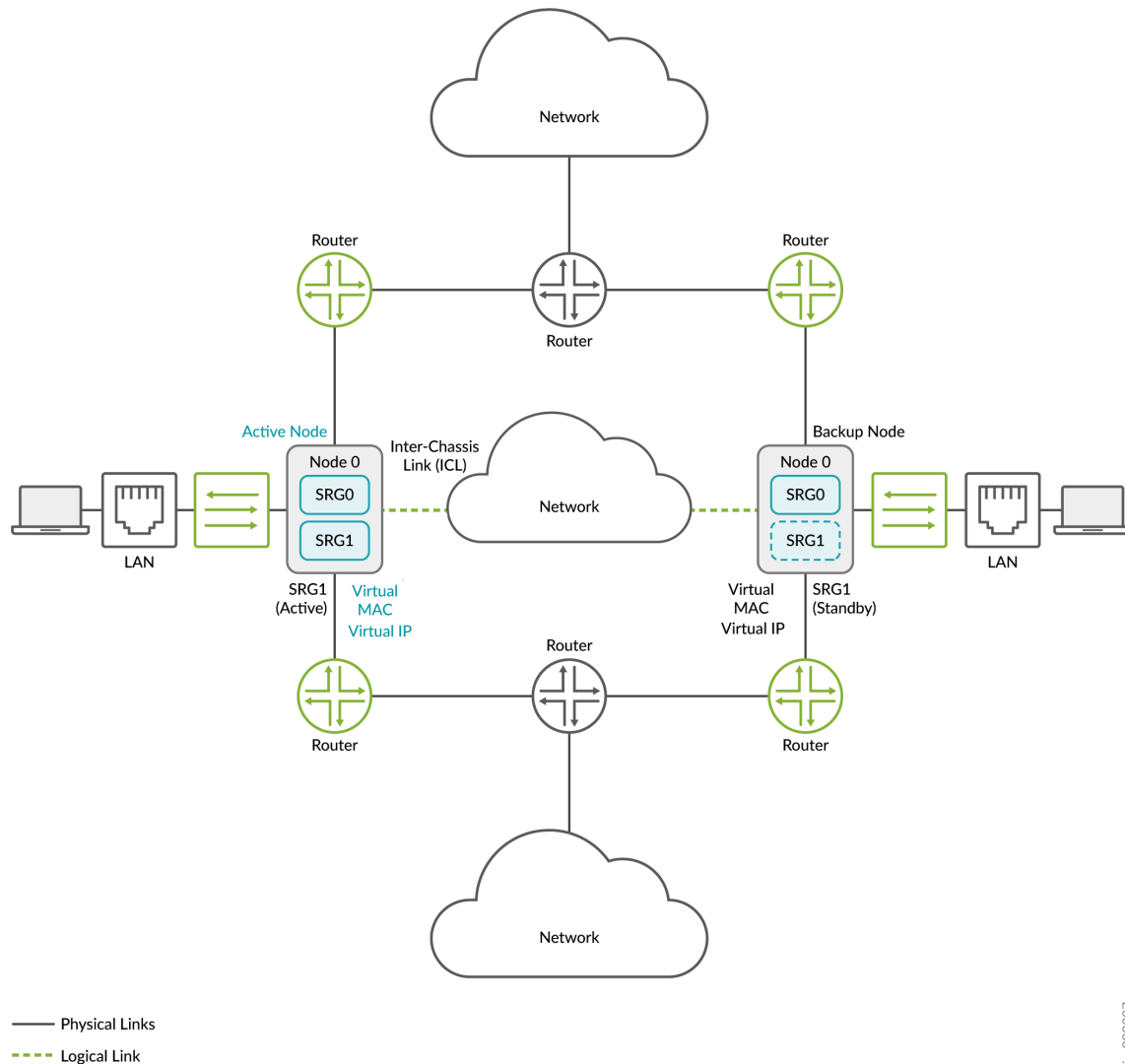


In this topology, two SRX Series devices are part of a Multinode High Availability setup. The setup has Layer 3 connectivity between SRX Series devices and neighboring routers. The devices are running on



separate physical Layer 3 networks and are operating as two independent nodes. The nodes shown in the illustration are co-located in the topology. The nodes can also be geographically separated.

**Figure 8: Default Gateway Deployment**



In a typical default gateway deployment, hosts and servers in a LAN are configured with a default route next-hop IP address to the security device. So the security device must host a virtual IP (VIP) address that moves between nodes based on the activeness. The configuration on hosts remains static, and security device failover is seamless from the hosts' perspective.

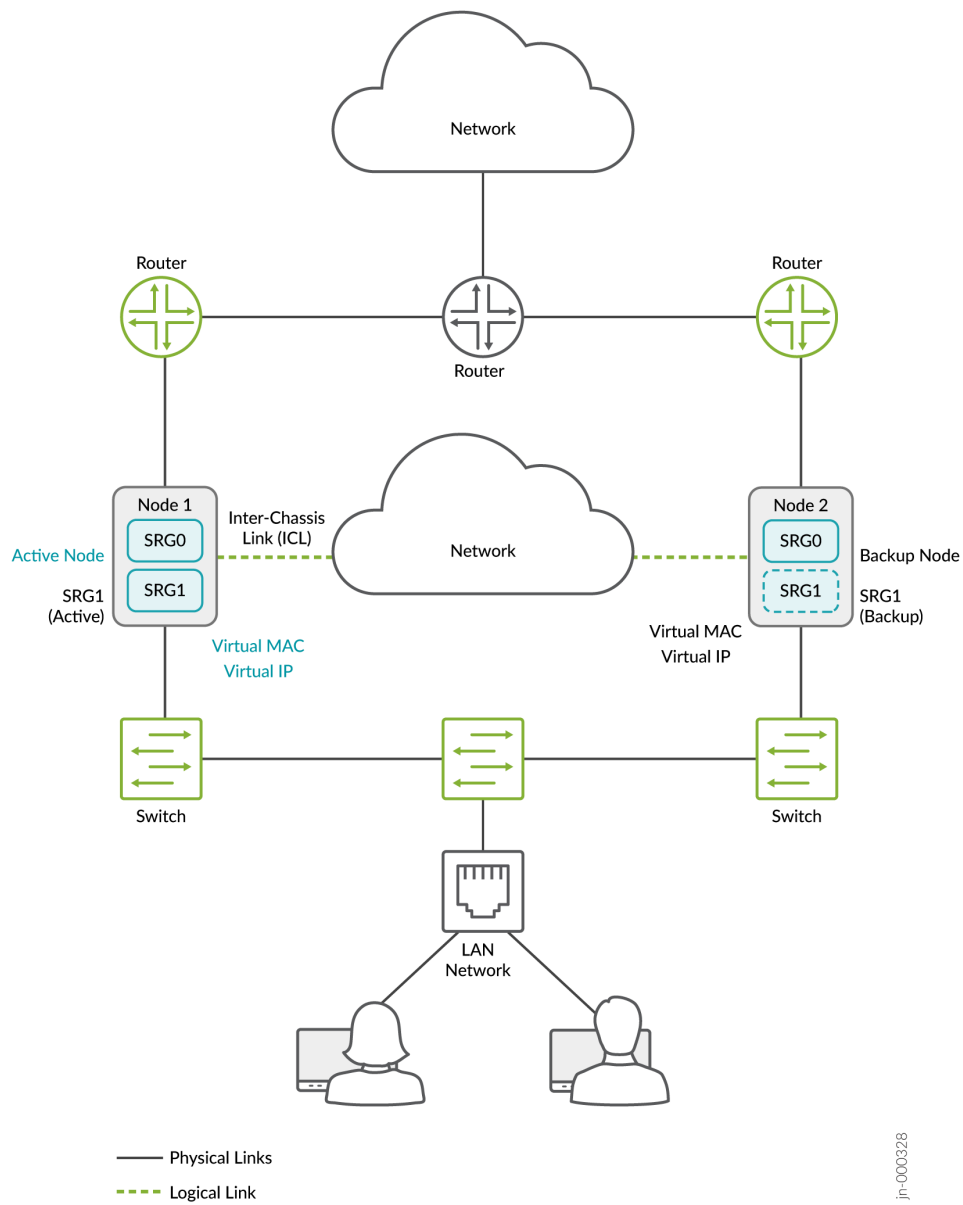
On the connected routers, you can either use dynamic routing or add the static route for the termination IP address with the next hop as the VIP address. You can also optionally configure static



Address Resolution Protocol (ARP) for the VIP address using the VMAC address to ensure no change in the IP address during the failover.

You must create the static route on SRX Series devices for the routers or hosts beyond the switches in both the directions.

**Figure 9: Hybrid Deployment**





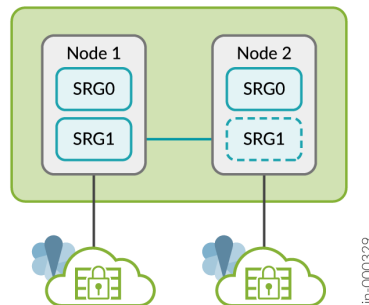
In hybrid mode, an SRX Series device uses a VIP address on the Layer 2 side to draw traffic toward it. On the Layer 3 side, routers can employ dynamic routing or use static route for hosts with next hop as the VIP address. You can also optionally configure the static ARP for the VIP using the VMAC to ensure no change in the IP address during the failover.

Let's now understand the components and functionality of Multinode High Availability in detail.

## Services Redundancy Groups

A services redundancy group (SRG) is a failover unit in a Multinode High Availability setup. [Figure 10 on page 18](#) shows SRG0 and SRG1 in a Multinode High Availability setup.

**Figure 10: SRG Support in Multinode High Availability**



- SRG0 - Manages control plane stateless services. SRG0 remains in the active state on both active node and backup node.
- SRG1 - Manages control plane stateful service (IPsec VPN). The node on which SRG1 remains active is active node and the node where SRG is on standby state is the backup node.

[Table 3 on page 19](#) explains the behavior of SRGs in a Multinode High Availability setup.



Table 3: Services Redundancy Group Details in Multinode High Availability

Related Services Redundancy Group (SRG)	Operates in	Affected Services	Synchronization Type	When Active Node Fails
SRG0	Active/active data plane	Manages control plane stateless services—services such as firewall, NAT, and ALG (services without control plane state). Stateless services operate in active/active mode and forwards traffic on both nodes	Synchronization for only data plane states across the nodes.	Data plane must fail over to the backup node. Activeness is maintained per session. The backup session remains in standby (or warm) state and moves to active state when it receives the first packet. The new active node notifies the other node about the session move. Other node changes the session state from active to standby.
SRG1	Active/backup control and data plane	Manages control plane stateful service (IPsec VPN). On the backup SRG, the real-time objects (RTOs) are synchronized.	Synchronization for both control plane and data plane states across the nodes.	Both control plane and data plane need to fail over to the backup node at the same time.

## Activeness Determination and Enforcement

In a Multinode High Availability setup, activeness is determined at the service level, not at the node level. That is, the active/backup state is at the SRG level and the traffic is steered toward the active SRG. SRG0 remains active on both the nodes, whereas SRG1 can remain in active or in backup state in each node. The node where SRG1 is in active state is considered as active node.

If you prefer a certain node to take over as the active node on boot, you can do one of the followings:

- Configure the upstream routers to include preferences for the path where the node is located.

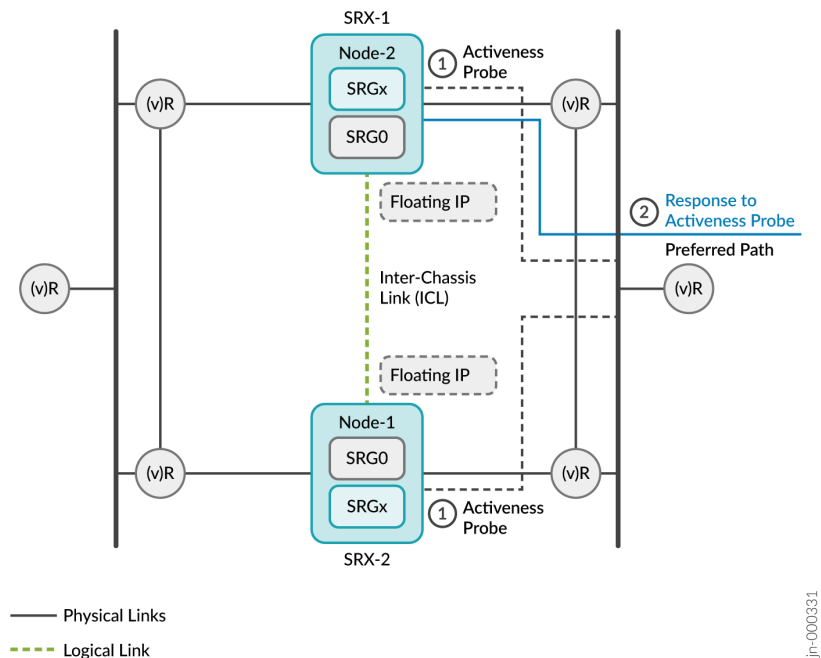


- Configure activeness priority.
- Allow the node with higher node ID (in case the above two options not configured) to take the active role.

In a Multinode High Availability setup, both the SRX Series devices initially advertise the route for the floating IP address to the upstream routers. There isn't a specific preference between the two paths advertised by SRX Series devices. However, the router can have its own preferences on one of the paths depending on the configured metrics.

Figure 11 on page 20 represents the sequence of events for activeness determination and activeness enforcement.

**Figure 11: Activeness Determination and Enforcement**

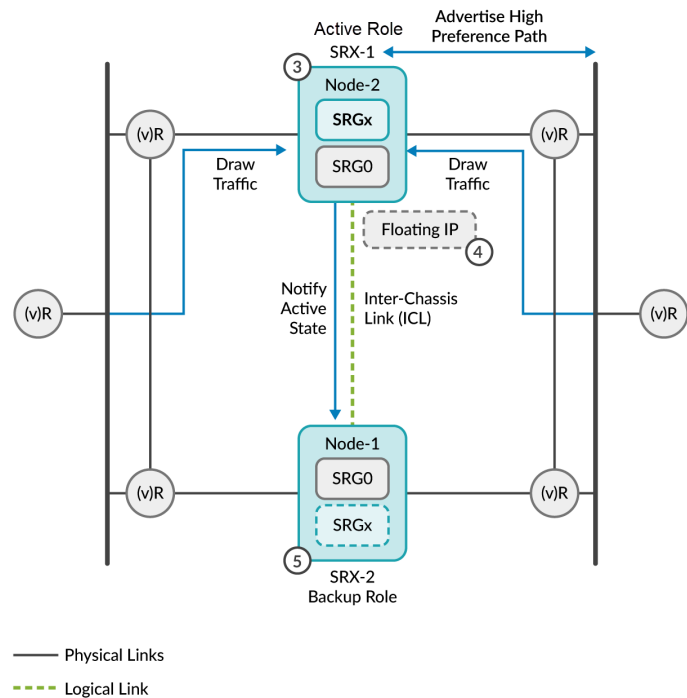


1. On boot, devices enter the hold state and start probing continuously. The devices use the floating IP address (activeness-probing source IP address) as the source IP address and IP addresses of the upstream routers as the destination IP address for the activeness determination probe.



2. The router hosting the probe destination IP address replies to the SRX Series device that is available on its preferred routing path. In the following example, SRX-1 gets the response from the upstream router.

**Figure 12: Activeness Determination and Enforcement**



3. SRX-1 promotes itself to the active role since it got the probe reply. SRX-1 communicates its role change to the other device and takes up the active role.
4. After the activeness is determined, the active node (SRX-1):
  - Hosts the floating IP address assigned to it.
  - Advertises the high-preference path to adjacent BGP neighbors.
  - Continues to advertise the active (higher) preference path for all remote and local routes to draw the traffic.
  - Notifies the active node status to the other node through the ICL.
5. The other device (SRX-2) stops probing and takes over the backup role. The backup node advertises the default (lower) priority, ensuring that the upstream routers do not forward any packets to the backup node.



The Multinode High Availability module adds active and backup signal routes for the SRG to the routing table when the node moves to the active role. In case of node failures, the ICL goes down and the current active node releases its active role and removes the active signal route. Now the backup node detects the condition through its probes and transitions to the active role. The route preference is swapped to drive all the traffic towards the new active node.

The switch in the route preference advertisement is part of routing policies configured on SRX Series devices. You must configure the routing policy to include the active signal route with the `if-route-exists` condition.

## For Default Gateway Deployments

If both the nodes are booting up at the same time, then the Multinode High Availability system uses the configured priority value of an SRG to determine activeness. Activeness enforcement takes place when the node with an active SRG owns the virtual IP (VIP) address and the virtual MAC (VMAC) address. This action triggers Gratuitous ARP (GARP) toward the switches on both sides and results in updating the MAC tables on the switches.

## For Hybrid Deployments

Activeness enforcement takes place on Layer 3 (router side) and Layer 2 (switch side). On the Layer 2 side, the SRX Series device enforces activeness by owning the VIP and VMAC addresses while triggering GARP. On the Layer 3 side, you enforce activeness by configuring signal route and triggering corresponding route advertisements.

When the failover happens and the old backup node transitions to the active role, the route preference is swapped to drive all the traffic to the new active node.

## Activeness Priority and Preemption

Configure the activeness priority (1-254) for SRG1 and enable the preemptive behavior on both the nodes. The preempt option ensures that the traffic always falls back to the specified node, when the node recovers from a failover.

You can configure activeness priority and preemption for an SRG1 as in the following sample:

```
[edit]
user@host# show chassis high-availability
services-redundancy-group 1 {
    preemption;
```



```
activeness-priority 200;
}
```

See ["Configuring Multinode High Availability In a Layer 3 Network" on page 48](#) for the complete configuration example.

As long as the nodes can communicate with each other through the ICL, the priority is honored.

## Failover and Resiliency

The Multinode High Availability solution supports redundancy at the service level. Service-level redundancy minimizes the effort needed to synchronize the control plane state across the node.

After the Multinode High Availability setup determines activeness, it negotiates subsequent high availability (HA) state through the ICL. The backup node sends ICMP probes using the floating IP address. If the ICL is up, the node gets the response to its probe and remains as the backup node. If the ICL is down, and there are no probe response, the backup node transitions into the active node.

The SRG1 of the previous backup node now transitions to the active state and continues to operate seamlessly. When the transition happens, the floating IP address is assigned to the active SRG1. In this way, the IP address floats between the active and backup nodes and remains reachable to all the connected hosts. Thus, traffic continues to flow without any disruption.

Services, such as IPsec VPN, that require both control plane and data plane states are synchronized across the nodes. Whenever an active node fails for this service function, both control plane and data plane fail over to the backup node at the same time.

The nodes use the following messages to synchronize data:

- Routing Engine to Routing Engine control application messages
- Routing Engine configuration-related messages
- Data plane RTO messages

## Interchassis Link (ICL) Encryption

In Multinode High Availability, active and backup nodes are connected by an ICL using IP addresses that are routable in the network.

You can establish an ICL with a logical IP link connecting both nodes using a loopback (lo0) interface or an aggregated Ethernet interface (ae0) for the nodes located across different locations or connect two ports on each node directly using a crossover cable in case the nodes are in same location.



Nodes use the ICL to synchronize control plane and data plane states between them. Communication goes over the network including upstream and downstream routers. Packets sent over the ICL may traverse a path that is not always trusted. Therefore, you must secure the packets traversing the ICL.

In Multinode High Availability, you must also separate the transit traffic in revenue interfaces from the high availability (HA) traffic.

For these reasons, you must encrypt the traffic traversing the ICL using IPsec standards. IPsec protects traffic by establishing an encryption tunnel for the ICL. When you apply HA link encryption, the HA traffic flows between the nodes only through the secure, encrypted tunnel. Without HA link encryption, communication between the nodes may not be secure.

To encrypt the HA link for the ICL:

- Install the Junos IKE package on your SRX Series device by using the following command:  

```
request system software add optional://junos-ike.tgz.
```
- Configure a VPN profile for the HA traffic and apply the profile for both the nodes. The IPsec tunnel negotiated between the SRX Series devices uses the IKEv2 protocol.
- Ensure you have included the statement `ha-link-encryption` in your IPsec VPN configuration.  
Example: `user@host# set security ipsec vpn vpn-name ha-link-encryption.`

For ICL set up, we recommend

- Use ports and network which is less likely to be saturated
- Not to use the dedicated HA ports (control and fabric ports, if available on your SRX Series device)

See ["Configuring Multinode High Availability" on page 48](#) for more details.

## PKI-Based Link Encryption for ICL

Starting in Junos OS Release 22.3R1, we support PKI-based link encryption for interchassis link (ICL) in Multinode High Availability. As a part of this support, you can now generate and store node-specific PKI objects such as local keypairs, local certificates, and certificate-signing requests on both nodes. The objects are specific to local nodes and are stored in the specific locations on both nodes.

The node local objects enable you to distinguish between PKI objects that are used for ICL encryption and PKI objects used for IPsec VPN tunnel created between two endpoints.

You can use the following commands run on local node to work with node-specific PKI objects.

Generating a private/public key pair for a local node

["request security pki node-local generate-key-pair" on page 314](#)



Generating and enrolling a local digital certificate in a local node	<ul style="list-style-type: none"> <li>• <a href="#">"request security pki node-local generate-certificate-request" on page 316</a></li> <li>• <a href="#">"request security pki node-local key-pair export" on page 312</a></li> <li>• <a href="#">"request security pki node-local local-certificate verify" on page 300</a></li> <li>• <a href="#">"request security pki node-local local-certificate re-enroll" on page 302</a></li> <li>• <a href="#">"request security pki node-local local-certificate load" on page 304</a></li> <li>• <a href="#">request security pki node-local local-certificate export</a></li> <li>• <a href="#">"request security pki node-local local-certificate enroll" on page 308</a></li> </ul>
Clear node-specific certificates	<ul style="list-style-type: none"> <li>• <a href="#">"clear security pki node-local certificate-request" on page 264</a></li> <li>• <a href="#">"clear security pki node-local local-certificate" on page 266</a></li> <li>• <a href="#">"clear security pki node-local key-pair" on page 268</a></li> </ul>
Display node-specific local certificates and certificate requests.	<ul style="list-style-type: none"> <li>• <a href="#">"show security pki node-local local-certificate" on page 290</a></li> <li>• <a href="#">"show security pki node-local certificate-request" on page 296</a></li> </ul>

On your security device in Multinode High Availability, if you've configured the automatic re-enrollment option and if the ICL goes down at the time of re-enrollment trigger, both the devices start enrolling the same certificate separately with the CA server and download the same CRL file. Once Multinode High Availability re-establishes the ICL, the setup uses only one local certificate. You must synchronize the certificates from the active node to backup node using the `user@host> request security pki sync-from-peer` command on the backup node.

If you don't synchronize the certificates, the certificate mismatch issue between peer nodes persists till the next re-enrollment.



Optionally you can enable TPM (Trusted Platform module) on both nodes before generating any keypairs on the nodes. See [Using Trusted Platform Module to Bind Secrets on SRX Series devices](#).

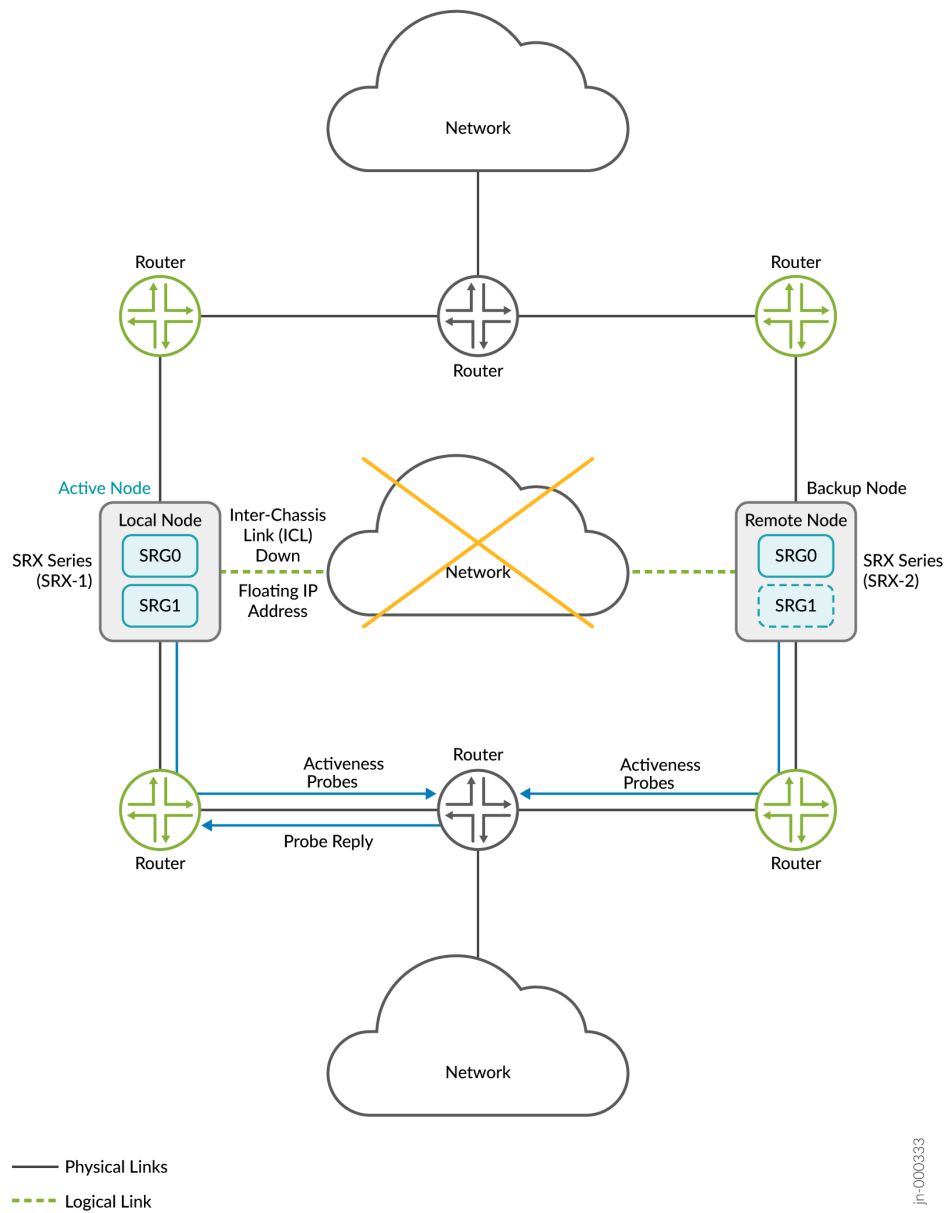
## Split-Brain Detection and Prevention

Split-brain detection or activeness conflict happens when the ICL between two Multinode High Availability nodes is down and both nodes cannot reach each other to gather the status of peer node anymore.

Consider a scenario where two SRX Series devices are part of Multinode High Availability setup. Lets consider SRX-1 as a local node and SRX-2 a remote node. The local node is currently in active role and the upstream router has higher priority path for the local node.

### Case 1: Active Node is Up



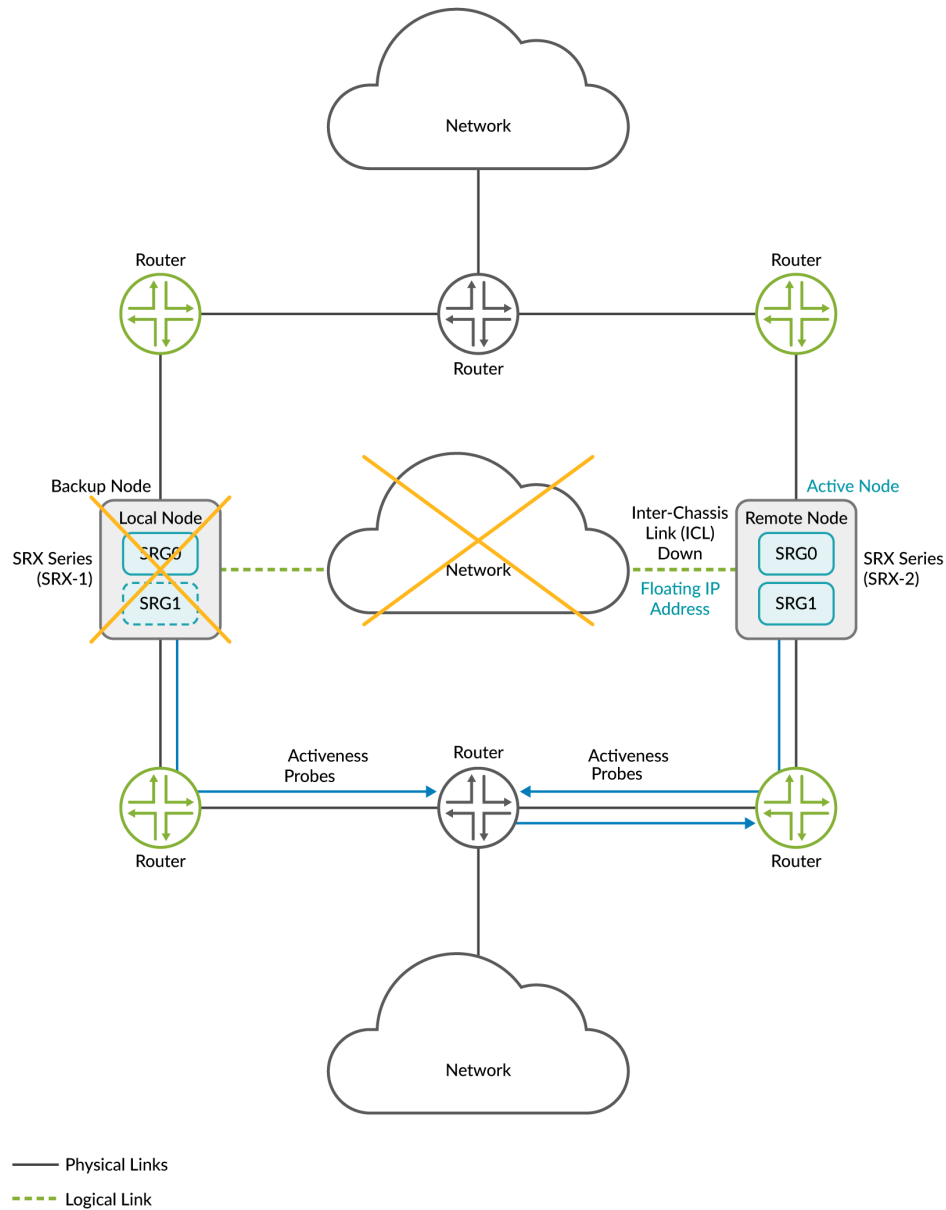


- The upstream router, that hosts the probe destination IP address, receives the ICMP probes from both nodes.
- Upstream router replies to only to the active node; because it's configuration has the higher preference path for the active node
- The active node retains the active role.

#### If Active Node is Down



When the ICL between the nodes goes down, both nodes initiate an activeness determination probe (ICMP probe). The nodes use the floating IP address (activeness determination IP address) as source IP address and IP addresses of the upstream routers as destination IP address for the probes.



- The remote node restarts the activeness determination probes.
- The router hosting the probe destination IP address has lost its higher preference path (of former active node) and replies to the remote node.
- The probe result is a success for the remote node and the remote node transitions to the active state.



As demonstrated in the above cases, activeness determination probes and the configuration of higher path preference in the upstream router ensures one node always stays in the active role and prevents split-brain taking place.

You must also ensure that ICMP packets can be reached and allowed all the way to the router hosting the probe destination IP address.

See ["Configuring Multinode High Availability In a Layer 3" on page 48](#) for details.

## Default Gateway Deployments

In a default gateway deployments, when the ICL connection is down, SRX Series devices are not able to communicate with each other. In such case, there is a possibility of both devices could claim active role. To prevent this, Multinode High Availability probes using ICMP-based ping to the virtual IP from the backup node. Following two scenarios are possible:

- **ICL Down and Active Node Up**

The active node owns the virtual IP address and hence when active node pings to the virtual IP using ICMP probes, the probe succeeds. The backup node remains in backup state.

- **ICL and Active Node Down**

The backup node pings the virtual IP using ICMP probes. Because the active node is down, it does not host VIP, and does not respond to the IP-based probes. So after a specified number of failures, the backup node transitions to active state.

See ["Configuring Multinode High Availability In a Default Gateway Deployment" on page 93](#) for details.

## Hybrid Deployments

You can use Layer 3 side or Layer 2 side for split brain prevention. If you use the Layer 2 side, then Multinode High Availability uses the VIP probing mechanism. If you use Layer 3 side, then the activeness determination probe (ICMP probe) method is used. If you configure the activeness determination probes, then the Layer 3 side probing takes place. Or else, virtual IP with index 1 is used for probing at the Layer 2 side.

See ["Configuring Multinode High Availability In a Default Gateway Deployment" on page 93](#) or ["Configuring Multinode High Availability In a Layer 3" on page 48](#).

In spite of the split brain prevention mechanism, theoretically the nodes can still get in to a active-active state. This happens when the ICL is down and there are other network issues on the probe router at the same time. Because of this, the probe router replies to probe requests from both the nodes. In this case,



once the situation improves and the ICL is up, one of the nodes takes up the active role based on your activeness-priority configuration. In case the activeness-priority configuration is not available, the node with lower local ID takes up the backup role.

## Multinode High Availability Monitoring

### IN THIS SECTION

- [Multinode High Availability Failure Scenarios | 31](#)
- [Node Failure | 31](#)
- [Network/Connectivity Failure | 34](#)

A high availability failure is a loss of connection between the nodes caused by device failure or network failure or loss in connectivity between the nodes. In a typical deployment of Multinode High Availability, the network includes multiple hops on north and southbound networks. There is a possibility of failures at any of the hops, or there could be hardware/software issues. Multinode High Availability system remains available even during failures with following monitoring capabilities:

- **BFD Monitoring**

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. The devices send hello packets at a specified, regular interval and detects a failure when the routing device stops responding after a specified interval. BFD monitoring ensures the reachability to the adjacent router and enables quick failover between a active and a backup node.

You can configure Multinode High Availability to monitor one or more links using BFD. This configuration triggers a failover in the event of BFD failure. You can use this optional feature for additional reliability in Multinode high Availability. You can configure BFD liveliness by specifying source and destination IP and the interface connecting to the peer device.

- **IP Monitoring**

IP monitoring is a technique that checks the reachability of an IP address or a set of IP addresses using ICMP ping messages. When using IP monitoring, both active node and backup node ping the specified destination IP address at the same time. If both nodes can successfully ping the target, no failover occurs. But, if one node can ping the target but the other node cannot, the SRG automatically fails-over to the other node.



Multinode High Availability uses IP monitoring for:

- Activeness determination for the nodes in case lost communication with the peer node
- Monitoring the routing path all the way up to the last router towards the end hosts
- **Interface Monitoring**

In a Multinode High Availability setup, if an interface on any side goes down, it can cause an outage. For Layer 3 deployments, BFD monitoring addresses the problem. However, for default gateway and hybrid mode, you must configure interface monitoring option for each SRG.

The node which detects the interface monitoring failure transitions to ineligible state for the corresponding SRG and the other node (if healthy) takes over the active role or that SRG and the subsequent GARP ensures traffic switching and recovery.

## Multinode High Availability Failure Scenarios

The following sections describe possible failure scenarios: how a failure is detected, what recovery action to take, and if applicable, the impact on the system caused by the failure.

### Node Failure

#### Hardware Failure

- **Cause**—A failed hardware component or an environmental issue such as a power failure.
- **Detection**— In Multinode High Availability
  - Affected device/node not accessible
  - SRG1 status changes to INELIGIBLE

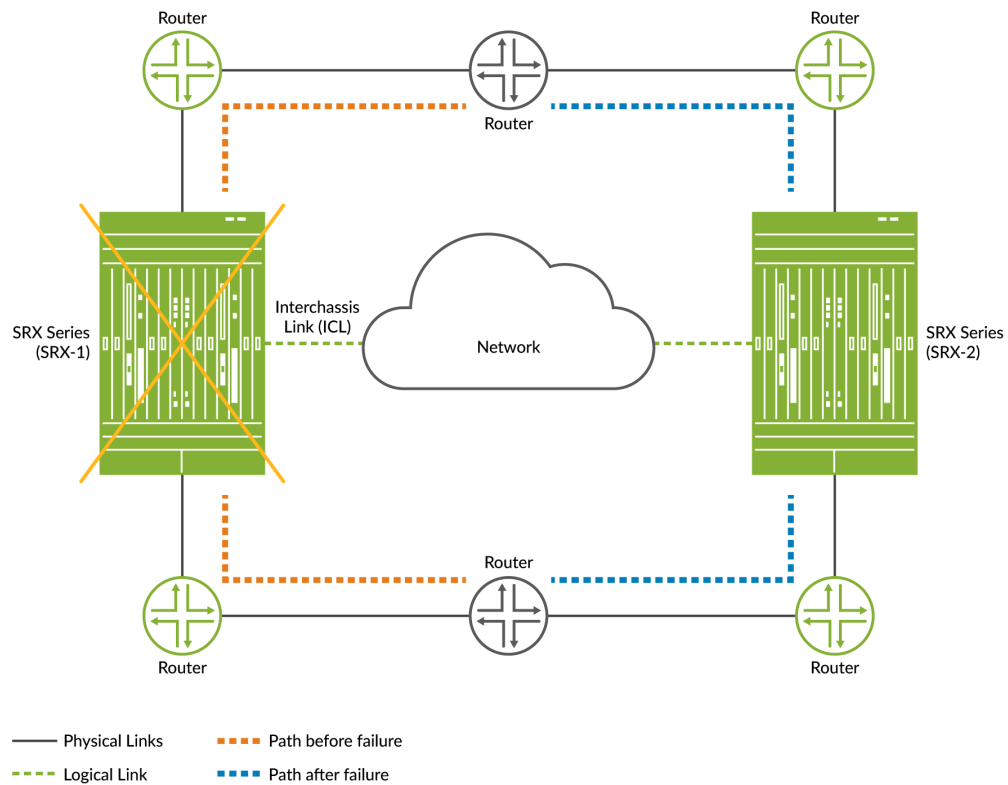
On SRX5000-line devices, Multinode High Availability automatically detects hardware failure based on chassis hardware failure detection results.

- **Impact** —Hardware failure on the active node triggers a failover immediately. The backup node takes over the active role and continues to process traffic. As shown in [Figure 13 on page 32](#), When a



hardware failure detected in SRX-1, the SRX-2 takes up active role after a failover. Traffic is diverted to SRX-2.

Figure 13: Hardware Failure in Multinode High Availability



- **Recovery**—Recovery of failure takes place when you clear chassis hardware failure (ex: replace or repair the failed hardware component).
- **Results**—Check status using the following commands:
  - `show chassis high-availability information detail` on page 275
  - `show chassis hardware`
  - `show chassis alarms`

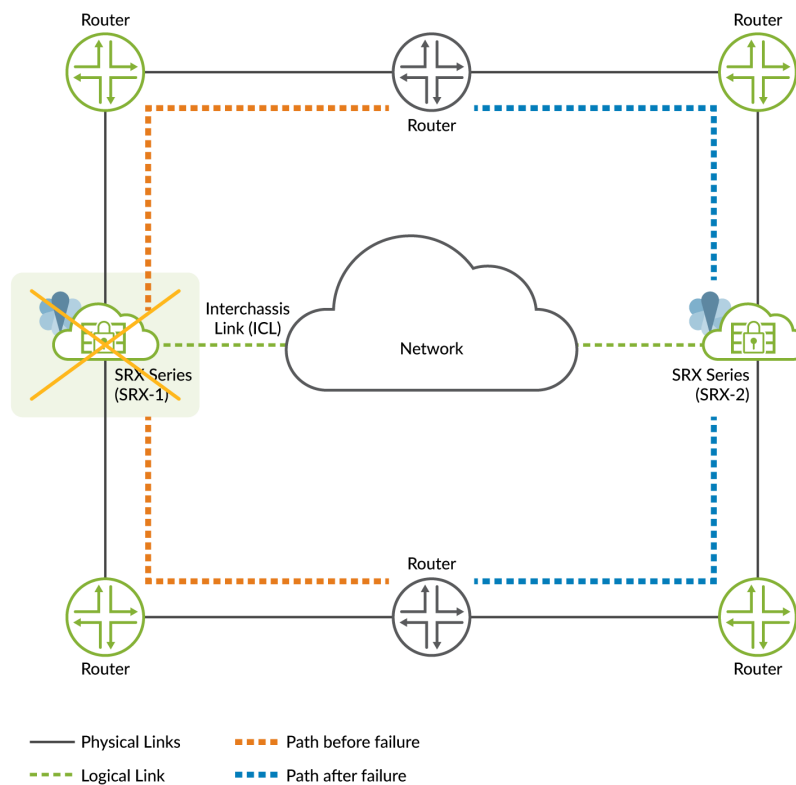
### System/Software Failure

- **Cause**—A failure in software process or service or issues with operating system.
- **Detection**— In Multinode High Availability



- Affected device/node not accessible
- SRG1 status changes to INELIGIBLE on the affected node.
- **Impact**—Software/system failure on the active node triggers a failover immediately. The backup nodes takes over the active role and continues to process traffic. [Figure 14 on page 33](#) shown software failure. When a software failure is detected in SRX-1, the SRX-1 is isolated and SRX-2 takes up the active role. After a failover, traffic and services are migrated to SRX-02.

**Figure 14: Software Failure in Multinode High Availability**



- **Recovery**—Automatically and gracefully recovers from the outage once the issue is addressed. The backup node that has taken the active role, continues to remain active. The formerly active node remains as the backup node.
- **Results**—Check status using the ["show chassis high-availability information detail"](#) on [page 275](#) command.



## Network/Connectivity Failure

### Physical Interfaces (Link) Failure

- **Cause**—A failure in interfaces could be due to network outages, or disruption with physical cable or inconsistent configurations.
- **Detection**— In Multinode High Availability
  - Affected device/node is not accessible.
  - SRG1 status changes to INELIGIBLE on the affected node.

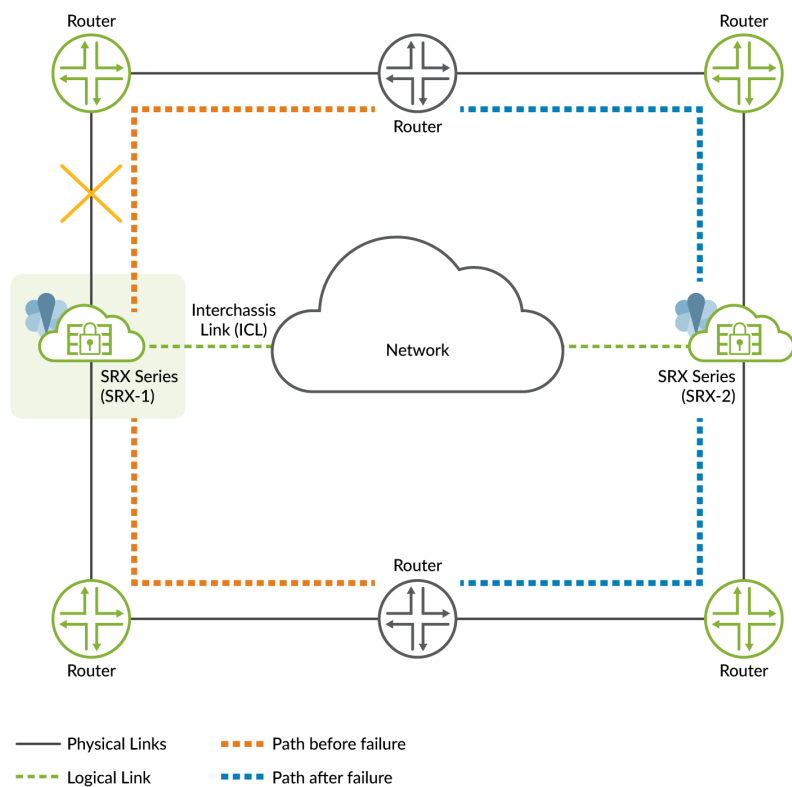
In a Layer 3 deployments, BFD monitoring detects interface failure. For Layer 2 (default gateway) and hybrid deployments, interface monitoring detects interface failure.

- **Impact**—A change in the link state of the interfaces triggers a failover. The backup node takes up the active role, and services that were running on the failed node are migrated to other node.



As shown in [Figure 15 on page 35](#), interface connected to SRX-1 goes down. In this case, SRX-1 changes into BACKUP state and SRX-2 takes over as active role and starts processing traffic.

**Figure 15: Interface Failure**



- **Configuration**—To configure BFD monitoring and interface monitoring, use the following configuration statement:

```
set chassis high-availability services-redundancy-group <1> monitor bfd-liveliness <source-ip-address> <destination-ip-address> routing-instance <routing-instance-name> <single-hop|multihop> <interface-name>
```

```
set chassis high-availability services-redundancy-group <1> monitor interface <interface-name>
```

All links critical to traffic flow should be monitored.

Checkout *Configuring Multinode High Availability In a Layer 3 Network* or *Configuring Multinode High Availability In a Default Gateway Deployment* for complete configuration details.



- **Recovery**—Recovers when you repair/replace the failed interface. After the network/connectivity failure recovers, SRG1 moves from the INELIGIBLE state to the BACKUP state. The new-active node continues advertise better metrics to its upstream router and processes traffic.
- **Results**—Check status using the following commands:
  - `show chassis high-availability information detail`
  - `monitor interfaces`
  - `show interfaces terse`
- For information on configuring interfaces, see *Configuring Multinode High Availability In a Layer 3 Network*, *Configuring Multinode High Availability In a Hybrid Deployment*, *Configuring Multinode High Availability In a Default Gateway Deployment*, [Troubleshooting Interfaces](#).

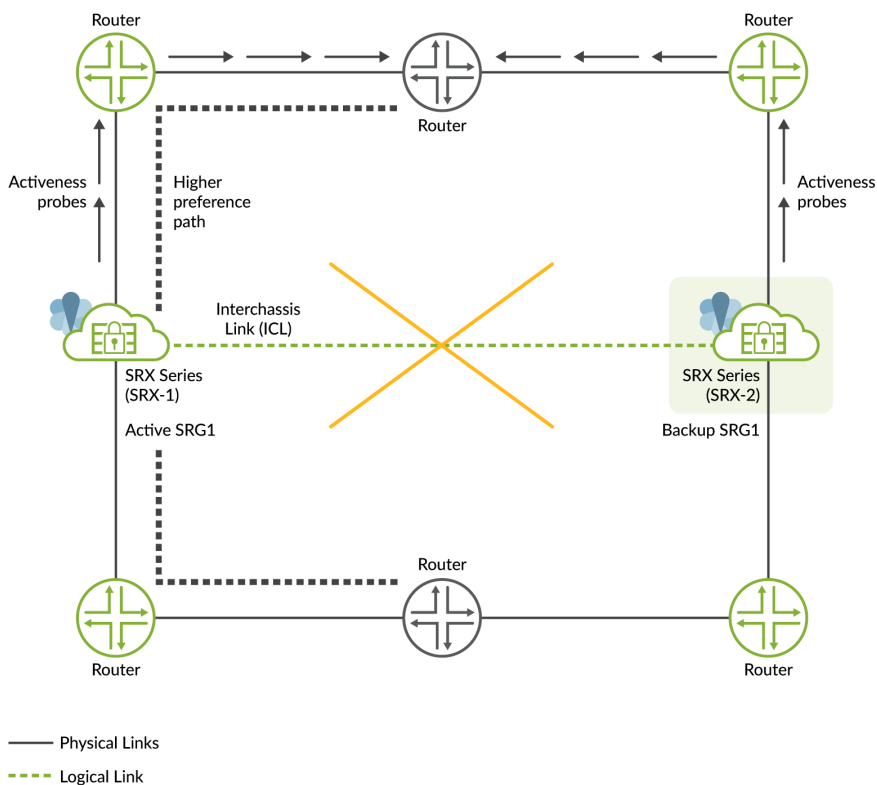
### Interchassis Link (ICL) Failure

- **Cause**—A failure in ICL could be due to network outages, or inconsistent configurations.
- **Detection**— In Multinode High Availability, nodes cannot reach each other and they initiate a activeness determination probe (ICMP probe).  
The IP monitoring can detect ICL failures.
- **Impact**— In a Multinode High Availability system, ICL connects active and backup nodes; if ICL goes down, both devices are affected. Both nodes initiate a activeness determination probe (ICMP probe). Based on the probe result, one of the node transitions to the active state.  
As shown in [Figure 16 on page 37](#), the ICL between SRX-1 and SRX-2 goes down. Both devices cannot reach each other and start sending activeness probes to the upstream router. Since SRX-1 is



on higher preferred path in the router configuration, it takes up active role and continues to process traffic and advertises higher preference path. The other takes up backup role.

**Figure 16: ICL Failure in Multinode High Availability**



- **Configuration**—To configure the activeness probing, use the following configuration statement:

```
set chassis high-availability services-redundancy-group <1> activeness-probe <destination-ip-address> routing-instance <routing-instance-name>
```

Checkout *Configuring Multinode High Availability In a Layer 3 Network* for complete configuration details.

- **Results**—Check status using the following commands:
  - `show chassis high-availability information detail`
  - `show chassis high-availability services-redundancy-group 1`



- Check ICMP packet reply from the upstream router using ping option. Example: `ping <activeness-probe-dest-ip> source <activeness-probe-source-ip> routing-instance <routing-instance-name>`.
- **Recovery**—Once one of the nodes assumes active role, Multinode High Availability restarts cold synchronization process and resynchronizes control-plane services (IPSec VPN). SRG state information is re-exchanged between the nodes.

### Unreachable Upstream/Downstream Routers

- **Cause**—Link failure, unreachable upstream routers on the untrust side results in external path failure. Link failure and unreachable downstream routers on the trust side results in internal path failure.
- **Detection**— In Multinode High Availability
  - Nodes cannot reach each other and they initiate a activeness determination probe (ICMP probe).
  - SRG1 changes to INELIGIBLE state on the current node.

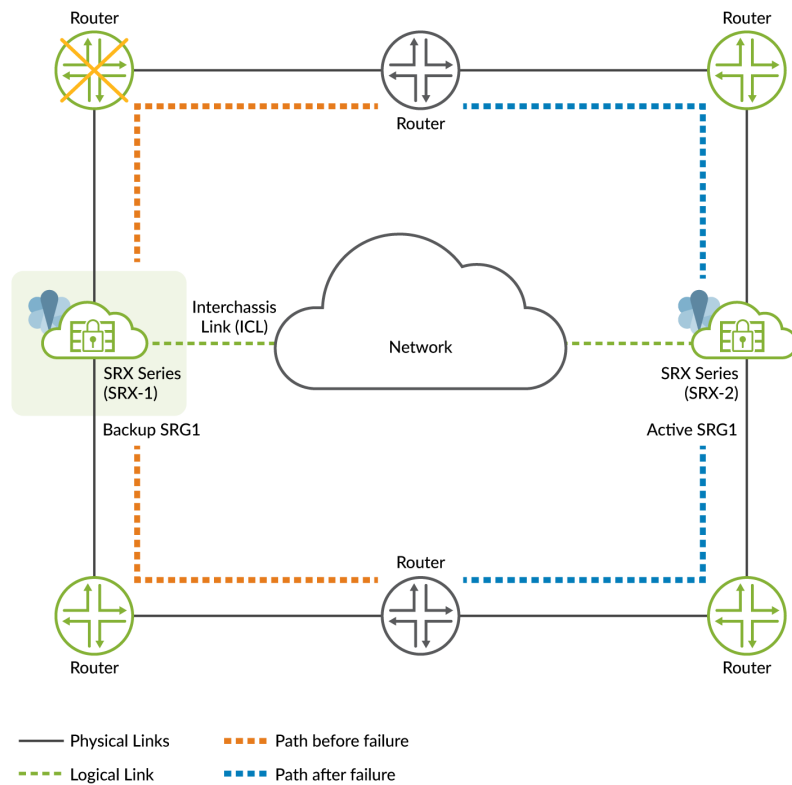
The BFD monitoring can detect path/link failures.

- **Impact**—The backup node transitions to the active state. Routes are re-advertised to swap the preference, so that packets start taking path to the new active node.



As shown in Figure 17 on page 39, the upstream router on SRX-1 side becomes inactive. In this case, SRX-1 changes into ineligible state and SRX-2 takes over as active role and starts processing traffic.

Figure 17: Unreachable Upstream/Downstream Routers



jrn-000402

- **Configuration**—To configure BFD monitoring, use the following configuration statement:

```
set chassis high-availability services-redundancy-group <1> monitor bfd-liveliness <source-ip-address> <destination-ip-address> routing-instance <routing-instance-name> <single-hop|multihop> <interface-name>
```

Checkout *Configuring Multinode High Availability In a Layer 3 Network* for complete configuration details.

- **Results**—Check status using the following commands:
  - `show chassis high-availability information detail`
  - `show chassis high-availability services-redundancy-group 1`



- [show bfd session](#)
- **Recovery**—Recovers when you enable/address the failed link or device. The backup node, that has taken the active role, continues to remain active. The formerly active node remains as the backup node. In case the backup node was down and never took over the active role for some reasons, the active node resumes the active role.

## SEE ALSO

[Prepare Your Environment for Multinode High Availability Deployment | 40](#)

[Multinode High Availability Services | 43](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)

[Insert Additional SRX5K-SPC3 in a Multinode High Availability Setup | 198](#)

[Software Upgrade in Multinode High Availability | 179](#)

[Multinode High Availability Support for vSRX Instances | 203](#)

# Prepare Your Environment for Multinode High Availability Deployment

This topic provides details to prepare the environment for Multinode High Availability deployment.

## Device Model

In Multinode High Availability, you must use the same SRX Series device model as your nodes. For example, if you use the SRX5600 as one node, you must use another SRX5600 as the other node

In case of the SRX5000 line of devices, ensure that SPCs, NPCs, and IOCs have the same slot placement and type.

We support Multinode High Availability on the following devices:

- SRX5800, SRX5600, SRX5400 with the following components running Junos OS Release 20.4R1 or later:



- Services Processing Card SPC3
- I/O card IOC3
- Switch Control Boards SCB3 and SCB4
- Routing Engine RE3
- SRX4600, SRX4200, SRX4100, and SRX1500 running Junos OS Release 22.3R1 or later
- vSRX running Junos OS Release 22.3R1 or later

## Software Version

Install the compatible version of Junos OS on the participating security devices.

## Latest Junos IKE Package

You must install IKE package for enabling ICL encryption in Multinode High Availability solution.

By default, when your SRX Series device boots up, the legacy IKE architecture is executed. To enable the new IKE architecture, you must install the new Junos IKE package. This is an optional package included in the Junos OS software download image.

Use the following command to install the IKE package:

```
user@host> request system software add optional://junos-ike.tgz
```

After you install the Junos IKE package, for subsequent software upgrades of the instance, the Junos IKE package is upgraded automatically from the new Junos OS releases installed on your device.

## Software Licenses

You do not need any specific license for the Multinode High Availability feature. However, licenses are unique to each SRX Series and cannot be shared between the nodes in a Multinode High Availability setup. Therefore, you must use identical licenses on both the nodes. If both SRX Series devices do not have an identical set of licenses, the system is not ready for the deployment.

## Network Accessibility

Both the nodes in the Multinode High Availability setup must be able to reach each other using the ICL path. This path uses (whether the ICL is encrypted or not) IP address, protocol, and port details. You



must ensure that this communication is allowed between the nodes if any firewall or other inspection is in place.

The floating IP address that you use for each node must be routable IP (logical routed path) across the network.

## IP Address Consideration

[Table 4 on page 42](#) provides details on IPv4 and IPv6 address support for Multinode High Availability deployments.

**Table 4: IP Address Consideration For Multinode High Availability**

MNHA Deployment Type	Layer 3 Network (Routers at Both Ends)	Hybrid Network (Router at One End and Switch at the Other End)	Default Gateway (Switches at Both Ends)
IPv4 and IPv6 addresses for IP monitoring	Yes	Yes	Yes
IPv4 and IPv6 addresses for activeness probing	Yes	Yes	Yes
Virtual IPv4 and IPv6 addresses	Not applicable	Yes	Yes

## RELATED DOCUMENTATION

[Multinode High Availability | 2](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)

[Multinode High Availability Services | 43](#)



# Multinode High Availability Services

## IN THIS SECTION

- [Control Plane Stateless Services | 43](#)
- [Network Address Translation | 44](#)
- [Firewall User Authentication | 44](#)
- [IPSec VPN | 45](#)

Multinode High Availability supports active/active mode for data plane and active/backup mode for control plane services. Lets learn about control plane stateless and stateful services in the following sections:

## Control Plane Stateless Services

SRG0 manages services without control plane state, such as application security, IDP, UTM, firewall, NAT, policies, ALG, and so on. Failover for these services is required at data plane level only and some of these services are pass through (not terminating on the device except NAT, firewall authentication).

SRG0 remains active on both nodes and forwards traffic from both the nodes. These feature works independently on both SRX Series devices in Multinode High Availability.

To configure the control plane stateless services:

- Configure the features as you configure them on a stand-alone SRX Series device.
- Install the same Junos OS version on the participating security devices (Junos OS Release 22.3R1 or later)
- Install identical licenses on both the nodes
- Download and install same versions of application signature package or IPS package on both nodes (if you are using application security and IDP)
- Configure conditional route advertisement, routing policy, and static routes as per your requirements.



- In Multinode High Availability, configuration synchronization does not happen by default. You need to configure applications as part of groups and then synchronize the configuration using the peer synchronization option or manage configuration independently on each node.

## Network Address Translation

Services such as Firewall, ALG, NAT do not have control plane state. For such services, only data plane state needs to be synchronized across the nodes.

In a Multinode High Availability setup, one device handles a NAT session at a time, and the other device takes over the active role when failover happens. So, a session remains active on one device, and on the other device, the session will be in warm (standby) state till failover happens.

NAT sessions and ALG state objects get synchronized between the nodes. If one node fails, the second node continues to process traffic for the synchronized sessions from the failed device, including NAT translations.

You must create NAT rules and pools with the same parameters on both the SRX Series devices. To steer the response path for the NAT traffic (destined to NAT pool IP address) to the correct SRX Series device (active device), you must have the required routing configuration on both active/backup devices. That is, the configuration must specify what routes are advertised via the routing protocols to the adjacent routing devices. Accordingly, you must also configure policy-option and route configuration.

When you run NAT-specific operational commands on both devices, you can see the same output. However, there could be instances where NAT rule / pool internal numerical IDs can be different between the nodes. Different numerical IDs don't impact the session sync/ NAT translations upon failover.

## Firewall User Authentication

With firewall authentication, you can restrict or permit users individually or in groups. Users can be authenticated using a local password database or using an external password database.

Multinode High Availability supports following authentication methods:

- Pass-through authentication
- Pass-through with web-redirect authentication
- Web authentication



Firewall user authentication is service with a active control plane state and requires control and data plane states synchronization across the nodes. While working in Multinode High Availability setup, the firewall user authentication feature works independently on both SRX Series devices and synchronizes the authentication table between the nodes. When a user authenticates successfully, authentication entry gets synced to the other node and is visible on both the nodes when you run show command (example: `show security firewall-authentication users` ).

Multinode High Availability supports Juniper Identity Management Service (JIMS) to obtain user identity information. Each node fetches the authentication entries from JIMS server and process them independently. Because of this,

You must run firewall user authentication commands separately on each node. For example, when you display the auth entries using the show commands, each node displays only those auth entries that it is handling currently (as if working independently in standalone mode:

- `show services user-identification authentication-table`
- `show service user-identification identity-management`

## IPSec VPN

When an IPsec VPN tunnel anchors at SRG1 in a Multinode High Availability setup, SRG1 acts in stateful active / backup mode, where the active node is responsible for all tunnel establishment and key exchanges. This is accomplished with the associated floating IP address at the SRG1, that moves between the nodes in case of a failure. When there is a need for a dynamic CA profile to authenticate the tunnel establishment, the CA profile will be generated on the active node of the SRG1. After a failover, new authentication takes place and the dynamic profile will be loaded on the newly active node and cleared on the old node.

Although you can run the show commands on both active and backup nodes to display the status of IKE and IPsec security associations, you can delete the IKE and IPsec security associations only on the active node.

VPN service is automatically enabled when you enable the active/backup mode using the `set chassis high-availability services-redundancy-group 1` command. See the configuration example for more details.

**NOTE:** PKI files are synchronized to the peer node only if you enable link encryption for the ICL.



**TIP:** We recommend following sequence when you configure VPN with Multinode High Availability on your security device:

- On the backup node, configure security IKE gateway, IPsec VPN, interfaces st0.x, and security zones and then commit the configuration.
- On the active node, configure security IKE gateway, IPsec VPN, st0.x interface, security zones, and static route and commit the configuration.

You must commit the configuration on the backup node before committing configuration on the active node if you don't use the commit synchronize option.

When you use the process packet on back up option in Multinode High Availability, this enables the packet forward engine (PFE) to forward packets on backup node for the corresponding SRG. This configuration processes VPN packets on the backup node even when the node is not active; thus eliminating the delay when backup node transitions to the active role after a failover. The packet process continues even during the transition period.

You can configure the process packet on backup on an SRG1 using the `[set chassis high-availability services-redundancy-group name process-packet-on-backup]` statement.

## RELATED DOCUMENTATION

[Multinode High Availability | 2](#)

[Prepare Your Environment for Multinode High Availability Deployment | 40](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)



# 2

CHAPTER

## Multinode High Availability Configuration

---

Example: Configure Multinode High Availability in a Layer 3 Network | 48

Example: Configure Multinode High Availability in a Default Gateway  
Deployment | 93

Example: Configure Multinode High Availability in a Hybrid Deployment | 134

---



# Example: Configure Multinode High Availability in a Layer 3 Network

## SUMMARY

Read this topic to understand how to configure the Multinode High Availability solution on SRX Series devices. The example covers configuration in active/backup mode when SRX Series devices are connected to routers on both sides.

## IN THIS SECTION

- [Overview | 48](#)
- [Requirements | 48](#)
- [Topology | 50](#)
- [Configuration | 52](#)
- [Verification | 79](#)

## Overview

In Multi-Node High Availability, participating SRX Series devices operate as independent nodes in a Layer 3 network. The nodes are connected to adjacent infrastructure belonging to different networks. An encrypted logical interchassis link (ICL) connects the nodes over a routed network. Participating nodes backup each other to ensure a fast synchronized failover in case of system or hardware failure.

In Multinode High Availability, activeness is determined at the services redundancy group (SRG) level. The SRX Series device, on which the SRG1 is active, hosts the floating IP address and steers traffic towards it using the floating IP address. During a failover, the floating IP address moves from the old active node to the new active node and continues the communication client devices.

**NOTE:** As of Junos OS Release 22.3R1, we support a two-node configuration in the Multinode High Availability solution.

## Requirements

This example uses the following hardware and software components:

- Two SRX Series devices or vSRX instances



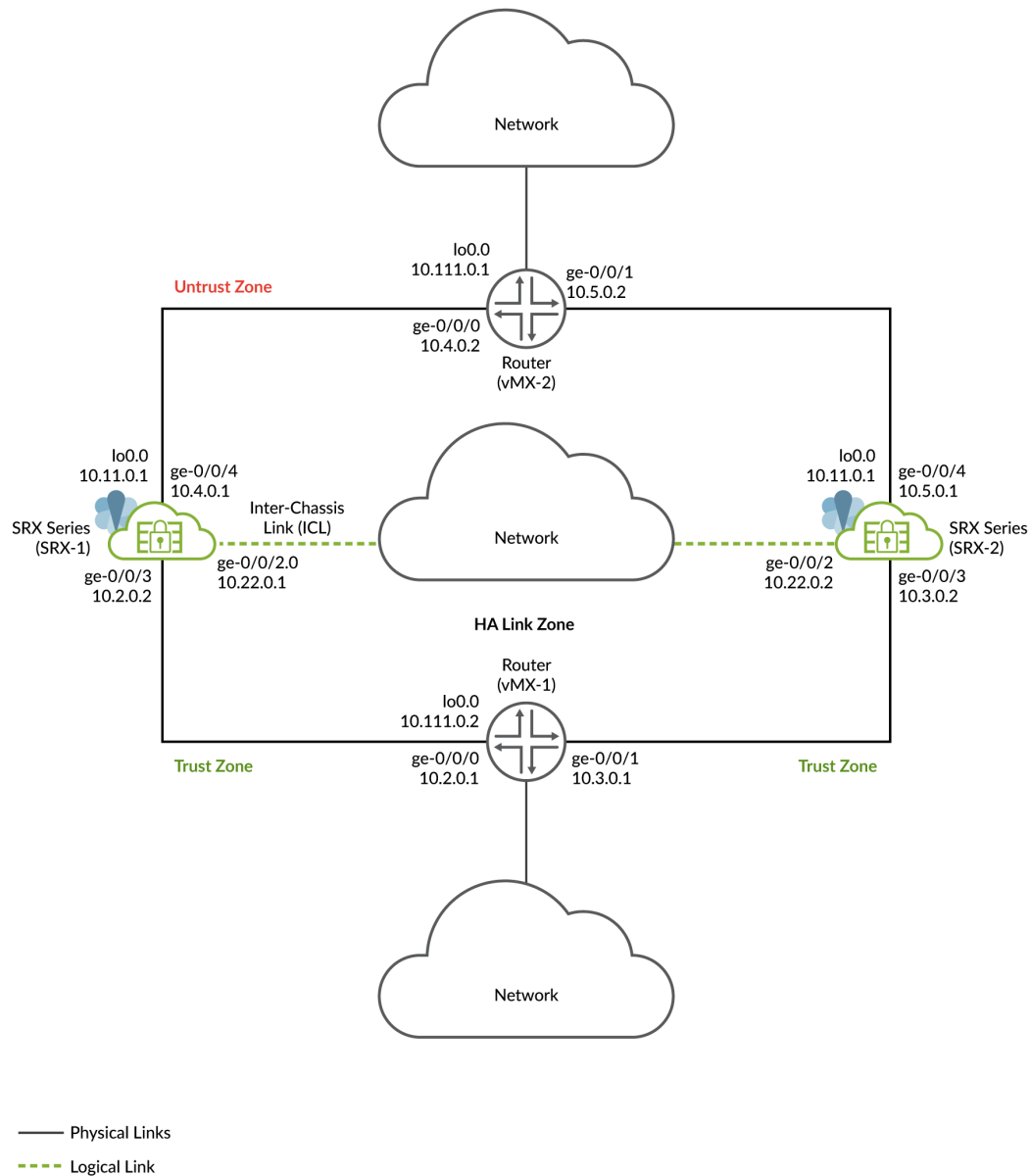
- Two Juniper Networks(R) MX960 Universal Routing Platform
- Junos OS Release 22.3R1



## Topology

Figure 18 on page 50 shows the topology used in this example.

Figure 18: Multinode High Availability in Layer 3 Network



As shown in the topology, two SRX Series devices are connected to adjacent routers on trust and untrust side forming a BGP neighborhood. An encrypted logical interchassis link (ICL) connects the nodes



over a routed network. The nodes communicate with each other using a routable IP address (floating IP address) over the network. Loopback interfaces are used to host the IP addresses on SRX Series and routers.

In general, you can use Aggregated Ethernet (AE) or a revenue Ethernet port on the SRX Series devices to setup an ICL connection. In this example, we've used GE ports for the ICL. We've also configured a routing instance for the ICL path to ensure maximum segmentation.

In a typical high availability deployment, you have multiple routers and switches on the northbound and southbound sides of the network. For this example, we are using two routers on both sides of SRX Series devices.

In this example, you'll establish high availability between the SRX Series devices and secure the tunnel traffic by enabling HA link encryption.

You'll perform the following tasks to build a Multinode High Availability setup:

- Configure a pair of SRX Series devices as local and peer nodes by assigning IDs.
- Configure services redundancy groups.
- Configure a loopback interface (lo0.0) to host the floating IP address.
- Configure IP probes for the activeness determination and enforcement
- Configure a signal route required for activeness enforcement and use it along with the route exists policy.
- Configure a VPN profile for the high availability (ICL) traffic using IKEv2.
- Configure BFD monitoring options
- Configure a routing policy and routing options
- Configure appropriate security policies to manage traffic in your network
- Configure stateless firewall filtering and quality of service (QoS) as per your network requirements.
- Configure interfaces and zones according to your network requirement. You must allow services such as IKE for link encryption and SSH for configuration synchronization as host-inbound system services on the security zone that is associated with the ICL.

You can configure the following options on SRG0 and SRG1:

- SRG1: Active/backup signal route, deployment type, activeness priority, preemption, virtual IP address (for default gateway deployments), activeness probing and process packet on backup.
- SRG1: BFD monitoring, IP monitoring, and interface monitoring options on SRG1.



- SRG0: shutdown on failure and install on failure route options.

For interchassis link (ICL), we recommend the following configuration settings:

- Use a loopback (lo0) interface using an aggregated Ethernet interface (ae0), or any revenue Ethernet interface to establish the ICL.
- Set MTU of 1514
- Allow the following services on the security zone associated with interfaces used for ICL
  - IKE, high-availability, SSH
  - Protocols depending on the routing protocol you need.
  - BFD to monitor the neighboring routes.

## Configuration

### IN THIS SECTION

- [Before You Begin | 52](#)
- [CLI Quick Configuration | 53](#)
- [Configuration | 59](#)
- [Results \(SRX-1\) | 67](#)
- [Results \(SRX-2\) | 73](#)

## Before You Begin

Install the Junos IKE package on your SRX Series. You require this step for ICL encryption.

```
user@host> request system software add optional://junos-ike.tgz
Verified junos-ike signed by PackageProductionECP256_2022 method ECDSA256+SHA256
Rebuilding schema and Activating configuration...
mgd: commit complete
Restarting MGD ...

WARNING: cli has been replaced by an updated version:
```



```
CLI release 20220208.163814_builder.r1239105 built by builder on 2022-02-08 17:07:55 UTC
Restart cli using the new version ? [yes,no] (yes)
```

## CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, 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 may vary based on the specific requirements of your environment.

### On SRX-1 Device

```
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 10.22.0.1
set chassis high-availability peer-id 2 peer-ip 10.22.0.2
set chassis high-availability peer-id 2 interface ge-0/0/2.0
set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL
set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
set chassis high-availability peer-id 2 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 0 peer-id 2
set chassis high-availability services-redundancy-group 1 deployment-type routing
set chassis high-availability services-redundancy-group 1 peer-id 2
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip 10.111.0.1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip src-ip
10.11.0.1
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.4.0.2 src-ip
10.4.0.1
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.4.0.2
session-type singlehop
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.4.0.2
interface ge-0/0/4.0
set chassis high-availability services-redundancy-group 1 active-signal-route 10.39.1.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 10.39.1.2
set chassis high-availability services-redundancy-group 1 preemption
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set interfaces ge-0/0/3 description "trust" unit 0 family inet address 10.2.0.2/16
set interfaces ge-0/0/4 description "untrust" unit 0 family inet address 10.4.0.1/16
set interfaces ge-0/0/2 description "ha_link" unit 0 family inet address 10.22.0.1/24
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.1/32
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.2/32
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.3/32
```



```

set routing-options autonomous-system 100
set routing-options static route 10.1.0.0/16 next-hop 10.2.0.1
set routing-options static route 10.6.0.0/16 next-hop 10.4.0.2
set routing-options static route 10.111.0.1 next-hop 10.2.0.1
set routing-options static route 10.111.0.2 next-hop 10.4.0.2
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic protocols bfd
set security zones security-zone untrust host-inbound-traffic protocols bgp
set security zones security-zone untrust interfaces ge-0/0/4
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3
set security zones security-zone halink host-inbound-traffic system-services ike
set security zones security-zone halink host-inbound-traffic system-services ping
set security zones security-zone halink host-inbound-traffic system-services high-availability
set security zones security-zone halink host-inbound-traffic system-services ssh
set security zones security-zone halink host-inbound-traffic protocols bfd
set security zones security-zone halink host-inbound-traffic protocols bgp
set security zones security-zone halink interfaces ge-0/0/2
set security policies default-policy permit-all
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
set system syslog file vpn_syslog any info
set system syslog file vpn_syslog match "iked|pkid|kmd|ikemd|authd|jsrpd|chassisd|bfd"
set system services netconf ssh
set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
set security ike proposal MNHA_IKE_PROP dh-group group14
set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
set security ike gateway MNHA_IKE_GW version v2-only
set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
set security ipsec proposal MNHA_IPSEC_PROP protocol esp
set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600

```



```

set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
set policy-options condition active_route_exists if-route-exists address-family inet 10.39.1.1
table inet.0
set policy-options condition backup_route_exists if-route-exists address-family inet 10.39.1.2
table inet.0
set policy-options policy-statement mnha-route-policy term 1 from protocol static
set policy-options policy-statement mnha-route-policy term 1 from protocol direct
set policy-options policy-statement mnha-route-policy term 1 from condition active_route_exists
set policy-options policy-statement mnha-route-policy term 1 then accept metric 10
set policy-options policy-statement mnha-route-policy term 2 from protocol static
set policy-options policy-statement mnha-route-policy term 2 from protocol direct
set policy-options policy-statement mnha-route-policy term 2 from condition backup_route_exists
set policy-options policy-statement mnha-route-policy term 2 then accept metric 20
set policy-options policy-statement mnha-route-policy term 3 from protocol static
set policy-options policy-statement mnha-route-policy term 3 from protocol direct
set policy-options policy-statement mnha-route-policy term 3 then accept metric 30
set policy-options policy-statement mnha-route-policy term default then reject
set protocols bgp group trust type internal
set protocols bgp group trust local-address 10.2.0.2
set protocols bgp group trust export mnha-route-policy
set protocols bgp group trust neighbor 10.2.0.1
set protocols bgp group trust bfd-liveness-detection minimum-interval 500
set protocols bgp group trust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group trust bfd-liveness-detection multiplier 3
set protocols bgp group trust local-as 100
set protocols bgp group untrust type internal
set protocols bgp group untrust local-address 10.4.0.1
set protocols bgp group untrust export mnha-route-policy
set protocols bgp group untrust neighbor 10.4.0.2
set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group untrust bfd-liveness-detection multiplier 3
set protocols bgp group untrust local-as 100

```

## On SRX-2 Device

```

set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 10.22.0.2

```



```

set chassis high-availability peer-id 1 peer-ip 10.22.0.1
set chassis high-availability peer-id 1 interface ge-0/0/2.0
set chassis high-availability peer-id 1 vpn-profile IPSEC_VPN_ICL
set chassis high-availability peer-id 1 liveness-detection minimum-interval 400
set chassis high-availability peer-id 1 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 0 peer-id 1
set chassis high-availability services-redundancy-group 1 deployment-type routing
set chassis high-availability services-redundancy-group 1 peer-id 1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip 10.111.0.1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip src-ip
10.11.0.1
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.5.0.2 src-ip
10.5.0.1
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.5.0.2
session-type singlehop
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.5.0.2
interface ge-0/0/4.0
set chassis high-availability services-redundancy-group 1 active-signal-route 10.39.1.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 10.39.1.2
set chassis high-availability services-redundancy-group 1 activeness-priority 1
set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
set security ike proposal MNHA_IKE_PROP dh-group group14
set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
set security ike gateway MNHA_IKE_GW version v2-only
set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
set security ipsec proposal MNHA_IPSEC_PROP protocol esp
set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
set interfaces ge-0/0/3 description "trust" unit 0 family inet address 10.3.0.2/16
set interfaces ge-0/0/4 description "untrust" unit 0 family inet address 10.5.0.1/16
set interfaces ge-0/0/2 description "ha_link" unit 0 family inet address 10.22.0.2/24

```



```

set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.1/32
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.2/32
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.3/32
set routing-options autonomous-system 100
set routing-options static route 10.1.0.0/16 next-hop 10.3.0.1
set routing-options static route 10.6.0.0/16 next-hop 10.5.0.2
set routing-options static route 10.111.0.1 next-hop 10.3.0.1
set routing-options static route 10.111.0.2 next-hop 10.5.0.2
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic protocols bfd
set security zones security-zone untrust host-inbound-traffic protocols bgp
set security zones security-zone untrust interfaces ge-0/0/4
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3
set security zones security-zone halink host-inbound-traffic system-services ike
set security zones security-zone halink host-inbound-traffic system-services ping
set security zones security-zone halink host-inbound-traffic system-services high-availability
set security zones security-zone halink host-inbound-traffic system-services ssh
set security zones security-zone halink host-inbound-traffic protocols bfd
set security zones security-zone halink host-inbound-traffic protocols bgp
set security zones security-zone halink interfaces ge-0/0/2
set security policies default-policy permit-all
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
set system syslog file vpn_syslog any info
set system syslog file vpn_syslog match "iked|pkid|kmd|ikemd|authd|jsrpd|chassisd|bfd"
set system services netconf ssh
set policy-options route-filter-list loopback 10.11.0.0/24 orlonger
set policy-options route-filter-list ipsec 10.6.0.0/16 orlonger
set policy-options condition active_route_exists if-route-exists address-family inet 10.39.1.1
table inet.0
set policy-options condition backup_route_exists if-route-exists address-family inet 10.39.1.2
table inet.0
set policy-options policy-statement mnha-route-policy term 1 from protocol static
set policy-options policy-statement mnha-route-policy term 1 from protocol direct
set policy-options policy-statement mnha-route-policy term 1 from condition active_route_exists
set policy-options policy-statement mnha-route-policy term 1 then accept metric 10
set policy-options policy-statement mnha-route-policy term 2 from protocol static
set policy-options policy-statement mnha-route-policy term 2 from protocol direct

```



```

set policy-options policy-statement mnha-route-policy term 2 from condition backup_route_exists
set policy-options policy-statement mnha-route-policy term 2 then accept metric 20
set policy-options policy-statement mnha-route-policy term 3 from protocol static
set policy-options policy-statement mnha-route-policy term 3 from protocol direct
set policy-options policy-statement mnha-route-policy term 3 then accept metric 30
set policy-options policy-statement mnha-route-policy term default then reject
set protocols bgp group trust type internal
set protocols bgp group trust local-address 10.3.0.2
set protocols bgp group trust export mnha-route-policy
set protocols bgp group trust neighbor 10.3.0.1
set protocols bgp group trust bfd-liveness-detection minimum-interval 500
set protocols bgp group trust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group trust bfd-liveness-detection multiplier 3
set protocols bgp group trust local-as 100
set protocols bgp group untrust type internal
set protocols bgp group untrust local-address 10.5.0.1
set protocols bgp group untrust export mnha-route-policy
set protocols bgp group untrust neighbor 10.5.0.2
set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group untrust bfd-liveness-detection multiplier 3
set protocols bgp group untrust local-as 100

```

The following sections show configuration snippets on the routers required for setting up Multinode High Availability setup in the network.

### Router (VMX-1)

```

set interfaces ge-0/0/2 description lan unit 0 family inet address 10.1.0.1/16
set interfaces ge-0/0/0 description ha unit 0 family inet address 10.2.0.1/16
set interfaces ge-0/0/1 description ha unit 0 family inet address 10.3.0.1/16
set interfaces lo0 description "loopback" unit 0 family inet address 10.111.0.1 primary preferred
set routing-options autonomous-system 100
set protocols bgp group mnha_r0 type internal
set protocols bgp group mnha_r0 local-address 10.2.0.1
set protocols bgp group mnha_r0 neighbor 10.2.0.2
set protocols bgp group mnha_r0 bfd-liveness-detection minimum-interval 500
set protocols bgp group mnha_r0 bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group mnha_r0 bfd-liveness-detection multiplier 3
set protocols bgp group mnha_r0 local-as 100
set protocols bgp group mnha_r0_b type internal
set protocols bgp group mnha_r0_b local-address 10.3.0.1

```



```

set protocols bgp group mnha_r0_b neighbor 10.3.0.2
set protocols bgp group mnha_r0_b bfd-liveness-detection minimum-interval 500
set protocols bgp group mnha_r0_b bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group mnha_r0_b bfd-liveness-detection multiplier 3
set protocols bgp group mnha_r0_b local-as 100

```

## Router (VMX-2)

```

set interfaces ge-0/0/0 description HA unit 0 family inet address 10.4.0.2/16
set interfaces ge-0/0/1 description HA unit 0 family inet address 10.5.0.2/16
set interfaces ge-0/0/2 description trust unit 0 family inet address 10.6.0.1/16
set interfaces lo0 description "loopback" unit 0 family inet address 10.111.0.2 primary preferred
set routing-options autonomous-system 100
set protocols bgp group mnha_r0 type internal
set protocols bgp group mnha_r0 local-address 10.4.0.2
set protocols bgp group mnha_r0 neighbor 10.4.0.1
set protocols bgp group mnha_r0 bfd-liveness-detection minimum-interval 500
set protocols bgp group mnha_r0 bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group mnha_r0 bfd-liveness-detection multiplier 3
set protocols bgp group mnha_r0 local-as 100
set protocols bgp group mnha_r0_b type internal
set protocols bgp group mnha_r0_b local-address 10.5.0.2
set protocols bgp group mnha_r0_b neighbor 10.5.0.1
set protocols bgp group mnha_r0_b bfd-liveness-detection minimum-interval 500
set protocols bgp group mnha_r0_b bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group mnha_r0_b bfd-liveness-detection multiplier 3
set protocols bgp group mnha_r0_b local-as 100

```

## Configuration

### Step-by-Step Procedure

We're showing the configuration of SRX-01 in the step-by-step procedure.

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the [CLI User Guide](#).

#### 1. Configure Interfaces.

```

[edit]
user@host# set interfaces ge-0/0/3 description "trust" unit 0 family inet address

```



```

10.2.0.2/16
user@host# set interfaces ge-0/0/4 description "untrust" unit 0 family inet address
10.4.0.1/16
user@host# set interfaces ge-0/0/2 description "ha_link" unit 0 family inet address
10.22.0.1/24

```

We're using ge-0/0/3 and ge-0/0/4 interfaces to connect to the upstream and downstream routers and using ge-0/0/2 interface to setup the ICL.

## 2. Configure the loopback interfaces.

```

[edit]
user@host# set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.1/32
user@host# set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.2/32
user@host# set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.3/32

```

The IP address (10.11.0.1) assigned to the loopback interface will be used as the floating IP address.

Using the loopback interface ensures that at any given point, traffic from the adjacent routers will be steered toward the floating IP address (that is, toward the active node).

## 3. Configure security zones, assign interfaces to the zones, and specify the allowed system services for the security zones.

```

[edit]
user@host# set security zones security-zone untrust host-inbound-traffic system-services ike
user@host# set security zones security-zone untrust host-inbound-traffic system-services
ping
user@host# set security zones security-zone untrust host-inbound-traffic protocols bfd
user@host# set security zones security-zone untrust host-inbound-traffic protocols bgp
user@host# set security zones security-zone untrust interfaces ge-0/0/4
user@host# set security zones security-zone untrust interfaces lo0.0
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces ge-0/0/3
user@host# set security zones security-zone halink host-inbound-traffic system-services ike
user@host# set security zones security-zone halink host-inbound-traffic system-services ping
user@host# set security zones security-zone halink host-inbound-traffic system-services
high-availability
user@host# set security zones security-zone halink host-inbound-traffic system-services ssh
user@host# set security zones security-zone halink host-inbound-traffic protocols bfd

```



```

user@host# set security zones security-zone halink host-inbound-traffic protocols bgp
user@host# set security zones security-zone halink interfaces ge-0/0/2

```

Assign the interfaces ge-0/0/3 and ge-0/0/4 the trust and untrust zones respectively. Assign the lo0.0 interface to the untrust zone to connect over the public IP network. Assign the interface ge-0/0/2 to the halink zone. You use this zone to set up the ICL.

#### 4. Configure routing options.

```

[edit]
user@host# set routing-options autonomous-system 100
user@host# set routing-options static route 10.1.0.0/16 next-hop 10.2.0.1
user@host# set routing-options static route 10.6.0.0/16 next-hop 10.4.0.2
user@host# set routing-options static route 10.111.0.1 next-hop 10.2.0.1
user@host# set routing-options static route 10.111.0.2 next-hop 10.4.0.2

```

#### 5. Configure both local node and peer node details such as node ID, IP addresses of local node and peer node, and the interface for the peer node.

```

[edit]
user@host# set chassis high-availability local-id 1
user@host# set chassis high-availability local-id local-ip 10.22.0.1
user@host# set chassis high-availability peer-id 2 peer-ip 10.22.0.2
user@host# set chassis high-availability peer-id 2 interface ge-0/0/2.0
user@host# set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL

```

You'll use the ge-0/0/2 interface for communicating with the peer node using the ICL.

#### 6. Attach the IPsec VPN profile IPSEC\_VPN\_ICL to the peer node.

```

[edit]
user@host# set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL

```

You'll need this configuration to establish a secure ICL link between the nodes.

#### 7. Configure Bidirectional Forwarding Detection (BFD) protocol options for the peer node.

```

[edit]
user@host# set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
user@host# set chassis high-availability peer-id 2 liveness-detection multiplier 5

```



8. Associate the peer node ID 2 to the services redundancy group 0 (SRG0).

```
[edit]
user@host# set chassis high-availability services-redundancy-group 0 peer-id 2
```

9. Configure the services redundancy group 1 (SRG1).

```
[edit]
user@host# set chassis high-availability services-redundancy-group 0 peer-id 2
user@host# set chassis high-availability services-redundancy-group 1 deployment-type routing
user@host# set chassis high-availability services-redundancy-group 1 peer-id 2
```

In this step, you are specifying deployment type as routing because you are setting up Multinode High Availability in a Layer 3 network.

10. Setup activeness determination parameters for SRG1.

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 activeness-probe dest-
ip 10.111.0.1
user@host# set chassis high-availability services-redundancy-group 1 activeness-probe dest-
ip src-ip 10.11.0.1
```

Use the floating IP address as source IP address (10.11.0.1) and IP addresses of the upstream routers as the destination IP address (10.111.0.1) for the activeness determination probe.

You can configure up to 64 IP addresses for IP monitoring and activeness probing. The total 64 IP addresses is sum of the number of IPv4 and IPv6 addresses)

11. Configure BFD monitoring parameters for the SRG1 to detect failures in network.

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.4.0.2 src-ip 10.4.0.1
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.4.0.2 session-type singlehop
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.4.0.2 interface ge-0/0/4.0
```



**12. Configure an active signal route required for activeness enforcement.**

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 active-signal-route
10.39.1.1
user@host# set chassis high-availability services-redundancy-group 1 backup-signal-route
10.39.1.2
user@host# set chassis high-availability services-redundancy-group 1 preemption
user@host# set chassis high-availability services-redundancy-group 1 activeness-priority 200
```

The active signal route IP address you assign is used for route preference advertisement. You must specify the active signal route along with the route-exists policy in the policy-options statement.

**13. Configure the security policy.**

```
[edit]
user@host# set security policies default-policy permit-all
```

Ensure you have configured security policies as per your network requirements.

**14. Configure CA certificates as per your requirements.**

```
[edit]
user@host# set security pki ca-profile Root-CA ca-identity Root-CA
user@host# set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/
mscep/mscep.dll
user@host# set security pki ca-profile Root-CA revocation-check disable
```

**15. Define Internet Key Exchange (IKE) configuration for Multinode High Availability. An IKE configuration defines the algorithms and keys used to establish a secure connection.**

```
[edit]
user@host# set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
user@host# set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
user@host# set security ike proposal MNHA_IKE_PROP dh-group group14
user@host# set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
user@host# set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
user@host# set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
user@host# set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
user@host# set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
```



```

user@host# set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
user@host# set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
user@host# set security ike gateway MNHA_IKE_GW version v2-only

```

For the Multinode High availability feature, you must configure the IKE version as v2-only

16. Specify the IPsec proposal protocol and encryption algorithm. Specify IPsec options to create a IPsec tunnel between two participant devices to secure VPN communication.

```

[edit]
user@host# set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
user@host# set security ipsec proposal MNHA_IPSEC_PROP protocol esp
user@host# set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
user@host# set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
user@host# set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
user@host# set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
user@host# set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
user@host# set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
user@host# set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL

```

Specifying the ha-link-encryption option encrypts the ICL to secure high availability traffic flow between the nodes.

The same VPN name IPSEC\_VPN\_ICL must be mentioned for *vpn\_profile* in chassis high availability configuration.

17. Configure policy options.

```

[edit]
user@host# set policy-options condition active_route_exists if-route-exists address-family
inet 10.39.1.1 table inet.0
user@host# set policy-options condition backup_route_exists if-route-exists address-family
inet 10.39.1.2 table inet.0
user@host# set policy-options policy-statement mnha-route-policy term 1 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 1 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists
user@host# set policy-options policy-statement mnha-route-policy term 1 then accept metric
10
user@host# set policy-options policy-statement mnha-route-policy term 2 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 2 from protocol
directuser@host# set policy-options policy-statement mnha-route-policy term 2 from
condition backup_route_exists

```



```

user@host# set policy-options policy-statement mnha-route-policy term 2 then accept metric
20
user@host# set policy-options policy-statement mnha-route-policy term 3 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 3 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 3 then accept metric
30
user@host# set policy-options policy-statement mnha-route-policy term default then reject

```

Configure the active signal route 10.39.1.1 with the route match condition (if-route-exists). The Multinode High Availability adds this route to the routing table when the node moves to the active role. The node also starts advertising the higher preference route. Configure the back up signal route (10.39.1.2) to advertise the backup node with a medium priority. In case of any failures, the high availability link goes down and the current active node releases it's primary role and removes the active-signal-route. Now the backup node detects the condition through it's probes and transitions to the active role. The route preference is swapped to drive all the traffic to the new active node

#### 18. Configure BFD peering sessions options and specify liveness detection timers.

```

[edit]
user@host# set protocols bgp group trust type internal
user@host# set protocols bgp group trust local-address 10.2.0.2
user@host# set protocols bgp group trust export mnha-route-policy
user@host# set protocols bgp group trust neighbor 10.2.0.1
user@host# set protocols bgp group trust bfd-liveness-detection minimum-interval 500
user@host# set protocols bgp group trust bfd-liveness-detection minimum-receive-interval 500
user@host# set protocols bgp group trust bfd-liveness-detection multiplier 3
user@host# set protocols bgp group trust local-as 100
user@host# set protocols bgp group untrust type internal
user@host# set protocols bgp group untrust local-address 10.4.0.1
user@host# set protocols bgp group untrust export mnha-route-policy
user@host# set protocols bgp group untrust neighbor 10.4.0.2
user@host# set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
user@host# set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval
500
user@host# set protocols bgp group untrust bfd-liveness-detection multiplier 3
user@host# set protocols bgp group untrust local-as

```



## Configuration Option for Software Upgrades (Optional)

In Multinode High Availability, during software upgrades, you can divert the traffic by changing the route. Use the following steps to add install route on failure configuration. Here, traffic can still go through the node and interface remains up.

Check ["Software Upgrade in Multinode High Availability" on page 179](#) for details.

1. Create a dedicated custom virtual router for the route used for diverting traffic during the upgrade.

```
user@host# set routing-instances MNHA-signal-routes instance-type virtual-router
```

2. Configure install route on failure statement for the SRGO.

```
user@host# set chassis high-availability services-redundancy-group 0 install-route-on-failure
10.39.1.3 routing-instance MNHA-signal-routes
user@host# set chassis high-availability services-redundancy-group 1 active-signal-route
10.39.1.1 routing-instance MNHA-signal-routes
user@host# set chassis high-availability services-redundancy-group 1 backup-signal-route
10.39.1.2 routing-instance MNHA-signal-routes
```

The routing table installs the route mentioned in the statement when the node fails.

3. Create a matching routing policy which refers the route as condition with the `route-exists` attribute.  
Example: Following configuration snippets show that you have configured the route with IP address 10.39.1.3 for SRGO as install on failure route. The routing policy statement includes the route 10.39.1.3 as the `if-route-exists` condition and the policy statement refers the condition as one of the matching term.

```
user@host# set policy-options condition active_route_exists if-route-exists address-family
inet 10.39.1.1/32
user@host# set policy-options condition active_route_exists if-route-exists address-family
inet table MNHA-signal-routes.inet.0
user@host# set policy-options condition backup_route_exists if-route-exists address-family
inet 10.39.1.2/32
user@host# set policy-options condition backup_route_exists if-route-exists address-family
inet table MNHA-signal-routes.inet.0
user@host# set policy-options condition failure_route_exists if-route-exists address-family
inet 10.39.1.3/32
```



```
user@host# set policy-options condition failure_route_exists if-route-exists address-family
inet table MNHA-signal-routes.inet.0
```

```
user@host# set policy-options policy-statement mnha-route-policy term 4 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 4 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 4 from condition
failure_route_exists
user@host# set policy-options policy-statement mnha-route-policy term 4 then metric 100
user@host# set policy-options policy-statement mnha-route-policy term 4 then accept
```

## Results (SRX-1)

From configuration mode, confirm your configuration by entering the following commands.

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis high-availability
local-id 1 local-ip 10.22.0.1;
peer-id 2 {
    peer-ip 10.22.0.2;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 0 {
    peer-id {
        2;
    }
}
services-redundancy-group 1 {
    deployment-type routing;
    peer-id {
        2;
    }
}
```



```

activeness-probe {
    dest-ip {
        10.111.0.1;
        src-ip 10.11.0.1;
    }
}
monitor {
    bfd-liveliness 10.4.0.2 {
        src-ip 10.4.0.1;
        session-type singlehop;
        interface ge-0/0/4.0;
    }
}
active-signal-route {
    10.39.1.1;
}
backup-signal-route {
    10.39.1.2;
}
preemption;
activeness-priority 200;
}

```

```

[edit]
user@host# show security ike
proposal MNHA_IKE_PROP {
    description mnha_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
policy MNHA_IKE_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IKE_PROP ;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway MNHA_IKE_GW {
    ike-policy MNHA_IKE_POL ;
    version v2-only;
}

```



```
}

```

```
[edit]
user@host# show security ipsec
proposal MNHA_IPSEC_PROP {
    description mnha_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3600;
}
policy MNHA_IPSEC_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IPSEC_PROP;
}
vpn IPSEC_VPN_ICL {
    ha-link-encryption;
    ike {
        gateway MNHA_IKE_GW;
        ipsec-policy MNHA_IPSEC_POL;
    }
}
```

```
[edit]
user@host# show policy-options
policy-statement mnha-route-policy {
    term 1 {
        from {
            protocol [ static direct ];
            condition active_route_exists;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 2 {
        from {
            protocol [ static direct ];
            condition backup_route_exists;

```



```

    }
    then {
        metric 20;
        accept;
    }
}
term 3 {
    from protocol [ static direct ];
    then {
        metric 30;
        accept;
    }
}
term default {
    then reject;
}
}
condition active_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}
}

```

[edit]

user@host# **show routing-options**



```

autonomous-system 100;
static {
    route 10.1.0.0/16 next-hop 10.2.0.1;
    route 10.6.0.0/16 next-hop 10.4.0.2;
    route 10.111.0.1/32 next-hop 10.2.0.1;
    route 10.111.0.2/32 next-hop 10.4.0.2;
}

```

```

[edit]
user@host# show security zones security-zone
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ike;
                ping;
            }
            protocols {
                bfd;
                bgp;
            }
        }
        interfaces {
            ge-0/0/4.0;
            lo0.0;
        }
    }
    security-zone trust {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            ge-0/0/3.0;
        }
    }
    security-zone halink {
        host-inbound-traffic {

```



```

        system-services {
            ike;
            ping;
            high-availability;
            ssh;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}

```

```

[edit]
user@host# show interfaces
ge-0/0/2 {
    description ha_link;
    unit 0 {
        family inet {
            address 10.22.0.1/24;
        }
    }
}
ge-0/0/3 {
    description trust;
    unit 0 {
        family inet {
            address 10.2.0.2/16;
        }
    }
}
ge-0/0/4 {
    description untrust;
    unit 0 {
        family inet {
            address 10.4.0.1/16;
        }
    }
}

```



```

    }
}
lo0 {
    description untrust;
    unit 0 {
        family inet {
            address 10.11.0.1/32;
            address 10.11.0.2/32;
            address 10.11.0.3/32;
        }
    }
}
}

```

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

## Results (SRX-2)

From configuration mode, confirm your configuration by entering the following commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show chassis high-availability
local-id 2 local-ip 10.22.0.2;
peer-id 1 {
    peer-ip 10.22.0.1;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 0 {
    peer-id {
        1;
    }
}
services-redundancy-group 1 {
    deployment-type routing;
    peer-id {
        1;
    }
}

```



```

}
activeness-probe {
    dest-ip {
        10.111.0.1;
        src-ip 10.11.0.1;
    }
}
monitor {
    bfd-liveliness 10.5.0.2 {
        src-ip 10.5.0.1;
        session-type singlehop;
        interface ge-0/0/4.0;
    }
}
active-signal-route {
    10.39.1.1;
}
backup-signal-route {
    10.39.1.2;
}
activeness-priority 1;
}

```

```

[edit]
user@host# show security ike
proposal MNHA_IKE_PROP {
    description mnha_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
policy MNHA_IKE_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IKE_PROP ;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway MNHA_IKE_GW {
    ike-policy MNHA_IKE_POL ;
}

```



```

    version v2-only;
}

```

```

[edit]
user@host# show security ipsec
proposal MNHA_IPSEC_PROP {
    description mnha_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3600;
}
policy MNHA_IPSEC_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IPSEC_PROP;
}
vpn IPSEC_VPN_ICL {
    ha-link-encryption;
    ike {
        gateway MNHA_IKE_GW;
        ipsec-policy MNHA_IPSEC_POL;
    }
}

```

```

[edit]
user@host# show policy-options

route-filter-list loopback {
    10.11.0.0/24 orlonger;
}
route-filter-list ipsec {
    10.6.0.0/16 orlonger;
}
policy-statement mnha-route-policy {
    term 1 {
        from {
            protocol [ static direct ];
            condition active_route_exists;
        }
        then {
            metric 10;

```



```

        accept;
    }
}
term 2 {
    from {
        protocol [ static direct ];
        condition backup_route_exists;
    }
    then {
        metric 20;
        accept;
    }
}
term 3 {
    from protocol [ static direct ];
    then {
        metric 30;
        accept;
    }
}
term default {
    then reject;
}
}
condition active_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}

```



```
    }
}
```

```
[edit]
user@host# show routing-options
autonomous-system 100;
static {
    route 10.1.0.0/16 next-hop 10.3.0.1;
    route 10.6.0.0/16 next-hop 10.5.0.2;
    route 10.111.0.1/32 next-hop 10.3.0.1;
    route 10.111.0.2/32 next-hop 10.5.0.2;
}
```

```
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/4.0;
        lo0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
}
```



```

    interfaces {
        ge-0/0/3.0;
    }
}
security-zone halink {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
            high-availability;
            ssh;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
}

```

```

[edit]
user@host# show interfaces
root@10.52.45.4# show interfaces
ge-0/0/2 {
    description ha_link;
    unit 0 {
        family inet {
            address 10.22.0.2/24;
        }
    }
}
ge-0/0/3 {
    description trust;
    unit 0 {
        family inet {
            address 10.3.0.2/16;
        }
    }
}
}

```



```

ge-0/0/4 {
  description untrust;
  unit 0 {
    family inet {
      address 10.5.0.1/16;
    }
  }
}
lo0 {
  description untrust;
  unit 0 {
    family inet {
      address 10.11.0.1/32;
      address 10.11.0.2/32;
      address 10.11.0.3/32;
    }
  }
}

```

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

On your security devices, you'll get the following message that asks you to reboot the device:

```

user@host# commit
warning: High Availability Mode changed, please reboot the device to avoid undesirable behavior
commit complete

```

## Verification

### IN THIS SECTION

- [Check Multinode High Availability Details | 80](#)
- [Check Multinode High Availability Peer Node Status | 82](#)
- [Check Multinode High Availability Service Redundancy Groups | 84](#)
- [Verify the Multinode High Availability Status Before and After Failover | 86](#)
- [Verify Interchassis Link \(ICL\) Encryption Status | 89](#)



- [Verify Link Encryption Tunnel Statistics | 90](#)
- [Verify Interchassis Link Active Peers | 91](#)

Confirm that the configuration is working properly.

## Check Multinode High Availability Details

### Purpose

View and verify the details of the Multinode High Availability setup configured on your security device.

### Action

From operational mode, run the following command:

On SRX-1

```
user@host> 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: 10.22.0.1
HA Peer Information:

  Peer Id: 2      IP address: 10.22.0.2    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 2
```



SRG failure event codes:

```
BF  BFD monitoring
IP  IP monitoring
IF  Interface monitoring
CP  Control Plane monitoring
```

Services Redundancy Group: 1

```
Deployment Type: ROUTING
Status: ACTIVE
Activeness Priority: 200
Preemption: ENABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 2
  Status : BACKUP
  Health Status: HEALTHY
  Failover Readiness: READY
```

## On SRX-2

```
user@host> 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: 10.22.0.2

HA Peer Information:

```
Peer Id: 1      IP address: 10.22.0.1   Interface: ge-0/0/2.0
Routing Instance: default
Encrypted: YES   Conn State: UP
Cold Sync Status: COMPLETE
```

Services Redundancy Group: 0

Current State: ONLINE



```

Peer Information:
  Peer Id: 1

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: ROUTING
  Status: BACKUP
  Activeness Priority: 1
  Preemption: DISABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: COMPLETE
  Failure Events: NONE
Peer Information:
  Peer Id: 1
  Status : ACTIVE
  Health Status: HEALTHY
  Failover Readiness: N/A

```

## Meaning

Verify these details from the command output:

- Local node and peer node details such as IP address and ID.
- The field Encrypted: YES indicates that the traffic is protected.
- The field Deployment Type: ROUTING indicates a Layer 3 mode configuration—that is, the network has routers on both sides.
- The field Services Redundancy Group: 1 indicates the status of the SRG1 (ACTIVE or BACKUP) on that node.

## Check Multinode High Availability Peer Node Status

### Purpose

View and verify the peer node details.



## Action

From operational mode, run the following command:

### SRX-1

```
user@host> user@host> show chassis high-availability peer-info
HA Peer Information:

Peer-ID: 2      IP address: 10.22.0.2      Interface: ge-0/0/2.0
Routing Instance: default
Encrypted: YES   Conn State: UP
Cold Sync Status: COMPLETE
Internal Interface: st0.16000
Internal Local-IP: 180.100.1.1
Internal Peer-IP: 180.100.1.2
Internal Routing-instance: __juniper_private1__
Packet Statistics:
    Receive Error : 0      Send Error : 0

    Packet-type      Sent      Received

    SRG Status Msg      4          4

    SRG Status Ack      4          3

    Attribute Msg      4          2

    Attribute Ack      2          2
```

### SRX-2

```
user@host> show chassis high-availability peer-info
HA Peer Information:

Peer-ID: 1      IP address: 10.22.0.1      Interface: ge-0/0/2.0
Routing Instance: default
Encrypted: YES   Conn State: UP
Cold Sync Status: COMPLETE
Internal Interface: st0.16000
```



```

Internal Local-IP: 180.100.1.2
Internal Peer-IP: 180.100.1.1
Internal Routing-instance: __juniper_private1__
Packet Statistics:
    Receive Error : 0      Send Error : 0

    Packet-type      Sent      Received

    SRG Status Msg   4         3

    SRG Status Ack    3         4

    Attribute Msg     3         2

    Attribute Ack     2         2

```

## Meaning

Verify these details from the command output:

- Peer node details such as interface used, IP address, and ID
- Encryption status, connection status, and cold synchronization status
- Packet statistics across the node.

## Check Multinode High Availability Service Redundancy Groups

### Purpose

Verify that the SRGs are configured and working correctly.

### Action

From operational mode, run the following command:



For SRG0:

```
user@host> show chassis high-availability services-redundancy-group 0
Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 2
```

For SRG1:

```
user@host> show chassis high-availability services-redundancy-group 1
SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: ROUTING
  Status: ACTIVE
  Activeness Priority: 200
  Preemption: ENABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: N/A
  Failure Events: NONE
  Peer Information:
    Peer Id: 2
    Status : BACKUP
    Health Status: HEALTHY
    Failover Readiness: READY

Signal Route Info:
  Active Signal Route:
    IP: 10.39.1.1
    Routing Instance: default
    Status: INSTALLED

  Backup Signal Route:
    IP: 10.39.1.2
    Routing Instance: default
```



```
Status: NOT INSTALLED
```

```
Split-brain Prevention Probe Info:
```

```
DST-IP: 10.111.0.1
```

```
SRC-IP: 10.11.0.1
```

```
Routing Instance: default
```

```
Status: NOT RUNNING
```

```
Result: N/A          Reason: N/A
```

```
BFD Monitoring:
```

```
Status: UP
```

```
SRC-IP: 10.4.0.1    DST-IP: 10.4.0.2
```

```
Routing Instance: default
```

```
Type: SINGLE-HOP
```

```
IFL Name: ge-0/0/4.0
```

```
State: UP
```

## Meaning

Verify these details from the command output:

- Peer node details such as deployment type, status, and active and back up signal routes.
- Virtual IP Information such as IP address and virtual MAC address.
- IP monitoring and BFD monitoring status.

## Verify the Multinode High Availability Status Before and After Failover

### Purpose

Check the change in node status before and after failover in a Multinode High Availability setup.

### Action

To check the Multinode High Availability status on the backup node (SRX-2), run the following command from operational mode:

```
user@host> 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: 10.22.0.2

HA Peer Information:

Peer Id: 1      IP address: 10.22.0.1      Interface: ge-0/0/2.0  
 Routing Instance: default  
 Encrypted: YES      Conn State: UP  
 Cold Sync Status: COMPLETE

Services Redundancy Group: 0

Current State: ONLINE

Peer Information:

Peer Id: 1

SRG failure event codes:

BF BFD monitoring  
 IP IP monitoring  
 IF Interface monitoring  
 CP Control Plane monitoring

Services Redundancy Group: 1

Deployment Type: ROUTING

Status: BACKUP

Activeness Priority: 1

Preemption: DISABLED

Process Packet In Backup State: NO

Control Plane State: READY

System Integrity Check: COMPLETE

Failure Events: NONE

Peer Information:

Peer Id: 1

Status : ACTIVE

Health Status: HEALTHY

Failover Readiness: N/A

Under the Services Redundancy Group: 1 section, you can see the Status: BACKUP field. This field value indicates that the status of SRG 1 is backup.



Initiate the failover on the active node (SRX-1 device) and again run the command on the backup node (SRX-2 device).

```

user@host> 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: 10.22.0.2
HA Peer Information:

    Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
    Routing Instance: default
    Encrypted: YES   Conn State: DOWN
    Cold Sync Status: IN PROGRESS

Services Redundancy Group: 0
    Current State: ONLINE
    Peer Information:
        Peer Id: 1

SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: ROUTING
    Status: ACTIVE
    Activeness Priority: 1
    Preemption: DISABLED
    Process Packet In Backup State: NO
    Control Plane State: READY
    System Integrity Check: N/A
    Failure Events: NONE
    Peer Information:
        Peer Id: 1
        Status : BACKUP

```



```
Health Status: HEALTHY
Failover Readiness: READY
```

Note that under the Services Redundancy Group: 1 section, the status of SRG1 has changed from **BACKUP** to **ACTIVE**.

You can also see peer node details under the Peer Information section. The output shows the status of peer as **BACKUP**.

## Verify Interchassis Link (ICL) Encryption Status

### Purpose

Verify the interchassis link (ICL) status.

### Action

From operational mode, run the following command:

```
user@host> show security ipsec security-associations ha-link-encryption detail
ID: 495001 Virtual-system: root, VPN Name: IPSEC_VPN_ICL
  Local Gateway: 10.22.0.1, Remote Gateway: 10.22.0.2
  Traffic Selector Name: __IPSEC_VPN_ICL__multi_node__
  Local Identity: ipv4(180.100.1.1-180.100.1.1)
  Remote Identity: ipv4(180.100.1.2-180.100.1.2)
  TS Type: traffic-selector
  Version: IKEv2
  PFS group: N/A
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.16000, Tunnel MTU: 0, Policy-
name: MNHA_IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  HA Link Encryption Mode: Multi-Node
  Location: FPC -, PIC -, KMD-Instance -
  Anchorship: Thread -
  Distribution-Profile: default-profile
  Direction: inbound, SPI: 0x0005a7ec, AUX-SPI: 0
               , VPN Monitoring: -
  Hard lifetime: Expires in 3597 seconds
  Lifesize Remaining: Unlimited
  Soft lifetime: Expires in 2900 seconds
  Mode: Tunnel(0 0), Type: dynamic, State: installed
```



```

Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
Anti-replay service: counter-based enabled, Replay window size: 64
Extended-Sequence-Number: Disabled
tunnel-establishment: establish-tunnels-immediately
Location: FPC 0, PIC 0, KMD-Instance 0
Anchorship: Thread 0
IKE SA Index: 4294966273
Direction: outbound, SPI: 0x000a2aba, AUX-SPI: 0
, VPN Monitoring: -
Hard lifetime: Expires in 3597 seconds
Lifesize Remaining: Unlimited
Soft lifetime: Expires in 2900 seconds
Mode: Tunnel(0 0), Type: dynamic, State: installed
Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
Anti-replay service: counter-based enabled, Replay window size: 64
Extended-Sequence-Number: Disabled
tunnel-establishment: establish-tunnels-immediately
Location: FPC 0, PIC 0, KMD-Instance 0
Anchorship: Thread 0
IKE SA Index: 4294966273

```

## Meaning

The command output provides the following information:

- The local gateway and remote gateway details.
- The IPsec SA pair for each threads in PIC.
- HA link encryption mode (as shown in the following line):

```
HA Link Encryption Mode: Multi-Node
```

- Authentication and encryption algorithms used

## Verify Link Encryption Tunnel Statistics

### Purpose

Verify link encryption tunnel statistics on both active and backup nodes.



## Action

From operational mode, run the following command:

```
user@host> show security ipsec statistics ha-link-encryption
ESP Statistics:
  Encrypted bytes:      984248
  Decrypted bytes:      462519
  Encrypted packets:    9067
  Decrypted packets:    8797
AH Statistics:
  Input bytes:          0
  Output bytes:         0
  Input packets:        0
  Output packets:       0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
  Invalid SPI: 0, TS check fail: 0
  Exceeds tunnel MTU: 0
  Discarded: 0
```

## Meaning

If you see packet loss issues across a VPN, you can run the `show security ipsec statistics ha-link-encryption` command several times to verify that the encrypted and decrypted packet counters are incrementing. You should also check whether the other error counters are incrementing.

Use the `clear security ipsec security-associations ha-link-encryption` command to clear all IPsec statistics.

## Verify Interchassis Link Active Peers

### Purpose

View only ICL active peers, but not regular IKE active peers.

### Action

From operational mode, run the following command:



## SRX-1

```
user@host> show security ike active-peer ha-link-encryption
```

Remote Address	Port	Peer IKE-ID	AAA username	Assigned IP
10.22.0.2	500	10.22.0.2	not available	0.0.0.0

## SRX-2

```
user@host> show security ike active-peer ha-link-encryption
```

Remote Address	Port	Peer IKE-ID	AAA username	Assigned IP
10.22.0.1	500	10.22.0.1	not available	0.0.0.0

**Meaning**

Command output displays only the active peer of the ICL with details such as the peer addresses and ports the active peer is using.

**SEE ALSO**

[Multinode High Availability | 2](#)

[Multinode High Availability Services | 43](#)

[Prepare Your Environment for Multinode High Availability Deployment | 40](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)



# Example: Configure Multinode High Availability in a Default Gateway Deployment

## SUMMARY

In this example, you'll establish Multinode High Availability between SRX Series devices in a default gateway (Layer 2 network) deployment.

## IN THIS SECTION

- [Overview | 93](#)
- [Requirements | 93](#)
- [Topology | 95](#)
- [Configuration | 97](#)
- [Verification | 122](#)

## Overview

In Multi-Node High Availability, participating SRX Series devices operate as independent nodes in a Layer 2 network. An encrypted logical interchassis link (ICL) connects the nodes over a routed network. Participating nodes backup each other to ensure a fast synchronized failover in case of system or hardware failure.

In Multinode High Availability, activeness is determined at the services redundancy group (SRG) level. The SRX Series device, on which the SRG1 is active, hosts the floating IP address and steers traffic towards it using the floating IP address. During a failover, the floating IP address moves from the old active node to the new active node and continues the communication client devices.

**NOTE:** As of Junos OS Release 22.3R1, we support a two-node configuration in the Multinode High Availability solution.

Lets start with an overview about the topology you'll be using in this example.

## Requirements

This example uses the following hardware and software components:



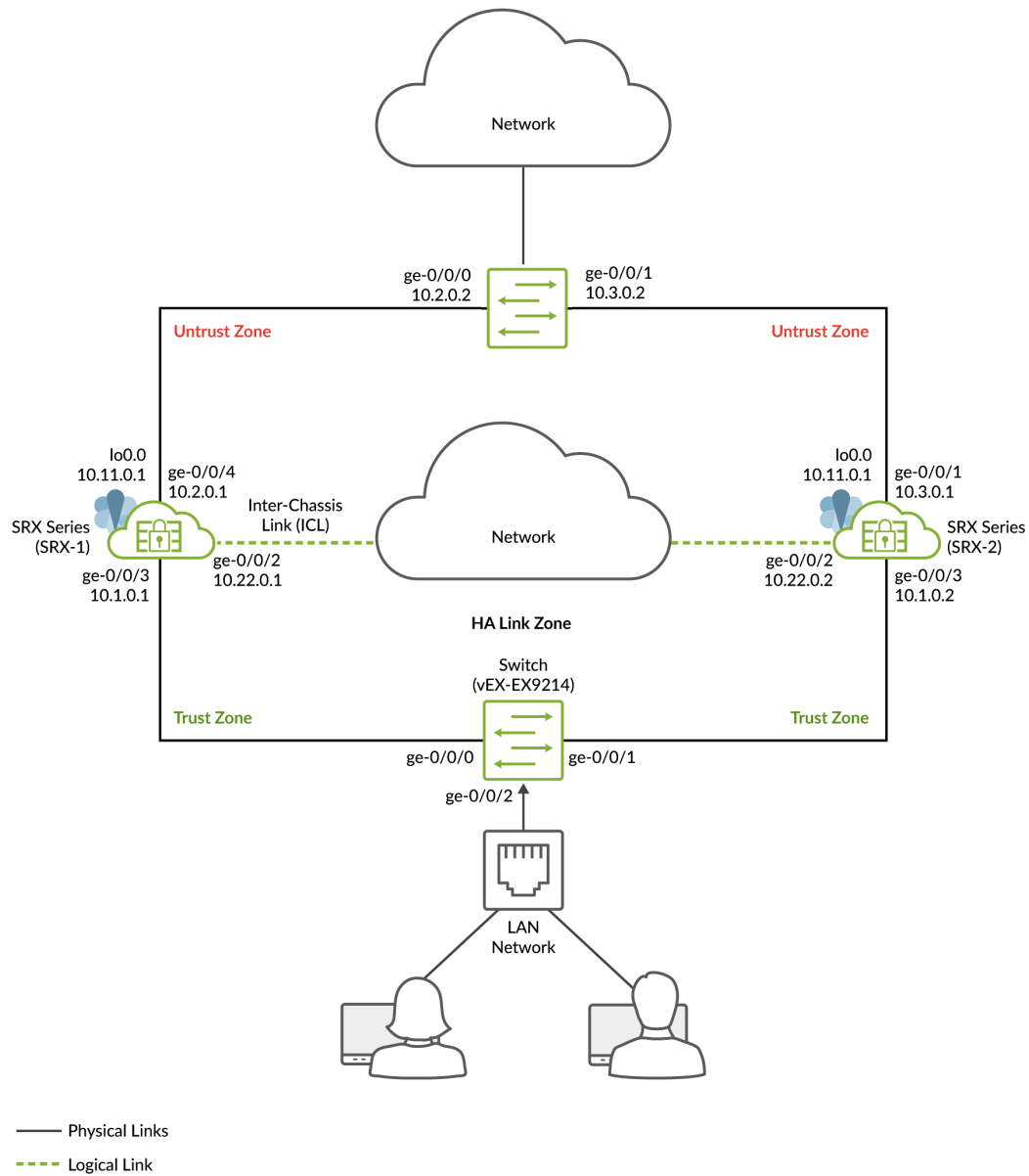
- Two SRX Series devices or vSRX instances
- Two Juniper Networks EX9214 Ethernet Switches
- Junos OS Release 22.3R1



## Topology

Figure 1 shows the topology used in this example.

**Figure 19: Multinode High Availability in Default Gateway Deployment**



As shown in the topology, two SRX Series devices are connected to switches on trust and untrust side. An encrypted logical interchassis link (ICL) connects the nodes over a routed network. The nodes



communicate with each other using a routable IP address (floating IP address) over the network. Loopback interfaces are used to host the IP addresses on SRX Series devices.

In general, you can use Aggregated Ethernet (AE) or a revenue Ethernet port on the SRX Series devices to setup an ICL connection. In this example, we've used GE ports for the ICL. We've also configured a routing instance for the ICL path to ensure maximum segmentation.

In a typical high availability deployment, you have multiple routers and switches on the northbound and southbound sides of the network. For this example, we are using two switches on both sides of SRX Series devices.

You'll perform the following tasks to build a Multinode High Availability setup:

- Configure a pair of SRX Series devices as local and peer nodes by assigning IDs.
- Configure services redundancy groups (SRGs).
- Configure a loopback interface (lo0.0) to host the floating IP address on the Layer 3 side.
- Configure virtual IP addresses for activeness determination and enforcement.
- Configure a signal route required for activeness enforcement and use it along with the route exists policy.
- Configure a VPN profile for the high availability (ICL) traffic using IKEv2.
- Configure BFD monitoring options.
- Configure a routing policy and routing options.
- Configure appropriate security policies to manage traffic in your network.
- Configure stateless firewall filtering and quality of service (QoS) as per your network requirements.
- Configure interfaces and zones according your network requirement. You must allow services such as IKE for link encryption and SSH for configuration synchronization as host inbound system services on the security zone that is associated with the ICL.

You can configure the following options on SRG0 and SRG1:

- SRG1: Active/backup signal route, deployment type, activeness priority, preemption, virtual IP address (for default gateway deployments), activeness probing and process packet on backup.
- SRG1: BFD monitoring, IP monitoring, and interface monitoring options on SRG1.
- SRG0: shutdown on failure and install on failure route options.

For interchassis link (ICL), we recommend the following configuration settings:



- Use a loopback (lo0) interface using an aggregated Ethernet interface (ae0), or any revenue Ethernet interface to establish the ICL.
- Set MTU of 1514
- Allow the following services on the security zone associated with interfaces used for ICL
  - IKE, high-availability, SSH
  - Protocols depends on the routing protocols you need
  - BFD to monitor the neighboring routes

## Configuration

### IN THIS SECTION

- [Before You Begin | 97](#)
- [CLI Quick Configuration | 98](#)
- [Configuration | 103](#)
- [Results \(SRX-1\) | 110](#)
- [Results \(SRX-2\) | 116](#)

## Before You Begin

Install the Junos IKE package on your SRX Series. You require this step for ICL encryption.

```
user@host> request system software add optional://junos-ike.tgz
```

```
Verified junos-ike signed by PackageProductionECP256_2022 method ECDSA256+SHA256
Rebuilding schema and Activating configuration...
mgd: commit complete
Restarting MGD ...
```

```
WARNING: cli has been replaced by an updated version:
CLI release 20220208.163814_builder.r1239105 built by builder on 2022-02-08 17:07:55 UTC
Restart cli using the new version ? [yes,no] (yes)
```



## CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, 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 may vary based on the specific requirements of your environment.

### On SRX-1 Device

```
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 10.22.0.1
set chassis high-availability peer-id 2 peer-ip 10.22.0.2
set chassis high-availability peer-id 2 interface ge-0/0/2.0
set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL
set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
set chassis high-availability peer-id 2 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 0 peer-id 2
set chassis high-availability services-redundancy-group 1 deployment-type switching
set chassis high-availability services-redundancy-group 1 peer-id 2
set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip 10.1.0.200/16
set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface ge-0/0/3.0
set chassis high-availability services-redundancy-group 1 virtual-ip 1 use-virtual-mac
set chassis high-availability services-redundancy-group 1 virtual-ip 2 ip 10.2.0.200/16
set chassis high-availability services-redundancy-group 1 virtual-ip 2 interface ge-0/0/4.0
set chassis high-availability services-redundancy-group 1 virtual-ip 2 use-virtual-mac
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/3
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/4
set chassis high-availability services-redundancy-group 1 preemption
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
set security ike proposal MNHA_IKE_PROP dh-group group14
set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
set security ike gateway MNHA_IKE_GW version v2-only
set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
set security ipsec proposal MNHA_IPSEC_PROP protocol esp
```



```

set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
set interfaces ge-0/0/3 description "trust" unit 0 family inet address 10.1.0.1/16
set interfaces ge-0/0/4 description "untrust" unit 0 family inet address 10.2.0.1/16
set interfaces ge-0/0/2 description "ha_link" unit 0 family inet address 10.22.0.1/24
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.1/32
set routing-options autonomous-system 100
set routing-options static route 10.2.0.0/16 next-hop 10.2.0.1
set routing-options static route 10.111.0.2 next-hop 10.2.0.1
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic protocols bfd
set security zones security-zone untrust host-inbound-traffic protocols bgp
set security zones security-zone untrust interfaces ge-0/0/4
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3
set security zones security-zone halink host-inbound-traffic system-services ike
set security zones security-zone halink host-inbound-traffic system-services ping
set security zones security-zone halink host-inbound-traffic system-services high-availability
set security zones security-zone halink host-inbound-traffic system-services ssh
set security zones security-zone halink host-inbound-traffic protocols bfd
set security zones security-zone halink host-inbound-traffic protocols bgp
set security zones security-zone halink interfaces ge-0/0/2
set security policies default-policy permit-all
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
set system syslog file vpn_syslog any info
set system syslog file vpn_syslog match "iked|pkid|kmd|ikemd|authd|jsrpd|chassisd|bfd"
set system services netconf ssh
set policy-options condition active_route_exists if-route-exists address-family inet 10.39.1.1
table inet.0
set policy-options condition backup_route_exists if-route-exists address-family inet 10.39.1.2
table inet.0
set policy-options policy-statement mnha-route-policy term 1 from protocol static
set policy-options policy-statement mnha-route-policy term 1 from protocol direct

```



```

set policy-options policy-statement mnha-route-policy term 1 from condition active_route_exists
set policy-options policy-statement mnha-route-policy term 1 then accept metric 10
set policy-options policy-statement mnha-route-policy term 2 from protocol static
set policy-options policy-statement mnha-route-policy term 2 from protocol direct
set policy-options policy-statement mnha-route-policy term 2 from condition backup_route_exists
set policy-options policy-statement mnha-route-policy term 2 then accept metric 20
set policy-options policy-statement mnha-route-policy term 3 from protocol static
set policy-options policy-statement mnha-route-policy term 3 from protocol direct
set policy-options policy-statement mnha-route-policy term 3 then accept metric 30
set policy-options policy-statement mnha-route-policy term default then reject
set protocols bgp group untrust type internal
set protocols bgp group untrust local-address 10.2.0.1
set protocols bgp group untrust export mnha-route-policy
set protocols bgp group untrust neighbor 10.2.0.1
set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group untrust bfd-liveness-detection multiplier 3
set protocols bgp group untrust local-as 100

```

### On SRX-2 Device

```

set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 10.22.0.2
set chassis high-availability peer-id 1 peer-ip 10.22.0.1
set chassis high-availability peer-id 1 interface ge-0/0/2.0
set chassis high-availability peer-id 1 vpn-profile IPSEC_VPN_ICL
set chassis high-availability peer-id 1 liveness-detection minimum-interval 400
set chassis high-availability peer-id 1 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 0 peer-id 1
set chassis high-availability services-redundancy-group 1 deployment-type switching
set chassis high-availability services-redundancy-group 1 peer-id 1
set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip 10.1.0.200/16
set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface ge-0/0/3.0
set chassis high-availability services-redundancy-group 1 virtual-ip 2 ip 10.2.0.200/16
set chassis high-availability services-redundancy-group 1 virtual-ip 2 interface ge-0/0/4.0
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/3
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/4
set chassis high-availability services-redundancy-group 1 activeness-priority 1
set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
set security ike proposal MNHA_IKE_PROP dh-group group14
set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256

```



```

set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
set security ike gateway MNHA_IKE_GW version v2-only
set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
set security ipsec proposal MNHA_IPSEC_PROP protocol esp
set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
set interfaces ge-0/0/3 description "trust" unit 0 family inet address 10.1.0.2/16
set interfaces ge-0/0/4 description "untrust" unit 0 family inet address 10.2.0.2/16
set interfaces ge-0/0/2 description "ha_link" unit 0 family inet address 10.22.0.2/24
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.1/32
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.2/32
set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.3/32
set routing-options autonomous-system 100
set routing-options static route 10.2.0.0/16 next-hop 10.2.0.2
set routing-options static route 10.111.0.2 next-hop 10.2.0.2
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic protocols bfd
set security zones security-zone untrust host-inbound-traffic protocols bgp
set security zones security-zone untrust interfaces ge-0/0/4
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3
set security zones security-zone halink host-inbound-traffic system-services ike
set security zones security-zone halink host-inbound-traffic system-services ping
set security zones security-zone halink host-inbound-traffic system-services high-availability
set security zones security-zone halink host-inbound-traffic system-services ssh
set security zones security-zone halink host-inbound-traffic protocols bfd
set security zones security-zone halink host-inbound-traffic protocols bgp
set security zones security-zone halink interfaces ge-0/0/2
set security policies default-policy permit-all
set security pki ca-profile Root-CA ca-identity Root-CA

```



```

set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
set system syslog file vpn_syslog any info
set system syslog file vpn_syslog match "iked|pkid|kmd|ikemd|authd|jsrpd|chassisd|bfd"
set system services netconf ssh
set policy-options route-filter-list loopback 10.11.0.0/24 orlonger
set policy-options route-filter-list ipsec 10.2.0.0/16 orlonger
set policy-options condition active_route_exists if-route-exists address-family inet 10.39.1.1
table inet.0
set policy-options condition backup_route_exists if-route-exists address-family inet 10.39.1.2
table inet.0
set policy-options policy-statement mnha-route-policy term 1 from protocol static
set policy-options policy-statement mnha-route-policy term 1 from protocol direct
set policy-options policy-statement mnha-route-policy term 1 from condition active_route_exists
set policy-options policy-statement mnha-route-policy term 1 then accept metric 10
set policy-options policy-statement mnha-route-policy term 2 from protocol static
set policy-options policy-statement mnha-route-policy term 2 from protocol direct
set policy-options policy-statement mnha-route-policy term 2 from condition backup_route_exists
set policy-options policy-statement mnha-route-policy term 2 then accept metric 20
set policy-options policy-statement mnha-route-policy term 3 from protocol static
set policy-options policy-statement mnha-route-policy term 3 from protocol direct
set policy-options policy-statement mnha-route-policy term 3 then accept metric 30
set policy-options policy-statement mnha-route-policy term default then reject
set protocols bgp group untrust type internal
set protocols bgp group untrust local-address 10.2.0.2
set protocols bgp group untrust export mnha-route-policy
set protocols bgp group untrust neighbor 10.2.0.2
set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group untrust bfd-liveness-detection multiplier 3
set protocols bgp group untrust local-as 100

```

The following sections show configuration snippets on the switches required for setting up Multinode High Availability setup in the network.

### On Switch (EX9214 Ethernet Switch)

```

set interfaces ge-0/0/2 description lan
set interfaces ge-0/0/2 mtu 9192
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lan
set interfaces ge-0/0/0 mtu 9192

```



```

set interfaces ge-0/0/0 description lan unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members lan
set interfaces ge-0/0/1 mtu 9192
set interfaces ge-0/0/1 description lan unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lan
set vlans lan vlan-id 1001

```

### On Switch (EX9214 Ethernet Switch)

```

set interfaces ge-0/0/2 description lan
set interfaces ge-0/0/2 mtu 9192
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lan
set interfaces ge-0/0/0 mtu 9192
set interfaces ge-0/0/0 description lan unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members lan
set interfaces ge-0/0/1 mtu 9192
set interfaces ge-0/0/1 description lan unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lan
set vlans lan vlan-id 1001

```

## Configuration

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the [CLI User Guide](#).

#### 1. Configure Interfaces.

```

[edit]
user@host# set interfaces ge-0/0/3 description "trust" unit 0 family inet address
10.1.0.1/16
user@host# set interfaces ge-0/0/4 description "untrust" unit 0 family inet address
10.2.0.1/16
user@host# set interfaces ge-0/0/2 description "ha_link" unit 0 family inet address
10.22.0.1/24

```



We're using the interfaces ge-0/0/3 and ge-0/0/4 to connect to the switches, and using the ge-0/0/2 interface for ICL.

## 2. Configure the loopback interface.

```
[edit]
user@host# set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.1/32
```

Assign the IP address (10.11.0.1) to the loopback interface. This IP address acts as the floating IP address.

Using the loopback interface ensures that at any given point, traffic from the adjacent devices will be steered toward the floating IP address (that is toward the active node).

## 3. Configure the security policy.

```
[edit]
user@host# set security policies default-policy permit-all
```

Ensure you have configured security policies as per your network requirements. In this example, you'll configure a policy to permit all traffic.

## 4. Configure security zones, assign interfaces to the zones, and specify the allowed system services for the security zones.

```
[edit]
user@host# set security zones security-zone untrust host-inbound-traffic system-services ike
user@host# set security zones security-zone untrust host-inbound-traffic system-services ping
user@host# set security zones security-zone untrust host-inbound-traffic protocols bfd
user@host# set security zones security-zone untrust host-inbound-traffic protocols bgp
user@host# set security zones security-zone untrust interfaces ge-0/0/4
user@host# set security zones security-zone untrust interfaces lo0.0
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces ge-0/0/3
user@host# set security zones security-zone halink host-inbound-traffic system-services ike
user@host# set security zones security-zone halink host-inbound-traffic system-services ping
user@host# set security zones security-zone halink host-inbound-traffic system-services high-availability
user@host# set security zones security-zone halink host-inbound-traffic system-services ssh
user@host# set security zones security-zone halink host-inbound-traffic protocols bfd
```



```

user@host# set security zones security-zone halink host-inbound-traffic protocols bgp
user@host# set security zones security-zone halink interfaces ge-0/0/2

```

Assign the interfaces ge-0/0/3 and ge-0/0/4 to the trust and untrust zones respectively. Assign the lo0.0 interface to the untrust zone to connect over the public IP network. Assign the interface ge-0/0/2 to the halink zone. You use this zone to setup the ICL.

## 5. Configure routing options.

```

[edit]
user@host# set routing-options autonomous-system 100
user@host# set routing-options static route 10.2.0.0/16 next-hop 10.2.0.1
user@host# set routing-options static route 10.111.0.2 next-hop 10.2.0.1

```

## 6. Configure both local node and peer node details such as node ID, IP addresses of the local node and peer node, and the interface for the peer node.

```

[edit]
user@host# set chassis high-availability local-id 1
user@host# set chassis high-availability local-id local-ip 10.22.0.1
user@host# set chassis high-availability peer-id 2 peer-ip 10.22.0.2
user@host# set chassis high-availability peer-id 2 interface ge-0/0/2.0

```

You'll use the ge-0/0/2 interface for communicating with the peer node using the ICL.

## 7. Attach the IPsec VPN profile IPSEC\_VPN\_ICL to the peer node.

```

[edit]
user@host# set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL

```

You'll need this configuration to establish a secure ICL link between the nodes.

## 8. Configure Bidirectional Forwarding Detection (BFD) protocol options for the peer node.

```

[edit]
user@host# set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
user@host# set chassis high-availability peer-id 2 liveness-detection multiplier 5

```



9. Associate the peer node ID 2 to the services redundancy group 0 (SRG0).

```
[edit]
user@host# set chassis high-availability services-redundancy-group 0 peer-id 2
```

10. Configure the services redundancy group 1 (SRG1).

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 deployment-type
switching
user@host# set chassis high-availability services-redundancy-group 1 peer-id 2
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip
10.1.0.200/16
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface
ge-0/0/3.0
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 1 use-
virtual-mac
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 2 ip
10.2.0.200/16
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 2 interface
ge-0/0/4.0
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 2 use-
virtual-mac
```

In this step, you are specifying the deployment type as switching because you are setting up Multinode High Availability as default gateway (Layer 2 network).

Assign a virtual IP (VIP) address and an interface for SRG1.

11. Configure IP and BFD monitoring parameters for SRG1 to check the reachability of an IP address and to detect failures in network.

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 monitor interface
ge-0/0/3
user@host# set chassis high-availability services-redundancy-group 1 monitor interface
ge-0/0/4
user@host# set chassis high-availability services-redundancy-group 1 preemption
```



12. Configure an active signal route required for activeness enforcement.

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 activeness-priority 200
```

The active signal route IP address you assign is used for route preference advertisement. You must specify the active signal route along with the route-exists policy in the policy-options statement.

13. Configure CA certificates as per your requirements.

```
[edit]
user@host# set security pki ca-profile Root-CA ca-identity Root-CA
user@host# set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/
mscep/mscep.dll
user@host# set security pki ca-profile Root-CA revocation-check disable
```

14. Define Internet Key Exchange (IKE) configuration for Multinode High Availability. An IKE configuration defines the algorithms and keys used to establish a secure connection.

```
[edit]
user@host# set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
user@host# set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
user@host# set security ike proposal MNHA_IKE_PROP dh-group group14
user@host# set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
user@host# set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
user@host# set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
user@host# set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
user@host# set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
user@host# set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
user@host# set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
user@host# set security ike gateway MNHA_IKE_GW version v2-only
```

For the Multinode High availability feature, you must configure the IKE version as v2-only

15. Specify the IPsec proposal protocol and encryption algorithm. Specify IPsec options to create a IPsec tunnel between two participant devices to secure VPN communication.

```
[edit]
user@host# set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
user@host# set security ipsec proposal MNHA_IPSEC_PROP protocol esp
```



```

user@host# set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
user@host# set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
user@host# set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
user@host# set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
user@host# set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
user@host# set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
user@host# set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL

```

Specifying the `ha-link-encryption` option encrypts the ICL to secure high availability traffic flow between the nodes.

The same VPN name `IPSEC_VPN_ICL` must be mentioned for *vpn\_profile* in chassis high availability configuration.

## 16. Configure policy options.

```

[edit]
user@host# set policy-options condition active_route_exists if-route-exists address-family
inet 10.39.1.1 table inet.0
user@host# set policy-options condition backup_route_exists if-route-exists address-family
inet 10.39.1.2 table inet.0
user@host# set policy-options policy-statement mnha-route-policy term 1 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 1 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists
user@host# set policy-options policy-statement mnha-route-policy term 1 then accept metric
10
user@host# set policy-options policy-statement mnha-route-policy term 2 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 2 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 2 from condition
backup_route_exists
user@host# set policy-options policy-statement mnha-route-policy term 2 then accept metric
20
user@host# set policy-options policy-statement mnha-route-policy term 3 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 3 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 3 then accept metric
30
user@host# set policy-options policy-statement mnha-route-policy term default then reject

```

Configure the active signal route 10.39.1.1 with the route match condition (if-route-exists). The Multinode High Availability module adds this route to the routing table when the node moves to the active role. The node also starts advertising the higher preference. Also configure the back up signal route (10.39.1.2) to advertise backup node with a medium priority. In case of any failures, the



high availability link goes down and the current active node releases its primary role and removes the active-signal-route,. Now the backup node detects the condition through its probes and transitions to the active role. The route preference is swapped to drive all the traffic to the new active node

**17. Configure BFD peering sessions options and specify liveness detection timers.**

```
[edit]
user@host# set protocols bgp group untrust type internal
user@host# set protocols bgp group untrust local-address 10.2.0.1
user@host# set protocols bgp group untrust export mnha-route-policy
user@host# set protocols bgp group untrust neighbor 10.2.0.1
user@host# set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
user@host# set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
user@host# set protocols bgp group untrust bfd-liveness-detection multiplier 3
user@host# set protocols bgp group untrust local-as 100
```

## Configuration Options for Software Upgrades

In Multinode High Availability, during software upgrade, you can divert the traffic by closing down interfaces on the node. Here, traffic cannot pass through the nodes. Check ["Software Upgrade in Multinode High Availability" on page 179](#) for details.

**1. Configure all traffic interfaces under “shutdown-on-failure” option.**

```
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
<interface-name>
```

Example:

```
[edit]
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
ge-0/0/3
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
ge-0/0/4
```



**CAUTION:** Donot use interfaces assigned for the interchassis link (ICL).



## Results (SRX-1)

From configuration mode, confirm your configuration by entering the following commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis high-availability
local-id 1 local-ip 10.22.0.1;
peer-id 2 {
    peer-ip 10.22.0.2;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 0 {
    peer-id {
        2;
    }
}
services-redundancy-group 1 {
    deployment-type switching;
    peer-id {
        2;
    }
    virtual-ip 1 {
        ip 10.1.0.200/16;
        interface ge-0/0/3.0;
        use-virtual-mac;
    }
    virtual-ip 2 {
        ip 10.2.0.200/16;
        interface ge-0/0/4.0;
        use-virtual-mac;
    }
}
monitor {
    interface {
        ge-0/0/3;
        ge-0/0/4;
```



```

    }
}
preemption;
activeness-priority 200;
}

```

```

[edit]
user@host# show security ike
proposal MNHA_IKE_PROP {
    description mnha_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
policy MNHA_IKE_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IKE_PROP ;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway MNHA_IKE_GW {
    ike-policy MNHA_IKE_POL ;
    version v2-only;
}

```

```

[edit]
user@host# show security ipsec
proposal MNHA_IPSEC_PROP {
    description mnha_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3600;
}
policy MNHA_IPSEC_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IPSEC_PROP;
}
vpn IPSEC_VPN_ICL {
    ha-link-encryption;
}

```



```

ike {
    gateway MNHA_IKE_GW;
    ipsec-policy MNHA_IPSEC_POL;
}

```

```

[edit]
user@host# show policy-options
policy-statement mnha-route-policy {
    term 1 {
        from {
            protocol [ static direct ];
            condition active_route_exists;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 2 {
        from {
            protocol [ static direct ];
            condition backup_route_exists;
        }
        then {
            metric 20;
            accept;
        }
    }
    term 3 {
        from protocol [ static direct ];
        then {
            metric 30;
            accept;
        }
    }
    term default {
        then reject;
    }
}
condition active_route_exists {

```



```

    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}
}

```

```

[edit]
user@host# show routing-options
autonomous-system 100;
static {
    route 10.2.0.0/16 next-hop 10.2.0.1;
    route 10.111.0.2/32 next-hop 10.2.0.1;
}

```

```

[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;

```



```

    }
}
interfaces {
    ge-0/0/4.0;
    lo0.0;
}
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone halink {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
            high-availability;
            ssh;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
}

```



```
[edit]
user@host# show interfaces

ge-0/0/2 {
  description ha_link;
  unit 0 {
    family inet {
      address 10.22.0.1/24;
    }
  }
}
ge-0/0/3 {
  description trust;
  unit 0 {
    family inet {
      address 10.1.0.1/16;
    }
  }
}
ge-0/0/4 {
  description untrust;
  unit 0 {
    family inet {
      address 10.2.0.1/16;
    }
  }
}
lo0 {
  description untrust;
  unit 0 {
    family inet {
      address 10.11.0.1/32;
    }
  }
}
```

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



## Results (SRX-2)

From configuration mode, confirm your configuration by entering the following commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis high-availability
local-id 2 local-ip 10.22.0.2;
peer-id 1 {
    peer-ip 10.22.0.1;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 0 {
    peer-id {
        1;
    }
}
services-redundancy-group 1 {
    deployment-type switching;
    peer-id {
        1;
    }
    virtual-ip 1 {
        ip 10.1.0.200/16;
        interface ge-0/0/3.0;
    }
    virtual-ip 2 {
        ip 10.2.0.200/16;
        interface ge-0/0/4.0;
    }
    monitor {
        interface {
            ge-0/0/3;
            ge-0/0/4;
        }
    }
}
```



```

    activeness-priority 1;
}

```

```

[edit]
user@host# show security ike
proposal MNHA_IKE_PROP {
    description mnha_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
policy MNHA_IKE_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IKE_PROP ;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway MNHA_IKE_GW {
    ike-policy MNHA_IKE_POL ;
    version v2-only;
}

```

```

[edit]
user@host# show security ipsec

proposal MNHA_IPSEC_PROP {
    description mnha_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3600;
}
policy MNHA_IPSEC_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IPSEC_PROP;
}
vpn IPSEC_VPN_ICL {
    ha-link-encryption;
    ike {
        gateway MNHA_IKE_GW;
    }
}

```



```

        ipsec-policy MNHA_IPSEC_POL;
    }
}

```

```

[edit]
user@host# show policy-options
route-filter-list loopback {
    10.11.0.0/24 orlonger;
}
route-filter-list ipsec {
    10.2.0.0/16 orlonger;
}
policy-statement mnha-route-policy {
    term 1 {
        from {
            protocol [ static direct ];
            condition active_route_exists;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 2 {
        from {
            protocol [ static direct ];
            condition backup_route_exists;
        }
        then {
            metric 20;
            accept;
        }
    }
    term 3 {
        from protocol [ static direct ];
        then {
            metric 30;
            accept;
        }
    }
    term default {

```



```

        then reject;
    }
}
condition active_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}
}

```

```

[edit]
user@host# show routing-options
autonomous-system 100;
static {
    route 10.2.0.0/16 next-hop 10.2.0.2;
    route 10.111.0.2/32 next-hop 10.2.0.2;
}

```

```

[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
    }
}

```



```

        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/4.0;
        lo0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone halink {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
            high-availability;
            ssh;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}

```



```
}
```

```
[edit]
```

```
user@host# show interfaces
```

```
ge-0/0/2 {
  description ha_link;
  unit 0 {
    family inet {
      address 10.22.0.2/24;
    }
  }
}
ge-0/0/3 {
  description trust;
  unit 0 {
    family inet {
      address 10.1.0.2/16;
    }
  }
}
ge-0/0/4 {
  description untrust;
  unit 0 {
    family inet {
      address 10.2.0.2/16;
    }
  }
}
lo0 {
  description untrust;
  unit 0 {
    family inet {
      address 10.11.0.1/32;
      address 10.11.0.2/32;
      address 10.11.0.3/32;
    }
  }
}
```



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

On your security devices, you'll get the following message that asks you to reboot the device:

```
user@host# commit
warning: High Availability Mode changed, please reboot the device to avoid undesirable behavior
commit complete
```

## Verification

### IN THIS SECTION

- [Check Multinode High Availability Details | 122](#)
- [Check Multinode High Availability Peer Node Status | 125](#)
- [Check Multinode High Availability Service Redundancy Groups | 127](#)
- [Verify the Multinode High Availability Status Before and After Failover | 129](#)
- [Verify Interchassis Link \(ICL\) Encryption Status | 131](#)
- [Verify Link Encryption Tunnel Statistics | 133](#)

Confirm that the configuration is working properly.

### Check Multinode High Availability Details

#### Purpose

View and verify the details of the Multinode High Availability setup configured on your security device.

#### Action

From operational mode, run the following command:



## On SRX-1

```

user@host> 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: 10.22.0.1
HA Peer Information:

  Peer Id: 2      IP address: 10.22.0.2    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 2

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: SWITCHING
  Status: ACTIVE
  Activeness Priority: 200
  Preemption: ENABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: N/A
  Failure Events: NONE
  Peer Information:
    Peer Id: 2
    Status : BACKUP

```



Health Status: HEALTHY  
Failover Readiness: READY

## On SRX-2

```
user@host> 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: 10.22.0.2

HA Peer Information:

Peer Id: 1	IP address: 10.22.0.1	Interface: ge-0/0/2.0
Routing Instance: default		
Encrypted: YES	Conn State: UP	
Cold Sync Status: COMPLETE		

Services Redundancy Group: 0

Current State: ONLINE

Peer Information:

Peer Id: 1

SRG failure event codes:

BF	BFD monitoring
IP	IP monitoring
IF	Interface monitoring
CP	Control Plane monitoring

Services Redundancy Group: 1

Deployment Type: SWITCHING

Status: BACKUP

Activeness Priority: 1

Preemption: DISABLED

Process Packet In Backup State: NO

Control Plane State: READY

System Integrity Check: COMPLETE

Failure Events: NONE

Peer Information:



```

Peer Id: 1
Status : ACTIVE
Health Status: HEALTHY
Failover Readiness: N/A

```

## Meaning

Verify these details from the command output:

- Local node and peer node details such as IP address and ID.
- The field Encrypted: YES indicates that the traffic is protected.
- The field Deployment Type: SWITCHING indicates a default gateway (switching) mode configuration—that is, the network has switches connected at both ends (Layer 2 network).
- The field Services Redundancy Group: 1 indicates the status of the SRG1 (ACTIVE or BACKUP) on that node.

## Check Multinode High Availability Peer Node Status

### Purpose

View and verify the peer node details.

### Action

From operational mode, run the following command:

SRX-1

```

user@host> show chassis high-availability peer-info
HA Peer Information:

Peer-ID: 2      IP address: 10.22.0.2      Interface: ge-0/0/2.0
Routing Instance: default
Encrypted: YES  Conn State: UP
Cold Sync Status: COMPLETE
Internal Interface: st0.16000
Internal Local-IP: 180.100.1.1
Internal Peer-IP: 180.100.1.2
Internal Routing-instance: __juniper_private1__

```



## Packet Statistics:

Receive Error : 0

Send Error : 0

Packet-type	Sent	Received
-------------	------	----------

SRG Status Msg	3	4
----------------	---	---

SRG Status Ack	4	3
----------------	---	---

Attribute Msg	3	2
---------------	---	---

Attribute Ack	2	2
---------------	---	---

## SRX-2

user@host> **show chassis high-availability peer-info**

## HA Peer Information:

Peer-ID: 1      IP address: 10.22.0.1      Interface: ge-0/0/2.0

Routing Instance: default

Encrypted: YES      Conn State: UP

Cold Sync Status: COMPLETE

Internal Interface: st0.16000

Internal Local-IP: 180.100.1.2

Internal Peer-IP: 180.100.1.1

Internal Routing-instance: \_\_juniper\_private1\_\_

## Packet Statistics:

Receive Error : 0

Send Error : 0

Packet-type	Sent	Received
-------------	------	----------

SRG Status Msg	10	8
----------------	----	---

SRG Status Ack	8	8
----------------	---	---

Attribute Msg	8	4
---------------	---	---

Attribute Ack	4	4
---------------	---	---



## Meaning

Verify these details from the command output:

- Peer node details such as interface used, IP address, and ID.
- Encryption status, connection status, and cold synchronization status
- Packet statistics across the node.

## Check Multinode High Availability Service Redundancy Groups

### Purpose

Verify that the SRGs are configured and working correctly.

### Action

From operational mode, run the following command:

For SRG0:

```
user@host> show chassis high-availability services-redundancy-group 0

Services Redundancy Group: 0
    Current State: ONLINE
    Peer Information:
        Peer Id: 2
```

For SRG1:

```
user@host> show chassis high-availability services-redundancy-group 1 >

SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: SWITCHING
    Status: ACTIVE
```



```

Activeness Priority: 200
Preemption: ENABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 2
  Status : BACKUP
  Health Status: HEALTHY
  Failover Readiness: READY

```

```

Virtual IP Info:
  Index: 2
  IP: 10.2.0.200/16
  VMAC: N/A
  Interface: ge-0/0/4.0
  Status: INSTALLED

```

```

  Index: 1
  IP: 10.1.0.200/16
  VMAC: N/A
  Interface: ge-0/0/3.0
  Status: INSTALLED

```

```

Split-brain Prevention Probe Info:
  DST-IP: 10.1.0.200
  Routing Instance: default
  Status: NOT RUNNING
  Result: N/A          Reason: N/A

```

```

Interface Monitoring:
  Status: UP

```

```

  IF Name: ge-0/0/4    State: Up

```

```

  IF Name: ge-0/0/3    State: Up

```

## Meaning

Verify these details from the command output:



- Peer node details such as deployment type, status, and active and back up signal routes.
- Virtual IP Information such as IP address and virtual MAC address.
- IP monitoring and BFD monitoring status.

## Verify the Multinode High Availability Status Before and After Failover

### Purpose

Check the change in node status before and after failover in a Multinode High Availability setup.

### Action

To check the Multinode High Availability status on the backup node (SRX-2), run the following command from operational mode:

```
user@host> 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: 10.22.0.2
HA Peer Information:

    Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
    Routing Instance: default
    Encrypted: YES   Conn State: UP
    Cold Sync Status: COMPLETE

Services Redundancy Group: 0
    Current State: ONLINE
    Peer Information:
        Peer Id: 1

SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
```



```
IF Interface monitoring
CP Control Plane monitoring
```

```
Services Redundancy Group: 1
  Deployment Type: SWITCHING
  Status: BACKUP
  Activeness Priority: 1
  Preemption: DISABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: COMPLETE
  Failure Events: NONE
  Peer Information:
    Peer Id: 1
    Status : ACTIVE
    Health Status: HEALTHY
    Failover Readiness: N/A
```

Under the Services Redundancy Group: 1 section, you can see the Status: BACKUP field. This field value indicates that the status of SRG 1 is backup.

Initiate the failover on the active node (SRX-1 device) and again run the command on the backup node (SRX-2).

```
user@host> 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: DOWN
  Cold Sync Status: IN PROGRESS

Services Redundancy Group: 0
  Current State: ONLINE
```



```

Peer Information:
  Peer Id: 1

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: SWITCHING
  Status: ACTIVE
  Activeness Priority: 1
  Preemption: DISABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: N/A
  Failure Events: NONE
  Peer Information:
    Peer Id: 1
    Status : BACKUP
    Health Status: HEALTHY
    Failover Readiness: READY

```

Note that under the Services Redundancy Group: 1 section, the status of SRG1 has changed from **BACKUP** to **ACTIVE**.

You can also see peer node details under the Peer Information section. The output shows the status of peer as **BACKUP**.

## Verify Interchassis Link (ICL) Encryption Status

### Purpose

Verify the interchassis link (ICL) status.

### Action

From operational mode, run the following command:

```

user@host> show security ipsec security-associations ha-link-encryption detail
ID: 495002 Virtual-system: root, VPN Name: IPSEC_VPN_ICL

```



```

Local Gateway: 10.22.0.1, Remote Gateway: 10.22.0.2
Traffic Selector Name: __IPSEC_VPN_ICL__multi_node__
Local Identity: ipv4(180.100.1.1-180.100.1.1)
Remote Identity: ipv4(180.100.1.2-180.100.1.2)
TS Type: traffic-selector
Version: IKEv2
PFS group: N/A
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.16000, Tunnel MTU: 0, Policy-
name: MNHA_IPSEC_POL
Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
HA Link Encryption Mode: Multi-Node
Location: FPC -, PIC -, KMD-Instance -
Anchorship: Thread -
Distribution-Profile: default-profile
Direction: inbound, SPI: 0x000afc7f, AUX-SPI: 0
                , VPN Monitoring: -
    Hard lifetime: Expires in 1888 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 1248 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    Location: FPC 0, PIC 0, KMD-Instance 0
    Anchorship: Thread 0
    IKE SA Index: 4294966274
Direction: outbound, SPI: 0x000079a0, AUX-SPI: 0
                , VPN Monitoring: -
    Hard lifetime: Expires in 1888 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 1248 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    Location: FPC 0, PIC 0, KMD-Instance 0
    Anchorship: Thread 0
    IKE SA Index: 4294966274

```



## Meaning

The command output provides the following information:

- The local gateway and remote gateway details.
- The IPsec SA pair for each threads in PIC.
- HA link encryption mode (as shown in the following line):

```
HA Link Encryption Mode: Multi-Node
```

- Authentication and encryption algorithms used

## Verify Link Encryption Tunnel Statistics

### Purpose

Verify link encryption tunnel statistics on both active and backup nodes.

### Action

From operational mode, run the following command:

```
user@host> show security ipsec statistics ha-link-encryption
```

ESP Statistics:

Encrypted bytes: 2455540

Decrypted bytes: 1186957

Encrypted packets: 22673

Decrypted packets: 22694

AH Statistics:

Input bytes: 0

Output bytes: 0

Input packets: 0

Output packets: 0

Errors:

AH authentication failures: 0, Replay errors: 0

ESP authentication failures: 0, ESP decryption failures: 0

Bad headers: 0, Bad trailers: 0

Invalid SPI: 0, TS check fail: 0



```
Exceeds tunnel MTU: 0
Discarded: 0
```

## Meaning

If you see packet loss issues across a VPN, you can run the `show security ipsec statistics ha-link-encryption` command several times to verify that the encrypted and decrypted packet counters are incrementing. You should also check whether the other error counters are incrementing.

Use the `show security ike active-peer ha-link-encryption` command to display details of ICL on the active peer node.

Use the `clear security ipsec statistics ha-link-encryption` command to clear all IPsec statistics.

## SEE ALSO

[Multinode High Availability | 2](#)

[Prepare Your Environment for Multinode High Availability Deployment | 40](#)

[Multinode High Availability Services | 43](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)

# Example: Configure Multinode High Availability in a Hybrid Deployment

## SUMMARY

Read this topic to learn how to configure Multinode High Availability solution on SRX Series devices. The example covers configuration in active/backup mode when SRX Series devices are connected to a router on one side and switch on the other side.

## IN THIS SECTION

- [Overview | 135](#)
- [Requirements | 135](#)
- [Topology | 136](#)
- [Configuration | 138](#)



## Overview

In a hybrid deployments, participating SRX Series devices operate as independent nodes in a mixed mode of routed networks on one side and locally connected networks on the other side. An encrypted logical interchassis link (ICL) connects the nodes over a routed network.

In Multinode High Availability, activeness is determined at the services redundancy group (SRG) level. The SRX Series device, on which the SRG1 is active, hosts the floating IP address and steers traffic towards it using the floating IP address. During a failover, the floating IP address moves from the old active node to the new active node and continues the communication client devices.

**NOTE:** As of Junos OS Release 22.3R1, we support a two-node configuration in the Multinode High Availability solution.

In this example, you'll establish high availability between the SRX Series devices and secure the tunnel traffic by enabling HA link encryption.

## Requirements

This example uses the following hardware and software components:

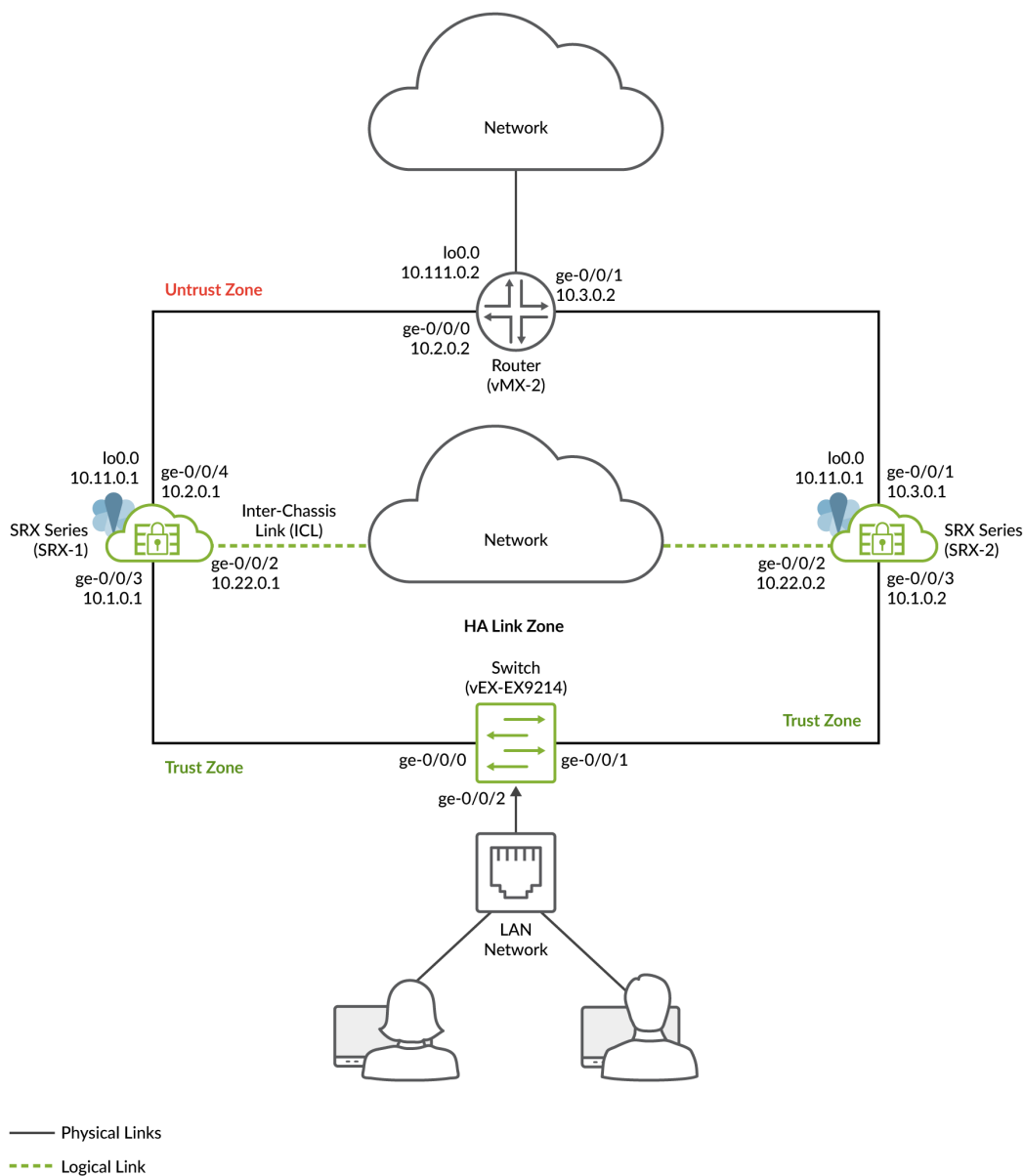
- Two SRX Series devices or vSRX Instances
- A Juniper Networks(R) MX960 Universal Routing Platform at one end
- A Juniper Networks(R) EX9214 Ethernet Switch at the other end
- Junos OS Release 22.3R1



## Topology

Figure 1 shows the topology used in this example.

Figure 20: Multinode High Availability In Hybrid Network



As shown in the topology, two SRX Series devices connected to routers on untrust side and to a switch trust side of the network. The nodes communicate with each other using a routable IP address (floating



IP address) over the network. Loopback interfaces are used to host the IP addresses on SRX Series and upstream router.

In general, you can use Aggregated Ethernet (AE) or a revenue Ethernet port on the SRX Series devices to setup an ICL connection. In this example, we've used GE ports for the ICL. We've also configured a routing instance for the ICL path to ensure maximum segmentation.

In a typical high availability deployment, you have multiple routers and switches on the northbound and southbound sides of the network. For this example, we are using one routers and one switch.

You'll perform the following tasks to build a Multinode High Availability setup:

- Configure a pair of SRX Series devices as local and peer nodes by assigning IDs.
- Configure services redundancy groups (SRGs).
- Configure a loopback interface (lo0.0) to host a floating IP address on the Layer 3 side.
- Configure virtual IP addresses for activeness determination and enforcement on the Layer 2 side.
- Configure a signal route required for activeness enforcement and use it along with the route exists policy.
- Configure a VPN profile for the high availability (ICL) traffic using IKEv2.
- Configure BFD monitoring options.
- Configure a routing policy and routing options.
- Configure appropriate security policies to manage traffic in your network.
- Configure stateless firewall filtering and quality of service (QoS) as per your network requirements.
- Configure interfaces and zones according to your network requirement. You must allow services such as IKE for link encryption and SSH for configuration synchronization as host-inbound system services on the security zone that is associated with the ICL.

You can configure the following options on SRG0 and SRG1:

- SRG1: Active/backup signal route, deployment type, activeness priority, preemption, virtual IP address (for default gateway deployments), activeness probing and process packet on backup.
- SRG1: BFD monitoring, IP monitoring, and interface monitoring options on SRG1.
- SRG0: shutdown on failure and install on failure route options.

For interchassis link (ICL), we recommend the following configuration settings:

- Use a loopback (lo0) interface using an aggregated Ethernet interface (ae0), or any revenue Ethernet interface to establish the ICL.



- Set MTU of 1514
- Allow the following services on the security zone associated with interfaces used for ICL
  - IKE, high-availability, SSH
  - Protocols depends on routing protocol you need
  - BFD to monitor the neighboring routes

## Configuration

### IN THIS SECTION

- [Before You Begin | 138](#)
- [CLI Quick Configuration | 139](#)
- [Configuration | 145](#)
- [Results \(SRX-1\) | 152](#)
- [Results \(SRX-2\) | 158](#)

## Before You Begin

Install the Junos IKE package on your SRX Series. You require this step for ICL encryption.

```
user@host> request system software add optional://junos-ike.tgz
Verified junos-ike signed by PackageProductionECP256_2022 method ECDSA256+SHA256
Rebuilding schema and Activating configuration...
mgd: commit complete
Restarting MGD ...

WARNING: cli has been replaced by an updated version:
CLI release 20220208.163814_builder.r1239105 built by builder on 2022-02-08 17:07:55 UTC
Restart cli using the new version ? [yes,no] (yes)
```



## CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, 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 may vary based on the specific requirements of your environment.

### On SRX-1 Device

```

set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 10.22.0.1
set chassis high-availability peer-id 2 peer-ip 10.22.0.2
set chassis high-availability peer-id 2 interface ge-0/0/2.0
set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL
set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
set chassis high-availability peer-id 2 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 0 peer-id 2
set chassis high-availability services-redundancy-group 1 deployment-type hybrid
set chassis high-availability services-redundancy-group 1 peer-id 2
set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip 10.1.0.200/16
set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface ge-0/0/3.0
set chassis high-availability services-redundancy-group 1 virtual-ip 1 use-virtual-mac
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.2.0.1 src-ip 10.2.0.2
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.2.0.1
session-type singlehop
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.2.0.1
interface ge-0/0/4.0
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/3
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/4
set chassis high-availability services-redundancy-group 1 active-signal-route 10.39.1.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 10.39.1.2
set chassis high-availability services-redundancy-group 1 preemption
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
set security ike proposal MNHA_IKE_PROP dh-group group14
set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc

```



```

set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
set security ike gateway MNHA_IKE_GW version v2-only
set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
set security ipsec proposal MNHA_IPSEC_PROP protocol esp
set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic protocols bfd
set security zones security-zone untrust host-inbound-traffic protocols bgp
set security zones security-zone untrust interfaces ge-0/0/4.0
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security zones security-zone halink host-inbound-traffic system-services ike
set security zones security-zone halink host-inbound-traffic system-services ping
set security zones security-zone halink host-inbound-traffic system-services high-availability
set security zones security-zone halink host-inbound-traffic system-services ssh
set security zones security-zone halink host-inbound-traffic protocols bfd
set security zones security-zone halink host-inbound-traffic protocols bgp
set security zones security-zone halink interfaces ge-0/0/2.0
set interfaces ge-0/0/2 description ha_link
set interfaces ge-0/0/2 unit 0 family inet address 10.22.0.1/24
set interfaces ge-0/0/3 description trust
set interfaces ge-0/0/3 unit 0 family inet address 10.1.0.1/16
set interfaces ge-0/0/4 description untrust
set interfaces ge-0/0/4 unit 0 family inet address 10.2.0.1/16
set interfaces lo0 description untrust
set interfaces lo0 unit 0 family inet address 10.11.0.1/32
set interfaces lo0 unit 0 family inet address 10.11.0.2/32
set interfaces lo0 unit 0 family inet address 10.11.0.3/32
set policy-options policy-statement mnha-route-policy term 1 from protocol static

```



```

set policy-options policy-statement mnha-route-policy term 1 from protocol direct
set policy-options policy-statement mnha-route-policy term 1 from condition active_route_exists
set policy-options policy-statement mnha-route-policy term 1 then metric 10
set policy-options policy-statement mnha-route-policy term 1 then accept
set policy-options policy-statement mnha-route-policy term 2 from protocol static
set policy-options policy-statement mnha-route-policy term 2 from protocol direct
set policy-options policy-statement mnha-route-policy term 2 from condition backup_route_exists
set policy-options policy-statement mnha-route-policy term 2 then metric 20
set policy-options policy-statement mnha-route-policy term 2 then accept
set policy-options policy-statement mnha-route-policy term 3 from protocol static
set policy-options policy-statement mnha-route-policy term 3 from protocol direct
set policy-options policy-statement mnha-route-policy term 3 then metric 30
set policy-options policy-statement mnha-route-policy term 3 then accept
set policy-options policy-statement mnha-route-policy term default then reject
set policy-options condition active_route_exists if-route-exists address-family inet 10.39.1.1/32
set policy-options condition active_route_exists if-route-exists address-family inet table inet.0
set policy-options condition backup_route_exists if-route-exists address-family inet 10.39.1.2/32
set policy-options condition backup_route_exists if-route-exists address-family inet table inet.0
set protocols bgp group untrust type internal
set protocols bgp group untrust local-address 10.2.0.1
set protocols bgp group untrust export mnha-route-policy
set protocols bgp group untrust local-as 100
set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group untrust bfd-liveness-detection multiplier 3
set protocols bgp group untrust neighbor 10.2.0.2
set routing-options autonomous-system 100
set routing-options static route 10.4.0.0/16 next-hop 10.2.0.2
set routing-options static route 10.111.0.2/32 next-hop 10.2.0.2

```

## On SRX-2 Device

```

set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 10.22.0.2
set chassis high-availability peer-id 1 peer-ip 10.22.0.1
set chassis high-availability peer-id 1 interface ge-0/0/2.0
set chassis high-availability peer-id 1 vpn-profile IPSEC_VPN_ICL
set chassis high-availability peer-id 1 liveness-detection minimum-interval 400
set chassis high-availability peer-id 1 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 0 peer-id 1
set chassis high-availability services-redundancy-group 1 deployment-type hybrid
set chassis high-availability services-redundancy-group 1 peer-id 1

```



```

set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip 10.1.0.200/16
set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface ge-0/0/3.0
set chassis high-availability services-redundancy-group 1 virtual-ip 1 use-virtual-mac
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.3.0.1 src-ip
10.3.0.2
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.3.0.1
session-type singlehop
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.3.0.1
interface ge-0/0/4.0
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/3
set chassis high-availability services-redundancy-group 1 monitor interface ge-0/0/4
set chassis high-availability services-redundancy-group 1 active-signal-route 10.39.1.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 10.39.1.2
set chassis high-availability services-redundancy-group 1 activeness-priority 1
set security pki ca-profile Root-CA ca-identity Root-CA
set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/mscep/mscep.dll
set security pki ca-profile Root-CA revocation-check disable
set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
set security ike proposal MNHA_IKE_PROP dh-group group14
set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
set security ike gateway MNHA_IKE_GW version v2-only
set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
set security ipsec proposal MNHA_IPSEC_PROP protocol esp
set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone untrust host-inbound-traffic system-services ike
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic protocols bfd
set security zones security-zone untrust host-inbound-traffic protocols bgp
set security zones security-zone untrust interfaces ge-0/0/4.0

```



```

set security zones security-zone untrust interfaces lo0.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/3.0
set security zones security-zone halink host-inbound-traffic system-services ike
set security zones security-zone halink host-inbound-traffic system-services ping
set security zones security-zone halink host-inbound-traffic system-services high-availability
set security zones security-zone halink host-inbound-traffic system-services ssh
set security zones security-zone halink host-inbound-traffic protocols bfd
set security zones security-zone halink host-inbound-traffic protocols bgp
set security zones security-zone halink interfaces ge-0/0/2.0
set interfaces ge-0/0/2 description ha_link
set interfaces ge-0/0/2 unit 0 family inet address 10.22.0.2/24
set interfaces ge-0/0/3 description trust
set interfaces ge-0/0/3 unit 0 family inet address 10.1.0.2/16
set interfaces ge-0/0/4 description untrust
set interfaces ge-0/0/4 unit 0 family inet address 10.3.0.1/16
set interfaces lo0 description untrust
set interfaces lo0 unit 0 family inet address 10.11.0.1/32
set interfaces lo0 unit 0 family inet address 10.11.0.2/32
set interfaces lo0 unit 0 family inet address 10.11.0.3/32
set policy-options route-filter-list loopback 10.11.0.0/24 orlonger
set policy-options route-filter-list ipsec 10.4.0.0/16 orlonger
set policy-options policy-statement mnha-route-policy term 1 from protocol static
set policy-options policy-statement mnha-route-policy term 1 from protocol direct
set policy-options policy-statement mnha-route-policy term 1 from condition active_route_exists
set policy-options policy-statement mnha-route-policy term 1 then metric 10
set policy-options policy-statement mnha-route-policy term 1 then accept
set policy-options policy-statement mnha-route-policy term 2 from protocol static
set policy-options policy-statement mnha-route-policy term 2 from protocol direct
set policy-options policy-statement mnha-route-policy term 2 from condition backup_route_exists
set policy-options policy-statement mnha-route-policy term 2 then metric 20
set policy-options policy-statement mnha-route-policy term 2 then accept
set policy-options policy-statement mnha-route-policy term 3 from protocol static
set policy-options policy-statement mnha-route-policy term 3 from protocol direct
set policy-options policy-statement mnha-route-policy term 3 then metric 30
set policy-options policy-statement mnha-route-policy term 3 then accept
set policy-options policy-statement mnha-route-policy term default then reject
set policy-options condition active_route_exists if-route-exists address-family inet 10.39.1.1/32
set policy-options condition active_route_exists if-route-exists address-family inet table inet.0
set policy-options condition backup_route_exists if-route-exists address-family inet 10.39.1.2/32
set policy-options condition backup_route_exists if-route-exists address-family inet table inet.0
set protocols bgp group untrust type internal

```



```

set protocols bgp group untrust local-address 10.3.0.1
set protocols bgp group untrust export mnha-route-policy
set protocols bgp group untrust local-as 100
set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group untrust bfd-liveness-detection multiplier 3
set protocols bgp group untrust neighbor 10.3.0.2
set routing-options autonomous-system 100
set routing-options static route 10.4.0.0/16 next-hop 10.3.0.2
set routing-options static route 10.111.0.2/32 next-hop 10.3.0.2

```

The following sections show configuration snippets on the router and switch required for setting up Multinode High Availability setup in the network.

### On the Router (MX960)

```

set interfaces ge-0/0/0 description HA
set interfaces ge-0/0/0 unit 0 family inet address 10.2.0.2/16
set interfaces ge-0/0/1 description HA
set interfaces ge-0/0/1 unit 0 family inet address 10.3.0.2/16
set interfaces ge-0/0/2 description trust
set interfaces ge-0/0/2 unit 0 family inet address 10.4.0.1/16
set interfaces lo0 description loopback
set interfaces lo0 unit 0 family inet address 10.111.0.2/32 primary
set interfaces lo0 unit 0 family inet address 10.111.0.2/32 preferred
set routing-options autonomous-system 100
set protocols bgp group mnha_r0 type internal
set protocols bgp group mnha_r0 local-address 10.2.0.2
set protocols bgp group mnha_r0 local-as 100
set protocols bgp group mnha_r0 bfd-liveness-detection minimum-interval 500
set protocols bgp group mnha_r0 bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group mnha_r0 bfd-liveness-detection multiplier 3
set protocols bgp group mnha_r0 neighbor 10.2.0.1
set protocols bgp group mnha_r0_b type internal
set protocols bgp group mnha_r0_b local-address 10.3.0.2
set protocols bgp group mnha_r0_b local-as 100
set protocols bgp group mnha_r0_b bfd-liveness-detection minimum-interval 500
set protocols bgp group mnha_r0_b bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group mnha_r0_b bfd-liveness-detection multiplier 3
set protocols bgp group mnha_r0_b neighbor 10.3.0.1

```



## On the Switch (EX9214)

```
set interfaces ge-0/0/0 description lan
set interfaces ge-0/0/0 mtu 9192
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members lan
set interfaces ge-0/0/1 description lan
set interfaces ge-0/0/1 mtu 9192
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lan
set interfaces ge-0/0/2 description lan
set interfaces ge-0/0/2 mtu 9192
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lan
set vlans lan vlan-id 1001
```

## Configuration

### Step-by-Step Procedure

We're showing the configuration of SRX-01 in the step-by-step procedure.

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the [CLI User Guide](#).

#### 1. Configure Interfaces.

```
[edit]
user@host# set interfaces ge-0/0/3 description "trust" unit 0 family inet address
10.1.0.1/16
user@host# set interfaces ge-0/0/4 description "untrust" unit 0 family inet address
10.2.0.1/16
user@host# set interfaces ge-0/0/2 description "ha_link" unit 0 family inet address
10.22.0.1/24
```

The interfaces ge-0/0/3 connects to the switch, ge-0/0/4 connects the router and the ge-0/0/2 interface is used for the ICL.



## 2. Configure the loopback interfaces.

```
[edit]
user@host# set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.1/32
user@host# set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.2/32
user@host# set interfaces lo0 description "untrust" unit 0 family inet address 10.11.0.3/32
```

Assign the IP address (10.11.0.1) to the loopback interface. This IP address acts as the floating IP address.

Using the loopback interface ensures that at any given point, traffic from the adjacent routers will be steered toward the floating IP address (that is, toward the active node).

## 3. Configure the security policies.

```
[edit]
user@host# set security policies default-policy permit-all
user@host# set security policies global policy All match source-address any
user@host# set security policies global policy All match destination-address any
user@host# set security policies global policy All match application any
user@host# set security policies global policy All then permit
```

Ensure you have configured security policies as per your network requirements. In this example, you'll configure a policy to permit all traffic.

## 4. Configure security zones, assign interfaces to the zones, and specify the allowed system services for the security zones.

```
[edit]
user@host# set security zones security-zone untrust host-inbound-traffic system-services ike
user@host# set security zones security-zone untrust host-inbound-traffic system-services ping
user@host# set security zones security-zone untrust host-inbound-traffic protocols bfd
user@host# set security zones security-zone untrust host-inbound-traffic protocols bgp
user@host# set security zones security-zone untrust interfaces ge-0/0/4
user@host# set security zones security-zone untrust interfaces lo0.0
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces ge-0/0/3
user@host# set security zones security-zone halink host-inbound-traffic system-services ike
user@host# set security zones security-zone halink host-inbound-traffic system-services ping
user@host# set security zones security-zone halink host-inbound-traffic system-services
```



```
high-availability
user@host# set security zones security-zone halink host-inbound-traffic system-services ssh
user@host# set security zones security-zone halink host-inbound-traffic protocols bfd
user@host# set security zones security-zone halink host-inbound-traffic protocols bgp
user@host# set security zones security-zone halink interfaces ge-0/0/2
```

Assign the interfaces ge-0/0/3 and ge-0/0/4 to the trust and untrust zones respectively. Assign the lo0.0 interface to the untrust zone to connect over the public IP network. Assign the interface ge-0/0/2 to the halink zone. You use this zone to set up the ICL.

5. Configure routing options.

```
[edit]
user@host# set routing-options autonomous-system 100
user@host# set routing-options static route 10.4.0.0/16 next-hop 10.2.0.2
user@host# set routing-options static route 10.111.0.2 next-hop 10.2.0.2
```

6. Configure both local node and peer node details such as node ID, IP addresses of the local node and peer node, and the interface for the peer node.

```
[edit]
user@host# set chassis high-availability local-id 1
user@host# set chassis high-availability local-id local-ip 10.22.0.1
user@host# set chassis high-availability peer-id 2 peer-ip 10.22.0.2
user@host# set chassis high-availability peer-id 2 interface ge-0/0/2.0
```

You'll use the ge-0/0/2 interface for communicating with the peer node using the ICL.

7. Attach the IPsec VPN profile IPSEC\_VPN\_ICL to the peer node.

```
[edit]
user@host# set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL
```

You'll need this configuration to establish a secure ICL link between the nodes.

8. Configure Bidirectional Forwarding Detection (BFD) protocol options for the peer node.

```
[edit]
user@host# set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
user@host# set chassis high-availability peer-id 2 liveness-detection multiplier 5
```



9. Associate the peer node ID 2 to the services redundancy group 0 (SRG0).

```
[edit]
user@host# set chassis high-availability services-redundancy-group 0 peer-id 2
```

10. Configure the services redundancy group 1 (SRG1).

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 deployment-type hybrid
user@host# set chassis high-availability services-redundancy-group 1 peer-id 2
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 1 ip
10.1.0.200/16
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 1 interface
ge-0/0/3.0
user@host# set chassis high-availability services-redundancy-group 1 virtual-ip 1 use-
virtual-mac
```

In this step, you specify the deployment type as hybrid, because you are setting up Multinode High Availability in a Layer 3 and Layer 2 network.

Assign a virtual IP (VIP) address and an interface for SRG1.

11. Configure IP and BFD monitoring parameters for SRG1 to check the reachability of an IP address and to detect failures in network.

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 monitor interface
ge-0/0/3
user@host# set chassis high-availability services-redundancy-group 1 monitor interface
ge-0/0/4
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.2.0.1 src-ip 10.2.0.2
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.2.0.1 session-type singlehop
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.2.0.1 interface ge-0/0/4.0
```

You can configure BFD liveliness by specifying source and destination IP addresses and the interface connecting to the peer device.

For IP monitoring, specify the interfaces used for connecting the neighboring router and switch.



**12. Configure an active signal route required for activeness enforcement.**

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 active-signal-route
10.39.1.1
user@host# set chassis high-availability services-redundancy-group 1 backup-signal-route
10.39.1.2
user@host# set chassis high-availability services-redundancy-group 1 preemption
user@host# set chassis high-availability services-redundancy-group 1 activeness-priority 200
```

The active signal route IP address you assign is used for route preference advertisement. You must specify the active signal route along with the route-exists policy in the policy-options statement.

**13. Configure CA certificates as per your requirements.**

```
[edit]
user@host# set security pki ca-profile Root-CA ca-identity Root-CA
user@host# set security pki ca-profile Root-CA enrollment url http://10.157.69.204/certsrv/
mscep/mscep.dll
user@host# set security pki ca-profile Root-CA revocation-check disable
```

**14. Define Internet Key Exchange (IKE) configuration for Multinode High Availability. An IKE configuration defines the algorithms and keys used to establish a secure connection.**

```
[edit]
user@host# set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
user@host# set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
user@host# set security ike proposal MNHA_IKE_PROP dh-group group14
user@host# set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
user@host# set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
user@host# set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
user@host# set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
user@host# set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
user@host# set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
user@host# set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
user@host# set security ike gateway MNHA_IKE_GW version v2-only
```

For the Multinode High availability feature, you must configure the IKE version as v2-only.



15. Specify the IPsec proposal protocol and encryption algorithm. Specify IPsec options to create a IPsec tunnel between two participant devices to secure VPN communication.

```
[edit]
user@host# set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
user@host# set security ipsec proposal MNHA_IPSEC_PROP protocol esp
user@host# set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
user@host# set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
user@host# set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
user@host# set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
user@host# set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
user@host# set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
user@host# set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
```

The same VPN name IPSEC\_VPN\_ICL must be mentioned for *vpn\_profile* in chassis high availability configuration. Specifying the ha-link-encryption option encrypts the ICL to secure high availability traffic flow between the nodes.

16. Configure policy options.

```
[edit]
user@host# set policy-options condition active_route_exists if-route-exists address-family
inet 10.39.1.1 table inet.0
user@host# set policy-options condition backup_route_exists if-route-exists address-family
inet 10.39.1.2 table inet.0
user@host# set policy-options policy-statement mnha-route-policy term 1 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 1 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists
user@host# set policy-options policy-statement mnha-route-policy term 1 then accept metric
10
user@host# set policy-options policy-statement mnha-route-policy term 2 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 2 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 2 from condition
backup_route_exists
user@host# set policy-options policy-statement mnha-route-policy term 2 then accept metric
20
user@host# set policy-options policy-statement mnha-route-policy term 3 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 3 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 3 then accept metric
30
```



```
user@host# set policy-options policy-statement mnha-route-policy term default then reject
```

Configure the active signal route 10.39.1.1 with the route match condition (if-route-exists). The Multinode High Availability module adds this route to the routing table when the node moves to the active role. The node also starts advertising the higher preference. Also configure the backup signal route (10.39.1.2) to advertise the backup node with a medium priority. In case of any failures, the high availability link goes down and the current active node releases its primary role and removes the active-signal-route. Now the backup node detects the condition through its probes and transitions to the active role. The route preference is swapped to drive all the traffic to the new active node.

#### 17. Configure BFD peering sessions options and specify liveness detection timers.

```
[edit]
user@host# set protocols bgp group untrust type internal
user@host# set protocols bgp group untrust local-address 10.2.0.1
user@host# set protocols bgp group untrust export mnha-route-policy
user@host# set protocols bgp group untrust neighbor 10.2.0.2
user@host# set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
user@host# set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval 500
user@host# set protocols bgp group untrust bfd-liveness-detection multiplier 3
user@host# set protocols bgp group untrust local-as 100
```

### Configuration Options for Software Upgrades

In Multinode High Availability, during software upgrade, you can divert the traffic by closing down interfaces on the node. Here, traffic cannot pass through the nodes. Check ["Software Upgrade in Multinode High Availability" on page 179](#) for details.

#### 1. Configure all traffic interfaces under "shutdown-on-failure" option.

```
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
<interface-name>
```

Example:

```
[edit]
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
```



```

ge-0/0/3
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
ge-0/0/4

```



**CAUTION:** Do not use interfaces assigned for the interchassis link (ICL).

## Results (SRX-1)

From configuration mode, confirm your configuration by entering the following commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show chassis high-availability
local-id 1 local-ip 10.22.0.1;
peer-id 2 {
    peer-ip 10.22.0.2;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 0 {
    peer-id {
        2;
    }
}
services-redundancy-group 1 {
    deployment-type hybrid;
    peer-id {
        2;
    }
    virtual-ip 1 {
        ip 10.1.0.200/16;
        interface ge-0/0/3.0;
    }
    monitor {

```



```

        bfd-liveliness 10.2.0.1 {
            src-ip 10.2.0.2;
            session-type singlehop;
            interface ge-0/0/4.0;
        }
        interface {
            ge-0/0/3;
            ge-0/0/4;
        }
    }
    active-signal-route {
        10.39.1.1;
    }
    backup-signal-route {
        10.39.1.2;
    }
    preemption;
    activeness-priority 200;
}

```

```

[edit]
user@host# show security ike
proposal MNHA_IKE_PROP {
    description mnha_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
policy MNHA_IKE_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IKE_PROP ;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway MNHA_IKE_GW {
    ike-policy MNHA_IKE_POL ;
}

```



```

    version v2-only;
}

```

```

[edit]
user@host# show security ipsec
proposal MNHA_IPSEC_PROP {
    description mnha_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3600;
}
policy MNHA_IPSEC_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IPSEC_PROP;
}
vpn IPSEC_VPN_ICL {
    ha-link-encryption;
    ike {
        gateway MNHA_IKE_GW;
        ipsec-policy MNHA_IPSEC_POL;
    }
}

```

```

[edit]
user@host# show policy-options
policy-statement mnha-route-policy {
    term 1 {
        from {
            protocol [ static direct ];
            condition active_route_exists;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 2 {
        from {
            protocol [ static direct ];
            condition backup_route_exists;

```



```

    }
    then {
        metric 20;
        accept;
    }
}
term 3 {
    from protocol [ static direct ];
    then {
        metric 30;
        accept;
    }
}
term default {
    then reject;
}
}
condition active_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}
}

```

```

user@host# show routing-options

```

```

autonomous-system 100;

```

```

static {

```



```

route 10.4.0.0/16 next-hop 10.2.0.2;
route 10.111.0.2/32 next-hop 10.2.0.2;
}

```

```

[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/4.0;
        lo0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}
security-zone halink {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
            high-availability;

```



```

        ssh;
    }
    protocols {
        bfd;
        bgp;
    }
}
interfaces {
    ge-0/0/2.0;
}
}

```

```

[edit]
user@host# show interfaces
ge-0/0/2 {
    description ha_link;
    unit 0 {
        family inet {
            address 10.22.0.1/24;
        }
    }
}
ge-0/0/3 {
    description trust;
    unit 0 {
        family inet {
            address 10.1.0.1/16;
        }
    }
}
ge-0/0/4 {
    description untrust;
    unit 0 {
        family inet {
            address 10.2.0.1/16;
        }
    }
}
lo0 {
    description untrust;
    unit 0 {

```



```

        family inet {
            address 10.11.0.1/32;
            address 10.11.0.2/32;
            address 10.11.0.3/32;
        }
    }
}

```

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

## Results (SRX-2)

From configuration mode, confirm your configuration by entering the following commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show chassis high-availability
local-id 2 local-ip 10.22.0.2;
peer-id 1 {
    peer-ip 10.22.0.1;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 0 {
    peer-id {
        1;
    }
}
services-redundancy-group 1 {
    deployment-type hybrid;
    peer-id {
        1;
    }
    virtual-ip 1 {
        ip 10.1.0.200/16;
        interface ge-0/0/3.0;
        use-virtual-mac;
    }
}

```



```

}
monitor {
    bfd-liveliness 10.3.0.1 {
        src-ip 10.3.0.2;
        session-type singlehop;
        interface ge-0/0/4.0;
    }
    interface {
        ge-0/0/3;
        ge-0/0/4;
    }
}
active-signal-route {
    10.39.1.1;
}
backup-signal-route {
    10.39.1.2;
}
activeness-priority 1;
}

```

```

[edit]
user@host# show security ike
proposal MNHA_IKE_PROP {
    description mnha_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
policy MNHA_IKE_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IKE_PROP ;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway MNHA_IKE_GW {
    ike-policy MNHA_IKE_POL ;
}

```



```

    version v2-only;
}

```

```

[edit]
user@host# show security ipsec
proposal MNHA_IPSEC_PROP {
    description mnha_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 3600;
}
policy MNHA_IPSEC_POL {
    description mnha_link_encr_tunnel;
    proposals MNHA_IPSEC_PROP;
}
vpn IPSEC_VPN_ICL {
    ha-link-encryption;
    ike {
        gateway MNHA_IKE_GW;
        ipsec-policy MNHA_IPSEC_POL;
    }
}

```

```

[edit]
user@host# show policy-options
route-filter-list loopback {
    10.11.0.0/24 orlonger;
}
route-filter-list ipsec {
    10.4.0.0/16 orlonger;
}
policy-statement mnha-route-policy {
    term 1 {
        from {
            protocol [ static direct ];
            condition active_route_exists;
        }
        then {
            metric 10;
            accept;
        }
    }
}

```



```

    }
}
term 2 {
    from {
        protocol [ static direct ];
        condition backup_route_exists;
    }
    then {
        metric 20;
        accept;
    }
}
term 3 {
    from protocol [ static direct ];
    then {
        metric 30;
        accept;
    }
}
term default {
    then reject;
}
}
condition active_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}

```



```

    }
}

```

```

[edit]
user@host# show routing-options
autonomous-system 100;
static {
    route 10.4.0.0/16 next-hop 10.3.0.2;
    route 10.111.0.2/32 next-hop 10.3.0.2;
}

```

```

[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/4.0;
        lo0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
    }
}

```



```

    }
}
security-zone halink {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
            high-availability;
            ssh;
        }
        protocols {
            bfd;
            bgp;
        }
    }
}
interfaces {
    ge-0/0/2.0;
}
}

```

```

[edit]
user@host# show interfaces
[edit]
root@10.52.45.32# show interfaces
ge-0/0/2 {
    description ha_link;
    unit 0 {
        family inet {
            address 10.22.0.2/24;
        }
    }
}
ge-0/0/3 {
    description trust;
    unit 0 {
        family inet {
            address 10.1.0.2/16;
        }
    }
}
ge-0/0/4 {

```



```

description untrust;
unit 0 {
    family inet {
        address 10.3.0.1/16;
    }
}
}
lo0 {
    description untrust;
    unit 0 {
        family inet {
            address 10.11.0.1/32;
            address 10.11.0.2/32;
            address 10.11.0.3/32;
        }
    }
}
}

```

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

On your security devices, you'll get the following message that asks you to reboot the device:

```

user@host# commit
warning: High Availability Mode changed, please reboot the device to avoid undesirable behavior
commit complete

```

## Verification

### IN THIS SECTION

- [Check Multinode High Availability Details | 165](#)
- [Check Multinode High Availability Peer Node Status | 167](#)
- [Check Multinode High Availability Service Redundancy Groups | 169](#)
- [Verify the Multinode High Availability Status Before and After Failover | 171](#)
- [Verify Interchassis Link \(ICL\) Encryption Status | 174](#)
- [Verify Link Encryption Tunnel Statistics | 176](#)



Confirm that the configuration is working properly.

## Check Multinode High Availability Details

### Purpose

View and verify the details of the Multinode High Availability setup configured on your security device.

### Action

From operational mode, run the following command:

On SRX-1

```
user@host> 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: 10.22.0.1
HA Peer Information:

  Peer Id: 2      IP address: 10.22.0.2    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 2

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
```



```

Deployment Type: HYBRID
Status: ACTIVE
Activeness Priority: 200
Preemption: ENABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 2
  Status : BACKUP
  Health Status: HEALTHY
  Failover Readiness: NOT READY

```

## On SRX-2

```

user@host> 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1   Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 1

SRG failure event codes:
  BF BFD monitoring
  IP IP monitoring

```



```
IF Interface monitoring
CP Control Plane monitoring
```

```
Services Redundancy Group: 1
  Deployment Type: HYBRID
  Status: BACKUP
  Activeness Priority: 1
  Preemption: DISABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: COMPLETE
  Failure Events: NONE
  Peer Information:
    Peer Id: 1
    Status : ACTIVE
    Health Status: HEALTHY
    Failover Readiness: N/A
```

## Meaning

Verify these details from the command output:

- Local node and peer node details such as IP address and ID.
- The field Encrypted: YES indicates that the traffic is protected.
- The field Deployment Type: HYBRID indicates a hybrid mode configuration—that is, the network has a router on one side and a switch on the other.
- The field Services Redundancy Group: 1 indicates the status of the SRG1 (ACTIVE or BACKUP) on that node.

## Check Multinode High Availability Peer Node Status

### Purpose

View and verify the peer node details.

### Action

From operational mode, run the following command:



## SRX-1

```

user@host> user@host> show chassis high-availability peer-info
HA Peer Information:

Peer-ID: 2      IP address: 10.22.0.2      Interface: ge-0/0/2.0
Routing Instance: default
Encrypted: YES   Conn State: UP
Cold Sync Status: COMPLETE
Internal Interface: st0.16000
Internal Local-IP: 180.100.1.1
Internal Peer-IP: 180.100.1.2
Internal Routing-instance: __juniper_private1__

Packet Statistics:
    Receive Error : 0      Send Error : 0

    Packet-type      Sent      Received

    SRG Status Msg      3          2

    SRG Status Ack      2          3

    Attribute Msg      4          2

    Attribute Ack      2          1

```

## SRX-2

```

user@host> show chassis high-availability peer-info
HA Peer Information:

Peer-ID: 1      IP address: 10.22.0.1      Interface: ge-0/0/2.0
Routing Instance: default
Encrypted: YES   Conn State: UP
Cold Sync Status: COMPLETE
Internal Interface: st0.16000
Internal Local-IP: 180.100.1.2
Internal Peer-IP: 180.100.1.1
Internal Routing-instance: __juniper_private1__

```



**Packet Statistics:**

Receive Error : 0

Send Error : 0

Packet-type	Sent	Received
-------------	------	----------

SRG Status Msg	2	3
----------------	---	---

SRG Status Ack	3	2
----------------	---	---

Attribute Msg	3	1
---------------	---	---

Attribute Ack	1	2
---------------	---	---

**Meaning**

Verify these details from the command output:

- Peer node details such as interface used, IP address, and ID
- Encryption status, connection status, and cold synchronization status
- Packet statistics across the node.

**Check Multinode High Availability Service Redundancy Groups****Purpose**

Verify that the SRGs are configured and working correctly.

**Action**

From operational mode, run the following command:

For SRG0:

```
user@host> show chassis high-availability services-redundancy-group 0
Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 2
```



For SRG1:

```
user@host> show chassis high-availability services-redundancy-group 1 >
```

SRG failure event codes:

```
BF  BFD monitoring
IP  IP monitoring
IF  Interface monitoring
CP  Control Plane monitoring
```

Services Redundancy Group: 1

```
Deployment Type: HYBRID
Status: ACTIVE
Activeness Priority: 200
Preemption: ENABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 2
  Status : BACKUP
  Health Status: HEALTHY
  Failover Readiness: NOT READY
```

Signal Route Info:

```
Active Signal Route:
IP: 10.39.1.1
Routing Instance: default
Status: INSTALLED
```

```
Backup Signal Route:
IP: 10.39.1.2
Routing Instance: default
Status: NOT INSTALLED
```

Virtual IP Info:

```
Index: 1
IP: 10.1.0.200/16
VMAC: N/A
Interface: ge-0/0/3.0
Status: INSTALLED
```



## Split-brain Prevention Probe Info:

DST-IP: 10.1.0.200

Routing Instance: default

Status: NOT RUNNING

Result: N/A Reason: N/A

## BFD Monitoring:

Status: UNKNOWN

SRC-IP: 10.2.0.2 DST-IP: 10.2.0.1

Routing Instance: default

Type: SINGLE-HOP

IFL Name: ge-0/0/4.0

State: INSTALLED

## Interface Monitoring:

Status: UP

IF Name: ge-0/0/4 State: Up

IF Name: ge-0/0/3 State: Up

**Meaning**

Verify these details from the command output:

- Peer node details such as deployment type, status, and active and back up signal routes.
- Virtual IP Information such as IP address and virtual MAC address.
- IP monitoring and BFD monitoring status.

**Verify the Multinode High Availability Status Before and After Failover****Purpose**

Check the change in node status before and after failover in a Multinode High Availability setup.



## Action

To check the Multinode High Availability status on the backup node (SRX-2), run the following command from operational mode:

```
user@host> 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 1

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: HYBRID
  Status: BACKUP
  Activeness Priority: 1
  Preemption: DISABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: COMPLETE
  Failure Events: NONE
  Peer Information:
```



```

Peer Id: 1
Status : ACTIVE
Health Status: HEALTHY
Failover Readiness: N/A

```

Under the Services Redundancy Group: 1 section, you can see the Status: BACKUP field. This field value indicates that the status of SRG 1 is backup.

Initiate the failover on the active node (SRX-1 device) and again run the command on the backup node (SRX-2).

```

user@host> 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: DOWN
  Cold Sync Status: IN PROGRESS

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 1

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: HYBRID
  Status: ACTIVE
  Activeness Priority: 1

```



```

Preemption: DISABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 1
  Status : BACKUP
  Health Status: HEALTHY
  Failover Readiness: READY

```

Note that under the Services Redundancy Group: 1 section, the status of SRG1 has changed from **BACKUP** to **ACTIVE**.

You can also see peer node details under the Peer Information section. The output shows the status of peer as **BACKUP**.

## Verify Interchassis Link (ICL) Encryption Status

### Purpose

Verify the interchassis link (ICL) status.

### Action

From operational mode, run the following command:

```

user@host> show security ipsec security-associations ha-link-encryption detail
ID: 495003 Virtual-system: root, VPN Name: IPSEC_VPN_ICL
  Local Gateway: 10.22.0.1, Remote Gateway: 10.22.0.2
  Traffic Selector Name: __IPSEC_VPN_ICL__multi_node__
  Local Identity: ipv4(180.100.1.1-180.100.1.1)
  Remote Identity: ipv4(180.100.1.2-180.100.1.2)
  TS Type: traffic-selector
  Version: IKEv2
  PFS group: N/A
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.16000, Tunnel MTU: 0, Policy-
name: MNHA_IPSEC_POL
  Port: 500, Nego#: 0, Fail#: 0, Def-Del#: 0 Flag: 0
  Multi-sa, Configured SAs# 0, Negotiated SAs#: 0
  HA Link Encryption Mode: Multi-Node
  Location: FPC -, PIC -, KMD-Instance -

```



```

Anchorship: Thread -
Distribution-Profile: default-profile
Direction: inbound, SPI: 0x00022d84, AUX-SPI: 0
                , VPN Monitoring: -
    Hard lifetime: Expires in 3395 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 2794 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    Location: FPC 0, PIC 0, KMD-Instance 0
    Anchorship: Thread 0
    IKE SA Index: 4294966277
Direction: outbound, SPI: 0x00028296, AUX-SPI: 0
                , VPN Monitoring: -
    Hard lifetime: Expires in 3395 seconds
    Lifesize Remaining: Unlimited
    Soft lifetime: Expires in 2794 seconds
    Mode: Tunnel(0 0), Type: dynamic, State: installed
    Protocol: ESP, Authentication: aes256-gcm, Encryption: aes-gcm (256 bits)
    Anti-replay service: counter-based enabled, Replay window size: 64
    Extended-Sequence-Number: Disabled
    tunnel-establishment: establish-tunnels-immediately
    Location: FPC 0, PIC 0, KMD-Instance 0
    Anchorship: Thread 0
    IKE SA Index: 4294966277

```

## Meaning

The command output provides the following information:

- The local gateway and remote gateway details.
- The IPsec SA pair for each threads in PIC.
- HA link encryption mode (as shown in the following line):

```
HA Link Encryption Mode: Multi-Node
```

- Authentication and encryption algorithms used



## Verify Link Encryption Tunnel Statistics

### Purpose

Verify link encryption tunnel statistics on both active and backup nodes.

### Action

From operational mode, run the following command:

```
user@host> show security ipsec statistics ha-link-encryption
```

#### ESP Statistics:

Encrypted bytes:	984248
Decrypted bytes:	462519
Encrypted packets:	9067
Decrypted packets:	8797

#### AH Statistics:

Input bytes:	0
Output bytes:	0
Input packets:	0
Output packets:	0

#### Errors:

AH authentication failures: 0, Replay errors: 0  
ESP authentication failures: 0, ESP decryption failures: 0  
Bad headers: 0, Bad trailers: 0  
Invalid SPI: 0, TS check fail: 0  
Exceeds tunnel MTU: 0  
Discarded: 0

### Meaning

If you see packet loss issues across a VPN, you can run the `show security ipsec statistics ha-link-encryption` command several times to verify that the encrypted and decrypted packet counters are incrementing. You should also check whether the other error counters are incrementing.

Use the `show security ike active-peer ha-link-encryption` command to display details of ICL on the active peer node.

Use the `clear security ipsec statistics ha-link-encryption` command to clear all IPsec statistics.



**SEE ALSO**

[Multinode High Availability | 2](#)

---

[Multinode High Availability Services | 43](#)

---

[Prepare Your Environment for Multinode High Availability Deployment | 40](#)

---

[Software Upgrade in Multinode High Availability | 179](#)

---

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

---

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)



# 3

CHAPTER

## Hardware and Software Upgrades

---

[Software Upgrade in Multinode High Availability | 179](#)

[Insert Additional SRX5K-SPC3 in a Multinode High Availability Setup | 198](#)

---



# Software Upgrade in Multinode High Availability

## IN THIS SECTION

- [Overview | 179](#)
- [Preinstallation Steps | 180](#)
- [Upgrade Software using install-route-on-failure | 183](#)
- [Upgrade Software using shutdown-on-failure interface | 190](#)

## Overview

In a Multinode High Availability setup, you can upgrade your SRX Series devices between two different Junos OS releases with minimal disruption of traffic.

We support a software upgrade method using the CLI as in Junos OS Release 22.3R1.

From Junos OS Release	To Junos OS Release	Use Software Upgrade Method
20.4	Any release post 20.4	No
22.3	Next version of Junos OS Release	Yes

For information about upgrade and downgrade support for Junos OS releases, see *Upgrade and Downgrade Support Policy for Junos OS Releases and Extended End-Of-Life Releases* in Release Notes.



**CAUTION:** When you are upgrading an SRX Series device from Junos OS Release 22.3 to the next version of the Junos OS release, you may experience some disruption in traffic.

You must install the same version of Junos OS on both the SRX Series devices in a Multinode High Availability setup. Therefore, when you upgrade Junos OS on one device, ensure that you upgrade the other device also to the same version.

We support following upgrade methods in Multinode High Availability setup:

- **For Layer 3 deployments:** The install-route-on-failure configuration (recommended). In this method, you can divert the traffic by changing the route. Here, traffic can still go through the node and



interface remains up. Go to ["Upgrade Software using install-route-on-failure" on page 183](#) for details. You can also use the shutdown-on-failure interfaces method for Layer 3 deployments.

- **For Hybrid deployment and Default gateway (Layer 2/switching) deployments:** The shutdown-on-failure interfaces method. In this method, you can divert the traffic by closing down interfaces on the node. Here, traffic cannot pass through the nodes. Go to ["Upgrade Software using shutdown-on-failure interface" on page 190](#) for details..

In the following procedure, we'll show you how to upgrade two SRX Series devices (SRX-01 and SRX-02) from Junos OS Release 22.3R1.1 to Junos OS Release 22.3R1.3 using CLI. To avoid downtime when upgrading SRX Series devices in Multinode High Availability setup, we'll update one device at a time.

## Best Practices for Upgrading Junos OS

Consider the following best practices when you plan your software upgrade:

- Ensure both nodes are online and have the same version of Junos OS.
- Prepare your SRX Series devices for an upgrade using the checklist available in [Preparing for Software Installation and Upgrade \(Junos OS\)](#).
- Check whether both nodes have sufficient storage in the `/var` file system by using the `show system storage` command.
- Check the status of all the cards on both the devices by using the `show chassis fpc pic-status` command.
- Verify that there are no major alarms on the devices by using the `show chassis alarms` command.
- Ensure that there are no uncommitted changes.
- Back up the active configuration and license keys.

We recommend that you perform software upgrades during a maintenance window.

## Preinstallation Steps

Complete the following tasks before you start the software upgrade.

- Check the current Junos OS software version on your device.

```
user@host> show version
Hostname: srx-01
```



```
Model: vSRX
Junos: 22.3R1.1
```

- Download the Junos OS image from the [Juniper Networks Support](#) page on both SRX Series devices and save it in the `/var/tmp` location.
- Use the `show chassis high-availability information` command to verify that your Multinode High Availability setup is healthy, functional, and that the interchassis link (ICL) is up.

### On SRX-01 Device

```
user@srx-01> 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: 10.22.0.1
HA Peer Information:

  Peer Id: 2      IP address: 10.22.0.2    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 2

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: ROUTING
  Status: ACTIVE
  Activeness Priority: 200
```



```

Preemption: ENABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 2
  Status : BACKUP
  Health Status: HEALTHY
  Failover Readiness: READY

```

## On SRX-02 Device

```

user@srx-02> 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1   Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 1

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: ROUTING

```



```

Status: BACKUP
Activeness Priority: 1
Preemption: DISABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: COMPLETE
Failure Events: NONE
Peer Information:
  Peer Id: 1
  Status : ACTIVE
  Health Status: HEALTHY
  Failover Readiness: N/A

```

These output samples confirm that the two SRX Series devices in the Multinode High Availability setup are in a healthy state and are operating normally.

You are now ready to proceed with software upgrade.

## Upgrade Software using install-route-on-failure

### Prerequisite for Diverting Transit Traffic

Check whether your device has the configuration required to divert transit traffic by changing the route as mentioned in ["Configuring Multinode High Availability In a Layer 3 Network" on page 48](#). If you haven't configured:

1. Create a dedicated custom virtual router for the route used for diverting traffic during the upgrade.

```
user@host# set routing-instances MNHA-signal-routes instance-type virtual-router
```

2. Configure the install-route-on-failure statement for SRG0. Here, you have configured the route with IP address 10.39.1.3 as the route to install when the node fails.

```

user@host# set routing-instances MNHA-signal-routes instance-type virtual-router
user@host# set chassis high-availability services-redundancy-group 0 install-route-on-failure
10.39.1.3 routing-instance MNHA-signal-routes
user@host# set chassis high-availability services-redundancy-group 1 active-signal-route
10.39.1.1 routing-instance MNHA-signal-routes

```



```
user@host# set chassis high-availability services-redundancy-group 1 backup-signal-route
10.39.1.2 routing-instance MNHA-signal-routes
```

The routing table installs the route mentioned in the statement when the node fails.

3. Configure a matching routing policy and define a policy condition based on the existence of routes. Here you include the route 10.39.1.3 as the route match condition for the if-route-exists.

```
user@host# set policy-options condition active_route_exists if-route-exists address-family
inet 10.39.1.1/32
user@host# set policy-options condition active_route_exists if-route-exists address-family
inet table MNHA-signal-routes.inet.0
user@host# set policy-options condition backup_route_exists if-route-exists address-family
inet 10.39.1.2/32
user@host# set policy-options condition backup_route_exists if-route-exists address-family
inet table MNHA-signal-routes.inet.0
user@host# set policy-options condition failure_route_exists if-route-exists address-family
inet 10.39.1.3/32
user@host# set policy-options condition failure_route_exists if-route-exists address-family
inet table MNHA-signal-routes.inet.0
```

Create the policy statement to refer the condition as one of the matching term.

```
user@host# set policy-options policy-statement mnha-route-policy term 4 from protocol static
user@host# set policy-options policy-statement mnha-route-policy term 4 from protocol direct
user@host# set policy-options policy-statement mnha-route-policy term 4 from condition
failure_route_exists
user@host# set policy-options policy-statement mnha-route-policy term 4 then metric 100
user@host# set policy-options policy-statement mnha-route-policy term 4 then accept
```

## Upgrade Multinode High Availability Software

Let's upgrade the device that is acting as the backup node (SRX-02).

1. Initiate the software upgrade process and commit the configuration.

```
user@srx-02# set chassis high-availability software-upgrade
```



This command initiates local failure for SRG0 and transitions SRG1 (if configured) to the INELIGIBLE state on the local device. The peer device now transitions to or stays in active state for SRG1. On the local node, the active and backup signal routes of SRG1 are removed. If you've configured the install-route-on-failure statement, the signal route associated with the install-route-on-failure configuration is installed. With the appropriate routing policies, the local device can advertise higher route metrics and divert the traffic away from the local device and steer the traffic toward the peer device,

## 2. Verify the status of Multinode High Availability.

```

user@srx-02> 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: OFFLINE [ SU ]
Local-id: 1
Local-IP: 10.22.0.1
HA Peer Information:

    Peer Id: 2      IP address: 10.22.0.2      Interface: ge-0/0/2.0
    Routing Instance: default
    Encrypted: YES   Conn State: UP
    Cold Sync Status: COMPLETE

Services Redundancy Group: 0
    Current State: ONLINE
    Peer Information:
        Peer Id: 2

SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: ROUTING
    Status: INELIGIBLE
    Activeness Priority: 200
    Preemption: ENABLED

```



```

Process Packet In Backup State: NO
Control Plane State: N/A
System Integrity Check: IN PROGRESS
Failure Events: NONE
Peer Information:
  Peer Id: 2
  Status : ACTIVE
  Health Status: HEALTHY
  Failover Readiness: N/A

```

The output shows Node Status: OFFLINE [ SU ], which indicates that the node is ready for the software upgrade. You can see that the status of the SRG1 has changed to INELIGIBLE.

3. Confirm that the other device (SRX-01) is in the active role and is functioning normally.

```

user@srx-01> show chassis high-availability informationNode 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:
    Peer Id: 1

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1

```



```

Deployment Type: ROUTING
Status: ACTIVE
Activeness Priority: 1
Preemption: DISABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 1
  Status : INELIGIBLE
  Health Status: UNHEALTHY
  Failover Readiness: NOT READY

```

The command output shows that the status of SRG1 is **ACTIVE**.

Also note that under the Peer Information section of the SRG1, the status is INELIGIBLE which indicates that the other node is in ineligible state.

4. Install the Junos OS software on the SRX-02 device.

```

user@srx-02> request system software add /var/tmp/junos-install-vsrx3-x86-64-22.3R1.3.tgz
no-copy

```

5. After a successful installation, reboot the device using the `request system reboot` command.
6. After the reboot, check the Junos OS version using the `show version` command.

```

user@srx-02> show version
Hostname: srx-02
Model: vSRX
Junos: 22.3R1.3

```

The output confirms that the device is upgraded to the correct Junos OS version.

7. Check status of the Multinode High Availability on the device.

```

user@srx-02> 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: OFFLINE [ SU ]

Local-id: 1

Local-IP: 10.22.0.1

HA Peer Information:

Peer Id: 2 IP address: 10.22.0.2 Interface: ge-0/0/2.0

Routing Instance: default

Encrypted: YES Conn State: UP

Cold Sync Status: COMPLETE

Services Redundancy Group: 0

Current State: ONLINE

Peer Information:

Peer Id: 2

SRG failure event codes:

BF BFD monitoring

IP IP monitoring

IF Interface monitoring

CP Control Plane monitoring

Services Redundancy Group: 1

Deployment Type: ROUTING

Status: INELIGIBLE

Activeness Priority: 200

Preemption: ENABLED

Process Packet In Backup State: NO

Control Plane State: N/A

System Integrity Check: COMPLETE

Failure Events: NONE

Peer Information:

Peer Id: 2

Status : ACTIVE

Health Status: HEALTHY

Failover Readiness: N/A

The output continues to display the node status as OFFLINE [ SU ] and SRG1 status as INELIGIBLE.



8. Remove the software-upgrade statement and commit the configuration.

```
user@srx-02# delete chassis high-availability software-upgrade
```

When you remove software-upgrade statement, the local failure state and installed routes are removed.

9. Check the Multinode High Availability status again to confirm that the device is online and the overall status is healthy and functioning.

```
user@srx02> 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: 10.22.0.1
HA Peer Information:

    Peer Id: 2      IP address: 10.22.0.2      Interface: ge-0/0/2.0
    Routing Instance: default
    Encrypted: YES   Conn State: UP
    Cold Sync Status: COMPLETE

Services Redundancy Group: 0
    Current State: ONLINE
    Peer Information:
        Peer Id: 2

SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: ROUTING
    Status: BACKUP
    Activeness Priority: 200
    Preemption: ENABLED
```



```

Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: IN PROGRESS
Failure Events: NONE
Peer Information:
  Peer Id: 2
  Status : ACTIVE
  Health Status: HEALTHY
  Failover Readiness: N/A

```

The output shows Node Status: ONLINE and SRG1 status as BACKUP, which indicates that the node is back online and is functioning normally in backup role.

10. Check interfaces, routing protocols, routes advertised and so on to confirm that your setup is operating normally.

Now you can proceed to upgrade the other device (SRX-01) using the same procedure.

**NOTE:** In case if you face any issues and are not able to complete the upgrade, you can roll back the software on the device, and then reboot the system. Use the `request system software rollback` command to restore the previously installed software version.

## Upgrade Software using shutdown-on-failure interface

### Prerequisite to Divert Transit Traffic

Check whether your SRX Series includes the configuration required to isolate traffic by shutting down interfaces as mentioned in ["Configuring Multinode High Availability In a Default Gateway Deployment" on page 93](#). if the feature is not configured:

1. Configure all traffic interfaces under the `shutdown-on-failure` option.

```

user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
interface-name

```



Example:

```
[edit]
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
ge-0/0/0
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
ge-0/0/1
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
ge-0/0/3
user@srx-02# set chassis high-availability services-redundancy-group 0 shutdown-on-failure
ge-0/0/4
```



**CAUTION:** Donot use interfaces assigned for the interchassis link (ICL).

## Upgrade Multinode High Availability Software

Let's upgrade the device that is acting as backup node (SRX-02).

1. Initiate the software upgrade and commit the configuration.

```
user@srx-02# set chassis high-availability software-upgrade
```

This command marks interfaces offline and transitions status to ineligible state.

2. Check the Multinode High Availability status.

```
user@srx-02> 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: OFFLINE [ SU ]
Local-id: 2
Local-IP: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1      Interface: ge-0/0/2.0
  Routing Instance: default
```



```

Encrypted: YES    Conn State: UP
Cold Sync Status: COMPLETE

```

```

Services Redundancy Group: 0
    Current State: ISOLATED [ Node Failure ]
    Peer Information:
        Peer Id: 1

    Shut-on-failures interfaces:
        ge-0/0/4

        ge-0/0/3

        ge-0/0/1

        ge-0/0/0

```

```

SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

```

```

Services Redundancy Group: 1
    Deployment Type: ROUTING
    Status: INELIGIBLE
    Activeness Priority: 1
    Preemption: DISABLED
    Process Packet In Backup State: NO
    Control Plane State: N/A
    System Integrity Check: COMPLETE
    Failure Events: NONE
    Peer Information:
        Peer Id: 1
        Status : ACTIVE
        Health Status: HEALTHY
        Failover Readiness: N/A

```

The output shows Node Status: OFFLINE [ SU ], which indicates that the node is ready for the software upgrade. You can also see SRG0 status as ISOLATED [ Node Failure ] and SRG1 status as INELIGIBLE.



### 3. Check the status of the interfaces.

```
user@host> show interfaces terse
```

Interface	Admin	Link	Proto	Local	Remote
ge-0/0/0	down	down			
ge-0/0/1	down	down			
ge-0/0/2	up	up			
ge-0/0/2.0	up	up	inet	10.22.0.2/24	
ge-0/0/3	down	down			
ge-0/0/3.0	up	down	inet	10.3.0.2/16	
ge-0/0/4	down	down			
ge-0/0/4.0	up	down	inet	10.5.0.1/16	

The output shows that interfaces marked for shutdown-on-failure are down.

### 4. Confirm that the other device (SRX-01) is in the active role and is functioning normally.

```
user@srx-01> show chassis high-availability information
```

Node failure codes:

```

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: 10.22.0.1

HA Peer Information:

```

Peer Id: 2      IP address: 10.22.0.2      Interface: ge-0/0/2.0
Routing Instance: default
Encrypted: YES  Conn State: UP
Cold Sync Status: COMPLETE

```

Services Redundancy Group: 0

```

Current State: ONLINE
Peer Information:
Peer Id: 2

```

SRG failure event codes:

```

BF  BFD monitoring
IP  IP monitoring

```



```

IF  Interface monitoring
CP  Control Plane monitoring

```

```

Services Redundancy Group: 1
  Deployment Type: ROUTING
  Status: ACTIVE
  Activeness Priority: 200
  Preemption: ENABLED
  Process Packet In Backup State: NO
  Control Plane State: READY
  System Integrity Check: N/A
  Failure Events: NONE
  Peer Information:
    Peer Id: 2
    Status : INELIGIBLE
    Health Status: UNHEALTHY
    Failover Readiness: NOT READY

```

The output shows that the status of SRG1 is ACTIVE.

Also note that under the Peer Information section of the SRG1, the status is INELIGIBLE which indicates that the other node is in ineligible state.

5. Install the Junos OS image on SRX-02.

```

user@srx-02> request system software add /var/tmp/junos-install-vsrx3-x86-64-22.3R1.3.tgz
no-copy

```

6. After the successful upgrade, reboot the device using the `request system reboot` command.

7. Check the Junos OS version.

```

user@srx-02> show version
Hostname: srx-02
Model: vSRX
Junos: 22.3R1.3

```

The output confirms that the device is upgraded to the correct Junos OS version.



## 8. Check the status of Multinode High Availability on the device.

```

user@srx-02> 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: OFFLINE [ SU ]
Local-id: 2
Local-IP: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1      Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ISOLATED [ Node Failure ]
  Peer Information:
    Peer Id: 1

  Shut-on-failures interfaces:
    ge-0/0/4

    ge-0/0/3

    ge-0/0/1

    ge-0/0/0

SRG failure event codes:
  BF  BFD monitoring
  IP  IP monitoring
  IF  Interface monitoring
  CP  Control Plane monitoring

Services Redundancy Group: 1
  Deployment Type: ROUTING
  Status: INELIGIBLE

```



```

Activeness Priority: 1
Preemption: DISABLED
Process Packet In Backup State: NO
Control Plane State: N/A
System Integrity Check: COMPLETE
Failure Events: NONE
Peer Information:
  Peer Id: 1
  Status : ACTIVE
  Health Status: HEALTHY
  Failover Readiness: N/A

```

The command output continues to display the node status as OFFLINE [ SU ] and SRG0 status as ISOLATED [ Node Failure ].

9. Remove the software-upgrade statement and commit the configuration.

```
user@srx-02# delete chassis high-availability software-upgrade
```

10. Check the Multinode High Availability status again on the device and confirm that the device is online and that the overall status is healthy.

```

user@srx-02> 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE

Services Redundancy Group: 0
  Current State: ONLINE
  Peer Information:

```



```

    Peer Id: 1

    Shut-on-failures interfaces:
    ge-0/0/4

    ge-0/0/3

    ge-0/0/1

    ge-0/0/0

SRG failure event codes:
BF  BFD monitoring
IP  IP monitoring
IF  Interface monitoring
CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: ROUTING
    Status: BACKUP
    Activeness Priority: 1
    Preemption: DISABLED
    Process Packet In Backup State: NO
    Control Plane State: READY
    System Integrity Check: COMPLETE
    Failure Events: NONE
    Peer Information:
        Peer Id: 1
        Status : ACTIVE
        Health Status: HEALTHY
        Failover Readiness: N/A
```

The output shows Node Status: ONLINE, and SRGO ONLINE, which indicates that the node is back online and is functioning normally.

11. Verify the status of interfaces.

```

user@srx-02> show interfaces terse

Interface          Admin Link Proto  Local          Remote
ge-0/0/0           up    up
gr-0/0/0           up    up
ge-0/0/1           up    up
```



ge-0/0/2	up	up		
ge-0/0/2.0	up	up	inet	10.22.0.2/24
ge-0/0/3	up	up		
ge-0/0/3.0	up	up	inet	10.3.0.2/16
ge-0/0/4	up	up		
ge-0/0/4.0	up	up	inet	10.5.0.1/16
.....				

The output shows that interfaces that were previously down are up now.

12. Check interfaces, routing protocols, routes advertised, and so on to confirm that your setup is operating normally.

Now you can proceed to upgrade the other device (SRX-01) using the same procedure.

RELATED DOCUMENTATION

<a href="#">Multinode High Availability   2</a>
<a href="#">Prepare Your Environment for Multinode High Availability Deployment   40</a>
<a href="#">Insert Additional SRX5K-SPC3 in a Multinode High Availability Setup   198</a>
<a href="#">Example: Configure Multinode High Availability in a Default Gateway Deployment   93</a>
<a href="#">Example: Configure Multinode High Availability in a Layer 3 Network   48</a>
<a href="#">Example: Configure Multinode High Availability in a Hybrid Deployment   134</a>

# Insert Additional SRX5K-SPC3 in a Multinode High Availability Setup

IN THIS SECTION

- [Insert SRX5K-SPC3 in a Multinode High Availability Setup | 199](#)



## Insert SRX5K-SPC3 in a Multinode High Availability Setup

Starting in Junos OS Release 22.2R1, you can insert additional Service Processing Cards (SPC3) cards in a SRX5000-Line devices in Multinode High Availability setup without interrupting the existing traffic flow or without incurring downtime on your network.

We strongly recommend that you install the additional SPC3 card during a maintenance window, or during times of low-traffic as the backup node is not available for some time.

### Requirements

Note the following requirements before you install additional SPC3 cards in a SRX5000-line device in a Multinode High Availability setup:

- Each security device must have at least one SPC3 card installed.
- When you are inserting a new SPC3 card, you must install it in a slot that has a higher number than the slots in which other SPCs are already installed. For example, if both nodes have an SPC3 card on slot 2, then you must insert the new SPC3 card in slot 3 or in a higher-numbered slot. You must not install the card in slot 0 or slot 1.
- Use the following table to know whether you can insert an additional SPC3 card on an SRX5000 chassis without interrupting the traffic based on the count of already installed SPC3 cards.

Existing Count of SPC3 Cards	Count After Inserting Additional SPC3 Cards	Installation Without Traffic Interruption
1	2	Yes
1	3 or more	No
2	3 or more	No
3 or more	4 or more	Yes

### Install Additional SPC3 Cards

Consider a Multinode High Availability setup with two SRX5000 line devices. You've two nodes—node 1 acting as the active node and node 2 as the backup node. You want to install SPC3 cards on both the nodes.



Familiarize yourself with the SPC3 installation procedure for your security device. See [Installing an SRX5400 Services Gateway SPC](#), or [Installing an SRX5600 Services Gateway SPC](#), or [Installing an SRX5800 Services Gateway SPC](#).

The following procedures guide you how to install an additional SPC3 card in a Multinode High Availability system.

#### **Case 1: Nonencrypted ICL**

1. Power off node 2 (backup node) using the `request system power off` command from operational mode.
2. Insert an SPC3 card or cards on node 2.
3. Boot up node 2.
4. Run the `show chassis high-availability information` command. If the device displays an error with the SPU Slot Mismatch message, you must halt the installation procedure and redo the procedure. If there are no error messages, continue with the next step.
5. When node 2 is back online and ready to failover on all SRGs, initiate a failover for all traffic and SRGs to node 2. You can use the `request chassis high-availability failover services-redundancy-group` command from the operational mode. When you run the command, the node 2 transitions to the active role.
6. Power off node 1.
7. Insert an SPC3 card or cards on node 1.
8. Boot up node 1 after you complete the installation.

#### **Case-2: Encrypted ICL**

1. Configure the `set chassis high-availability hardware-upgrade` statement and commit the configuration on both nodes.
2. Power off node 2 (backup node) using the `request system power off` command from operational mode.
3. Insert an SPC3 card or cards on node 2.
4. Run the `show chassis high-availability information` command. If the device displays an error with the SPU Slot Mismatch message, you must halt the upgrade procedure to not cause any disruption to the traffic. If there are no error messages, continue with the next step.
5. Boot up node 2.
6. When node 2 is back online and ready to fail over on all SRGs, initiate a failover for all traffic and SRGs to node 2 using the `request chassis high-availability failover services-redundancy-group` command from the operational mode. When you run the command, the node 2 transitions to the active role.



7. Power off node 1.
8. Insert an SPC3 card or cards on node 1.
9. Boot up node 1 after you complete the installation.
10. After node 1 is back online, configure the `delete chassis high-availability hardware-upgrade` statement on both the nodes and commit the configuration.

## How to Address SPC3 Slot Mismatch

If you face any issues while installing an additional SPC3 card, use the following steps to address the issue:

1. Run the `show chassis high-availability information` command.

If the device displays an error with the `Peer Hardware Incompatible: SPU Slot Mismatch` message, you must halt the upgrade procedure to not cause any disruption to the traffic.

2. Run the `show chassis fpc pic-status` command to check mismatched chassis slots between the two nodes.
3. Remove the wrongly placed card, and reinsert it into a correct slot, and perform the upgrade procedure once again.

## SEE ALSO

[Multinode High Availability | 2](#)

[Prepare Your Environment for Multinode High Availability Deployment | 40](#)

[Software Upgrade in Multinode High Availability | 179](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)



# 4

CHAPTER

## Multinode High Availability Support for vSRX

---

[Multinode High Availability Support for vSRX Instances](#) | 203

---



# Multinode High Availability Support for vSRX Instances

## SUMMARY

Read this topic to understand Multinode High Availability support for vSRX instances.

## IN THIS SECTION

- [Overview | 203](#)
- [Multinode High Availability in AWS | 203](#)
- [Configuring Multinode High Availability In Amazon Web Services \(AWS\) Deployment | 206](#)

## Overview

Starting in Junos OS Release 22.3R1, we support Multinode High Availability on vSRX virtual firewalls. Multinode High availability addresses high availability requirements for private and public cloud deployments by offering inter-chassis resiliency.

We support Multinode High Availability for vSRX instances for the following private and public cloud platforms:

- Kernel-based virtual machine (KVM) and VMWare ESXi
- Amazon Web Services (AWS)

## Multinode High Availability in AWS

### IN THIS SECTION

- [Terminology | 204](#)
- [Architecture | 205](#)



You can configure Multinode High Availability on the vSRX firewalls deployed on AWS. Participating nodes run both control and data-plane active at the same time and backup each other to ensure a fast synchronized failover in case of system or hardware failure. The Interchassis link (ICL) connection between the two devices synchronizes and maintains the state information and handles device failover scenarios.

Lets begin by getting familiar with Multinode High Availability terms specific to the AWS deployment.

## Terminology

Term	Description
Elastic IP Addresses	An Elastic IP address is a public IPv4 address, which is routable from the network/Internet. EIPs are dynamically bound to an interface of any node in Multinode High Availability setup. At any given time, EIPs are bound to only one interface and bound to the same node. The Multinode High Availability setup uses EIPs to control the traffic in AWS deployments. EIP acts similar to floating IP address in Layer 3 deployment or virtual IP address as in default gateway deployment. The node with an active SRG1 owns the EIP and draws the traffic toward it.
Inter-chassis link (ICL)	IP-based link (logical link) that connects nodes over a routed network in a Multinode High Availability system. The security device uses the ICL to synchronize and maintain the state information and to handle device failover scenarios. You can use only ge-0/0/0 interface to configure an ICL. The ICL uses MAC address assigned by AWS (not the virtual MAC created by vSRX VM). When you configure the ICL, ensure that the IP address is subnet of VPC. Note that cross VPC deployment is not supported.
Juniper Services Redundancy Protocol (jsrpd) process	JSRPD manages activeness determination and enforcement, and split-brain protection.

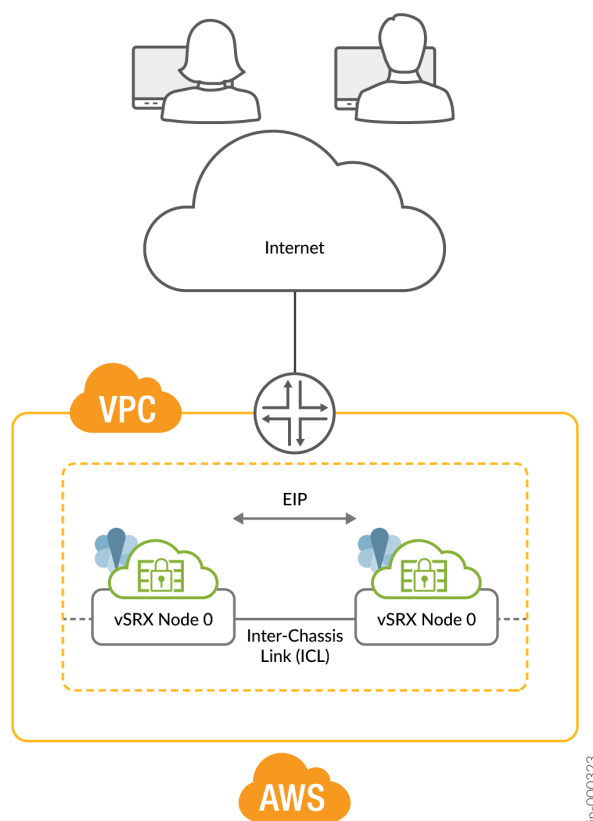
In Junos OS Release 22.3R1, we don't support IPSec VPN for Multinode High Availability in AWS deployments.



## Architecture

Figure 21 on page 205 shows two vSRX instances in Multinode High Availability setup deployed on AWS. In this deployment, two vSRX instances, one acting as the active node and the other as the backup node form a high availability pair. Two nodes run identical Junos OS image and have equal number of network interfaces configured.

**Figure 21: Public Cloud Deployment**



In Multinode High Availability setup, two vSRX instances are operating in active/backup mode. Both nodes connect to each other using an ICL for synchronizing control and data plane states. The vSRX instance, on which the SRG1 is active, hosts the Elastic IP address. The active node steers traffic towards it using the Elastic IP address. Backup node remains in standby mode and takes over on failover.

Juniper Services Redundancy Protocol (jsrpd) process communicates with AWS infrastructure to perform activeness determination and enforcement and provides split-brain protection.



During a failover, the Elastic IP address moves from the old active node to the new active node by triggering API (AWS SDK API) and draws traffic towards it. AWS updates the route tables to divert the traffic to the new active node.

This mechanism enables clients to communicate with the nodes using a single IP address. The Elastic IP is configured on the interface that connects to participating networks/segments.

### Split-Brain Protection

When the ICL between two nodes goes down, each node starts pinging to the peer node's interface IP using the probes. If the peer node is healthy, it responds to the probes. Otherwise, the jsrpd process communicates with AWS infrastructure to enforce the active role for the healthy node.

## Configuring Multinode High Availability In Amazon Web Services (AWS) Deployment

### IN THIS SECTION

- [Requirements | 206](#)
- [Topology | 207](#)
- [Configuration | 209](#)
- [Results | 213](#)
- [Verification | 219](#)

In this example, we'll show you how to configure Multinode High Availability on two vSRX instances in the Amazon Virtual Private Cloud (Amazon VPC).

### Requirements

This example uses the following hardware and software components:

- Two vSRX instances
- Junos OS Release 22.3R1

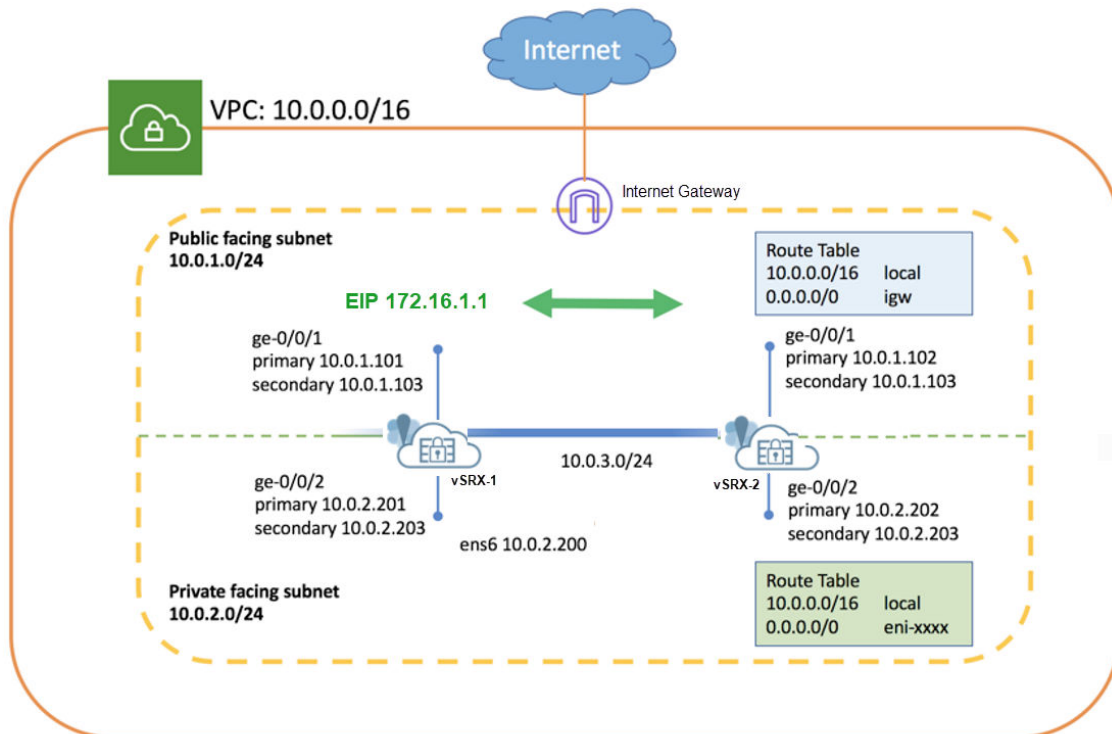


- An Amazon Web Services (AWS) account and an Identity and Access Management (IAM) role, with all required permissions to access, create, modify, and delete Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (S3), and Amazon Virtual Private Cloud (Amazon VPC) objects. Check [Configure an Amazon Virtual Private Cloud for vSRX](#) for details.
- In this example, we've set up an Amazon VPC with its associated Internet gateway, subnets, route table, and security groups. See [Configure an Amazon Virtual Private Cloud for vSRX](#).
- Launched and configured a vSRX instance in Amazon VPC. See [Launch a vSRX Instance on an Amazon Virtual Private Cloud](#).

## Topology

Figure 22 on page 207 shows the topology used in this example.

Figure 22: Multinode High Availability In AWS Deployment



As shown in the topology, two vSRX instances are deployed in the Amazon Virtual Private Cloud (Amazon VPC). The nodes communicate with each other using a routable IP address (Elastic IP address). Untrust side connects to public network and trust side connects to the protected resources.

Complete the following configurations before configuring Multinode High Availability on vSRX instances:



- Use instance tag in AWS to identify two vSRX instances as Multinode High Availability peers. For example, in the Name option, you can say vsrx-node-1 and for ha-peer option, you can say vsrx-node-2.
- Deploy both vSRX instances in the same Amazon VPC and availability zone.
- Assign IAM role for vSRX instance and launch vSRX as an Amazon Elastic Compute Cloud (EC2) instance with full permissions.
- Enable communication to the Internet by placing vSRX instances in public subnet. In the Amazon VPC, public subnets have access to the Internet gateway.
- Configure one VPC with multiple subnets to form high availability pair. The subnets are used to connect the two vSRX nodes (using a logical connection, similar to the physical cables connecting ports). In this example, we have CIDR for VPC is defined as 10.0.0.0/16, and created a total of four subnets to host vSRX traffic. Also you need a minimum four interfaces for both vSRX instances. [Table 5 on page 208](#) provides subnet and interfaces details.

**Table 5: Subnets Configurations**

Function	Port Number	Interface	Connection	Traffic Type	Subnet
Management	0	Fxp0	Management Interface	Management traffic	10.0.254.0/24
ICL	1	ge-0/0/0	Inter-chassis link to peer node	RTO, Sync, and probes-related traffic	10.0.253.0/24
Public	2	ge-0/0/1	Connect to public network. (Revenue Interface)	External traffic	10.0.1.0/24
Private	3	ge-0/0/2	Connect to private network. (Revenue Interface)	Internal traffic	10.0.2.0/24

Note that interface mapping with functionality mentioned in the table are default configuration and we recommend to use the same mapping in the configuration.

- Configure interfaces with primary and secondary IP addresses. You can associate EIP (Elastic IP address) as secondary IP addresses for an interface. Primary IP address is required during launching of instance and secondary IP address is transferable from one vSRX node to another during a failover. [Table 6 on page 209](#) show interface and IP address mappings used in this example.



**Table 6: Interface and IP Address Mappings**

Instance	Interface	Primary IP	Secondary IP
vSRX-1	ge-0/0/1	10.0.1.101	10.0.1.103 (EIP)
	ge-0/0/2	10.0.2.201	10.0.2.203 (EIP)
vSRX-2	ge-0/0/1	10.0.1.102	10.0.1.103 (EIP)
	ge-0/0/2	10.0.2.202	10.0.2.203 (EIP)

- Configure neighboring routers to include vSRX in the data path and mark vSRX as the next hop for the traffic. You can use EIP to configure the route. Example: `sudo ip route x.x.x.x/x dev ens6 via 10.0.2.203` where the 10.0.2.203 address is EIP.

## Configuration

### CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, 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 may vary based on the specific requirements of your environment.

#### On vSRX-1 Device

```
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 10.0.3.10
set chassis high-availability peer-id 2 peer-ip 10.0.3.11
set chassis high-availability peer-id 2 interface ge-0/0/0.0
set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
set chassis high-availability peer-id 2 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 1 deployment-type cloud
set chassis high-availability services-redundancy-group 1 peer-id 2
set chassis high-availability services-redundancy-group 1 preemption
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set security policies default-policy permit-all
set security zones security-zone fab host-inbound-traffic system-services all
```



```

set security zones security-zone fab host-inbound-traffic protocols all
set security zones security-zone fab interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/2.0
set security cloud high-availability aws eip-based
set security cloud high-availability aws peer-liveliness probe-ip 10.0.1.102
set security cloud high-availability aws peer-liveliness probe-ip routing-instance s1-router
set interfaces ge-0/0/0 mtu 9192
set interfaces ge-0/0/0 unit 0 family inet address 10.0.3.10/24
set interfaces ge-0/0/1 mtu 9192
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.101/24 primary
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.103/24
set interfaces ge-0/0/2 mtu 9192
set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.201/24 primary
set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.203/24
set routing-instances s1-router instance-type virtual-router
set routing-instances s1-router routing-options static route 0.0.0.0/0 next-hop 10.0.1.1
set routing-instances s1-router interface ge-0/0/1.0
set routing-instances s1-router interface ge-0/0/2.0

```

### On vSRX-2 Device

```

set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 10.0.3.11
set chassis high-availability peer-id 1 peer-ip 10.0.3.10
set chassis high-availability peer-id 1 interface ge-0/0/0.0
set chassis high-availability peer-id 1 liveness-detection minimum-interval 400
set chassis high-availability peer-id 1 liveness-detection multiplier 5
set chassis high-availability services-redundancy-group 1 deployment-type cloud
set chassis high-availability services-redundancy-group 1 peer-id 1
set chassis high-availability services-redundancy-group 1 preemption
set chassis high-availability services-redundancy-group 1 activeness-priority 100
set security policies default-policy permit-all
set security zones security-zone fab host-inbound-traffic system-services all
set security zones security-zone fab host-inbound-traffic protocols all
set security zones security-zone fab interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all

```



```

set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/2.0
set security cloud high-availability aws eip-based
set security cloud high-availability aws peer-liveliness probe-ip 10.0.1.101
set security cloud high-availability aws peer-liveliness probe-ip routing-instance s1-router
set interfaces ge-0/0/0 mtu 9192
set interfaces ge-0/0/0 unit 0 family inet address 10.0.3.11/24
set interfaces ge-0/0/1 mtu 9192
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.102/24 primary
set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.103/24
set interfaces ge-0/0/2 mtu 9192
set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.202/24 primary
set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.203/24
set routing-instances s1-router instance-type virtual-router
set routing-instances s1-router routing-options static route 0.0.0.0/0 next-hop 10.0.1.1
set routing-instances s1-router interface ge-0/0/1.0
set routing-instances s1-router interface ge-0/0/2.0

```

## Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#) in the [CLI User Guide](#).

### 1. Configure the interface for the ICL.

```

[edit]
user@host# set interfaces ge-0/0/0 mtu 9192
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.0.3.11/24

```

### 2. Configure interfaces for internal and external traffic.

```

[edit]
user@host# set interfaces ge-0/0/1 mtu 9192
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.102/24 primary
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.0.1.103/24
user@host# set interfaces ge-0/0/2 mtu 9192
user@host# set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.202/24 primary
user@host# set interfaces ge-0/0/2 unit 0 family inet address 10.0.2.203/24

```



The secondary IP address assigned to ge-0/0/1 interface is used as EIP.

3. Configure security zones, assign interfaces to the zones, and specify allowed system services for the security zones .

```
[edit]
user@host# set security zones security-zone fab host-inbound-traffic system-services all
user@host# set security zones security-zone fab host-inbound-traffic protocols all
user@host# set security zones security-zone fab interfaces ge-0/0/0.0
user@host# set security zones security-zone untrust host-inbound-traffic system-services all
user@host# set security zones security-zone untrust host-inbound-traffic protocols all
user@host# set security zones security-zone untrust interfaces ge-0/0/1.0
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces ge-0/0/2.0
```

4. Configure routing options.

```
[edit]
user@host# set routing-instances s1-router instance-type virtual-router
user@host# set routing-instances s1-router routing-options static route 0.0.0.0/0 next-hop
10.0.1.1
user@host# set routing-instances s1-router interface ge-0/0/1.0
user@host# set routing-instances s1-router interface ge-0/0/2.0
```

Create a separate routing instance type virtual router to separate management traffic and revenue traffic.

5. Configure local node and peer node details.

```
[edit]
user@host# set chassis high-availability local-id 1
user@host# set chassis high-availability local-id local-ip 10.0.3.10
user@host# set chassis high-availability peer-id 2 peer-ip 10.0.3.11
```

6. Associate the interface to peer node for interface monitoring and configure liveness detection details.

```
[edit]
user@host# set chassis high-availability peer-id 2 interface ge-0/0/0.0
```



```
user@host# set chassis high-availability peer-id 2 liveness-detection minimum-interval 400
user@host# set chassis high-availability peer-id 2 liveness-detection multiplier 5
```

## 7. Configure SRG1 by setting mode, deployment type.

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 deployment-type cloud
user@host# set chassis high-availability services-redundancy-group 1 peer-id 2
user@host# set chassis high-availability services-redundancy-group 1 preemption
user@host# set chassis high-availability services-redundancy-group 1 activeness-priority 200
```

## 8. Associate a peer ID to SRG1 and define activeness-priority and preemption.

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 peer-id 2
user@host# set chassis high-availability services-redundancy-group 1 preemption
user@host# set chassis high-availability services-redundancy-group 1 activeness-priority 200
```

## 9. Configure AWS deployment related options such as service type as EIP-based and specify details for monitoring options.

```
[edit]
user@host# set security cloud high-availability aws eip-based
user@host# set security cloud high-availability aws peer-liveliness probe-ip 10.0.1.101
user@host# set security cloud high-availability aws peer-liveliness probe-ip routing-instance
s1-router
```

# Results

## vSRX-1

From configuration mode, confirm your configuration by entering the following commands.

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show chassis high-availability
```



```

local-id 1 local-ip 10.0.3.10;
peer-id 2 {
    peer-ip 10.0.3.11;
    interface ge-0/0/0.0;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 1 {
    deployment-type cloud;
    peer-id {
        2;
    }
    preemption;
    activeness-priority 200;
}

```

```

[edit]
user@host# show routing-instances
s1-router {
    instance-type virtual-router;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.0.1.1;
        }
    }
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
}

```

```

[edit]
user@host# show security zones security-zone
security-zone fab {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    protocols {
        all;
    }
}

```



```

    }
  }
  interfaces {
    ge-0/0/0.0;
  }
}
security-zone untrust {
  host-inbound-traffic {
    system-services {
      all;
    }
    protocols {
      all;
    }
  }
  interfaces {
    ge-0/0/1.0;
  }
}
security-zone trust {
  host-inbound-traffic {
    system-services {
      all;
    }
    protocols {
      all;
    }
  }
  interfaces {
    ge-0/0/2.0;
  }
}
}

```

```

[edit]
user@host# show interfaces
ge-0/0/0 {
  mtu 9192;
  unit 0 {
    family inet {

```



```

        address 10.0.3.10/24;
    }
}
ge-0/0/1 {
    mtu 9192;
    unit 0 {
        family inet {
            address 10.0.1.101/24 {
                primary;
            }
            address 10.0.1.103/24;
        }
    }
}
ge-0/0/2 {
    mtu 9192;
    unit 0 {
        family inet {
            address 10.0.2.201/24 {
                primary;
            }
            address 10.0.2.203/24;
        }
    }
}

```

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

## vSRX-2

From configuration mode, confirm your configuration by entering the following commands.

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show chassis high-availability
local-id 2 local-ip 10.0.3.11;
peer-id 1 {
    peer-ip 10.0.3.10;
    interface ge-0/0/0.0;
}

```



```

    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 1 {
    deployment-type cloud;
    peer-id {
        1;
    }
    preemption;
    activeness-priority 100;
}

```

```

[edit]
user@host# show routing-instances
s1-router {
    instance-type virtual-router;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.0.1.1;
        }
    }
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
}

```

```

[edit]
user@host# show security zones
security-zone fab {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
}

```



```

    interfaces {
        ge-0/0/0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
}

```

```

[edit]
user@host# show interfaces
ge-0/0/0 {
    mtu 9192;
    unit 0 {
        family inet {
            address 10.0.3.11/24;
        }
    }
}

```



```

    }
}
ge-0/0/1 {
    mtu 9192;
    unit 0 {
        family inet {
            address 10.0.1.102/24 {
                primary;
            }
            address 10.0.1.103/24;
        }
    }
}
ge-0/0/2 {
    mtu 9192;
    unit 0 {
        family inet {
            address 10.0.2.202/24 {
                primary;
            }
            address 10.0.2.203/24;
        }
    }
}
}

```

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

## Verification

### Check Multinode High Availability Details

#### Purpose

View and verify the details of the Multinode High Availability setup configured on your vSRX instance.

#### Action

From operational mode, run the following command:

vSRX-1

```

user@host> 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: 10.0.3.10

HA Peer Information:

Peer Id: 2	IP address: 10.0.3.11	Interface: ge-0/0/0.0
Routing Instance: default		
Encrypted: NO	Conn State: UP	
Cold Sync Status: COMPLETE		

SRG failure event codes:

BF	BFD monitoring
IP	IP monitoring
IF	Interface monitoring
CP	Control Plane monitoring

Services Redundancy Group: 1

Deployment Type: CLOUD

Status: ACTIVE

Activeness Priority: 200

Preemption: ENABLED

Process Packet In Backup State: NO

Control Plane State: READY

System Integrity Check: N/A

Failure Events: NONE

Peer Information:

Peer Id: 2

Status : BACKUP

Health Status: HEALTHY

Failover Readiness: NOT READY

vSRX-2

user@host> **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: 10.0.3.11
HA Peer Information:

    Peer Id: 1      IP address: 10.0.3.10      Interface: ge-0/0/0.0
    Routing Instance: default
    Encrypted: NO   Conn State: UP
    Cold Sync Status: COMPLETE

SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: CLOUD
    Status: BACKUP
    Activeness Priority: 100
    Preemption: ENABLED
    Process Packet In Backup State: NO
    Control Plane State: NOT READY
    System Integrity Check: COMPLETE
    Failure Events: NONE
    Peer Information:
        Peer Id: 1
        Status : ACTIVE
        Health Status: HEALTHY
        Failover Readiness: N/A

```

## Meaning

Verify these details from the command output:

- Local node and peer node details such as IP address and ID.
- The field Deployment Type: CLOUD indicates that configuration is for the cloud deployment.
- The field Services Redundancy Group: 1 indicates the status of the SRG1 (ACTIVE or BACKUP) on that node.



## Check Multinode High Availability Information on AWS

### Purpose

View and verify cloud deployment details.

### Action

From operational mode, run the following command:

```
user@host> show security cloud high-availability information
Cloud HA Information:

Cloud Type      Cloud Service Type  Cloud Service Status
AWS             EIP                 Bind to Local Node
```

### Meaning

Verify these details from the command output:

- The field Cloud Type: AWS indicates the deployment is for AWS.
- The field Cloud Service Type: EIP indicates that EIP service type is used to control the traffic in AWS deployment.
- The field Cloud Service Status: Bind to Local Node displays EIP binding with local node. For the backup node, this field displays Bind to Peer Node

## Check Multinode High Availability Peer Node Status

### Purpose

Check the Multinode High Availability peer node status.

### Action

From operational mode, run the following command:

vSRX-1

```
user@host> show chassis high-availability peer-info
HA Peer Information:
```



```

Peer-ID: 2      IP address: 10.0.3.11      Interface: ge-0/0/0.0
Routing Instance: default
Encrypted: NO    Conn State: UP
Cold Sync Status: COMPLETE
Internal Interface: N/A
Internal Local-IP: N/A
Internal Peer-IP: N/A
Internal Routing-instance: N/A

```

Packet Statistics:

```

Receive Error : 0      Send Error : 0

```

Packet-type	Sent	Received
SRG Status Msg	7	6
SRG Status Ack	6	7
Attribute Msg	2	1
Attribute Ack	1	1

## vSRX-2

```

user@host> show chassis high-availability peer-info

```

HA Peer Information:

```

Peer-ID: 1      IP address: 10.0.3.10      Interface: ge-0/0/0.0
Routing Instance: default
Encrypted: NO    Conn State: UP
Cold Sync Status: COMPLETE
Internal Interface: N/A
Internal Local-IP: N/A
Internal Peer-IP: N/A
Internal Routing-instance: N/A

```

Packet Statistics:

```

Receive Error : 0      Send Error : 0

```

Packet-type	Sent	Received
SRG Status Msg	9	9



SRG Status Ack	9	9
Attribute Msg	3	2
Attribute Ack	2	2

### Meaning

Verify these details from the command output:

- Peer node details such as interface used, IP address, and ID.
- Packet statistics across the node.

## Check Multinode High Availability SRG

### Purpose

View and verify SRG details in Multinode High Availability.

### Action

From operational mode, run the following command:

```
user@host> show chassis high-availability services-redundancy-group 1
    SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: CLOUD
    Status: ACTIVE
    Activeness Priority: 200
    Preemption: ENABLED
    Process Packet In Backup State: NO
    Control Plane State: READY
    System Integrity Check: N/A
    Failure Events: NONE
    Peer Information:
        Peer Id: 2
```



```

Status : BACKUP
Health Status: HEALTHY
Failover Readiness: READY

Split-brain Prevention Probe Info:
DST-IP: 10.0.1.102
SRC-IP: 0.0.0.0
Routing Instance: s1-router
Status: NOT RUNNING
Result: N/A          Reason: N/A

```

### Meaning

Verify these details from the command output:

- SRG details such deployment type, status, activeness priority and preemption details.
- Peer node details.
- Split-brain prevention probe details.

## Verify the Multinode High Availability Status Before and After Failover

### Purpose

Check the change in node status before and after a failover in a Multinode High Availability setup.

### Action

Check the Multinode High Availability status on the backup node (SRX-2).

From operational mode, run the following command:

```

user@host> 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: 10.0.3.11
HA Peer Information:

Peer Id: 1      IP address: 10.0.3.10    Interface: ge-0/0/0.0

```



```

Routing Instance: default
Encrypted: NO      Conn State: UP
Cold Sync Status: COMPLETE

```

SRG failure event codes:

```

BF  BFD monitoring
IP  IP monitoring
IF  Interface monitoring
CP  Control Plane monitoring

```

Services Redundancy Group: 1

```

Deployment Type: CLOUD
Status: BACKUP
Activeness Priority: 100
Preemption: ENABLED
Process Packet In Backup State: NO
Control Plane State: NOT READY
System Integrity Check: COMPLETE
Failure Events: NONE
Peer Information:
  Peer Id: 1
  Status : ACTIVE
  Health Status: HEALTHY
  Failover Readiness: N/A

```

Under Services Redundancy Group: 1 option, you can see the Status: BACKUP. This indicates that the SRG-1 is in backup mode for the device.

Initiate the failover on the active node (vSRX-1) and again run the command on the backup node (vSRX-2).

```

user@host> 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: 10.0.3.11

HA Peer Information:



```

Peer Id: 1      IP address: 10.0.3.10    Interface: ge-0/0/0.0
Routing Instance: default
Encrypted: NO   Conn State: UP
Cold Sync Status: COMPLETE

```

SRG failure event codes:

```

BF  BFD monitoring
IP  IP monitoring
IF  Interface monitoring
CP  Control Plane monitoring

```

Services Redundancy Group: 1

```

Deployment Type: CLOUD
Status: ACTIVE
Activeness Priority: 100
Preemption: ENABLED
Process Packet In Backup State: NO
Control Plane State: READY
System Integrity Check: N/A
Failure Events: NONE
Peer Information:
  Peer Id: 1
  Status : BACKUP
  Health Status: HEALTHY
  Failover Readiness: NOT READY

```

## Meaning

You can notice that under Services Redundancy Group: 1 option, the status of SRG1 is changed from BACKUP to ACTIVE. This indicates that the node has transitioned into the active role and the other node (previous active) has transitioned to the backup role. You can see the other node's status in the Peer Information option. Here, the status says BACKUP.

## SEE ALSO

[Multinode High Availability | 2](#)

[Multinode High Availability Services | 43](#)

[Prepare Your Environment for Multinode High Availability Deployment | 40](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)



Example: Configure Multinode High Availability in a Hybrid Deployment | 134



# 5

CHAPTER

## Configuration Statements

---

activeness-probe | 230

hardware-upgrade | 232

high-availability (Chassis) | 233

high-availability (security cloud) | 236

liveness-detection (high availability) | 238

local-id | 241

peer-id | 243

monitor (Multinode High Availability) | 245

services-redundancy-group | 247

software-upgrade | 252

traceoptions | 253

virtual-ip | 256

---



# activeness-probe

## IN THIS SECTION

- [Syntax | 230](#)
- [Hierarchy Level | 230](#)
- [Description | 231](#)
- [Options | 231](#)
- [Required Privilege Level | 231](#)
- [Release Information | 231](#)

## Syntax

```
activeness-probe {  
  dest-ip {  
    ip-address;  
    routing-instance routing-instance;  
    src-ip src-ip;  
  }  
}
```

## Hierarchy Level

```
[edit chassis high-availability services-redundancy-group]
```



## Description

Specify the probe destination IP details for activeness determination

## Options

**activeness-probe**      **dest-ip** *ip-address*—Destination IP address to send the probe requests.  
                                 **routing-instance** *name*—Routing instance used by the probe requests.  
                                 **src-ip** *ip-address*—Source IP address to send the probe requests.

## Required Privilege Level

**interface**—To view this statement in the configuration.

**interface-control**—To add this statement to the configuration.

## Release Information

Statement introduced in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

[Example: Configure Multinode High Availability in a Layer 3 Network](#) | 48



# hardware-upgrade

## IN THIS SECTION

- [Syntax | 232](#)
- [Hierarchy Level | 232](#)
- [Description | 232](#)
- [Required Privilege Level | 233](#)
- [Release Information | 233](#)

## Syntax

```
hardware-upgrade
```

## Hierarchy Level

```
[edit chassis high-availability]
```

## Description

Install an additional SPC3 card in a Multinode High Availability statement. Use this statement when you insert an additional SPC3 on an SRX5000-line device in Multinode High Availability and has an interchassis link (ICL) in encryption mode. You must run the command on both nodes in Multinode High Availability before you start SPC3 installation.

Once you complete the installation, and both nodes are online, delete the statement on both nodes using the command `delete chassis high-availability hardware-upgrade`.



## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 22.2R1.

### RELATED DOCUMENTATION

[Hardware Upgrade for SRX5000-Line SPC3 in a Multinode High Availability Setup](#)

# high-availability (Chassis)

## IN THIS SECTION

- [Syntax | 233](#)
- [Hierarchy Level | 235](#)
- [Description | 235](#)
- [Options | 235](#)
- [Required Privilege Level | 236](#)
- [Release Information | 236](#)

## Syntax

```
high-availability {  
  local-id id local-ip local-ip;  
  peer-id name {  
    desc desc;  
    interface interface;
```



```

liveness-detection {
    no-adaptation;
    detection-time {
        threshold milliseconds;
    }
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    multiplier multiplier;
    transmit-interval {
        minimum-interval milliseconds;
        threshold milliseconds;
    }
    version (0 | 1 | automatic);
}
peer-ip peer-ip;
routing-instance routing-instance;
vpn-profile vpn-profile;
}
services-redundancy-group name {
    active-signal-route {
        ip-address;
        routing-instance routing-instance;
    }
    activeness-priority activeness-priority;
    activeness-probe {
        dest-ip {
            ip-address;
            routing-instance routing-instance;
        }
    }
    backup-signal-route {
        ip-address;
        routing-instance routing-instance;
    }
    monitor {
        bfd-liveliness name {
            interface interface;
            routing-instance routing-instance;
            session-type (multihop | singlehop);
            src-ip src-ip;
        }
        ip name {
            routing-instance routing-instance;

```



```

    }
  }
  peer-id id;
  process-packet-on-backup;
  shutdown-on-failure name;
}
traceoptions {
  file <filename> <files files> <match match> <size size> <(world-readable | no-world-
readable)>;
  flag name;
  level (alert | all | critical | debug | emergency | error | info | notice | warning);
  no-remote-trace;
}
}

```

## Hierarchy Level

[edit chassis]

## Description

Configure Multinode High Availability options on SRX Series devices.

## Options

["local-id" on page 241](#)

Define local node identifier.

["peer-id" on page 243](#)

Define peer node related information.

["services-redundancy-group" on page 247](#)

Define the services redundancy group details.

["traceoptions" on page 253](#)

Define high availability traceoptions.



## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

[Multinode High Availability | 2](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)

# high-availability (security cloud)

### IN THIS SECTION

- [Syntax | 237](#)
- [Hierarchy Level | 237](#)
- [Description | 237](#)
- [Options | 237](#)
- [Required Privilege Level | 238](#)
- [Release Information | 238](#)



## Syntax

```
high-availability {
    aws {
        eip-based;
        peer-liveliness {
            probe-ip {
                destination-ip-address;
                routing-instance instance-name;
                source-ip source-ip-address;
            }
        }
    }
}
```

## Hierarchy Level

```
[edit security cloud]
```

## Description

Configure vSRX instances to work in Multinode High Availability mode with Amazon Web Services (AWS).

## Options

aws—Define deployment type as Amazon Web Server (AWS)

- EIP-based—Deployment type based on EIP.



- peer-liveliness—Configure activeness probe options.
  - probe-ip *destination-ip*—IP address for activeness probe configuration. Specify destination IP address (of upstream router) for probing.
  - routing-instance—Routing-instance name.
  - source-ip—Source IP address to initiate probes.

## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

[Multinode High Availability Support for vSRX Instances](#) | 203

# liveness-detection (high availability)

## IN THIS SECTION

- [Syntax](#) | 239
- [Hierarchy Level](#) | 239
- [Description](#) | 239
- [Options](#) | 239
- [Required Privilege Level](#) | 241
- [Release Information](#) | 241



## Syntax

```
liveness-detection {
  detection-time {
    threshold milliseconds;
  }
  minimum-interval milliseconds;
  minimum-receive-interval milliseconds;
  multiplier multiplier;
  transmit-interval {
    minimum-interval milliseconds;
    threshold milliseconds;
  }
  version (0 | 1 | automatic);
}
```

## Hierarchy Level

```
[edit chassis high-availability peer-id]
```

## Description

Configure Bidirectional Forwarding Detection (BFD) options for the peer node.

## Options

**detection-time** Specify BFD failure detection time.



- **threshold**—Specify the threshold for the adaptation of the BFD session detection time. When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

Range: 1 through 255,000 milliseconds

**minimum-interval**

Configure the minimum interval at which the device transmits hello packets and then expects to receive a reply from a neighbor with which it has established a BFD session

- Range: 1 through 255,000 milliseconds

**minimum-receive-interval**

(Optional) Configure the minimum interval after which the local device must receive a reply from a neighbor with which it has established a BFD session.

- Range: 1 through 255,000 milliseconds

**multiplier**

(Optional) Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

- Range: 1 through 255
- Default: 3

**transmit-interval**

The negotiated transmit interval for a peer is the interval between the sending of BFD packets to peers. The receive interval for a peer is the minimum time that it requires between packets sent from its peer; the receive interval is not negotiated between peers.

- **minimum-interval *milliseconds***—Configure the minimum interval at which the local routing device transmits hello packets to a neighbor with which it has established a BFD session.
  - Range: 1 through 255
- **threshold *milliseconds***—Specify the threshold for the adaptation of the BFD session transmit interval. When the transmit interval adapts to a value greater than the threshold, a single trap and a single system message are sent.
  - Range: 0 through 4,294,967,295 ( $2^{32} - 1$ )

The threshold value specified in the threshold statement must be greater than the value specified in the minimum-interval statement for the transmit-interval statement.

**version**

BFD protocol version number

- 0—BFD version 0 (deprecated)



- 1—BFD version 1
- automatic—Choose BFD version automatically

The remaining statements are explained separately. See [CLI Explorer](#).

## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

[Multinode High Availability | 2](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

# local-id

### IN THIS SECTION

- [Syntax | 242](#)
- [Hierarchy Level | 242](#)
- [Description | 242](#)
- [Options | 242](#)
- [Required Privilege Level | 242](#)
- [Release Information | 242](#)



## Syntax

```
local-id id-number local-ip ip-address
```

## Hierarchy Level

```
[edit chassis high-availability]
```

## Description

Configure the local node identifier for Multinode High Availability.

## Options

<b>id</b>	Local identifier number
	<ul style="list-style-type: none"><li>• Range: 1 through 10</li></ul>
<b>local-ip</b>	Local IPv4 address

## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 20.4R1.



## RELATED DOCUMENTATION

[Multinode High Availability | 2](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

# peer-id

## IN THIS SECTION

- [Syntax | 243](#)
- [Hierarchy Level | 244](#)
- [Description | 244](#)
- [Options | 244](#)
- [Required Privilege Level | 245](#)
- [Release Information | 245](#)

## Syntax

```
peer-id name {  
    desc desc;  
    interface interface;  
    liveness-detection {  
        no-adaptation;  
        detection-time {  
            threshold milliseconds;  
        }  
        minimum-interval milliseconds;  
        minimum-receive-interval milliseconds;  
        multiplier multiplier;  
        transmit-interval {  
            minimum-interval milliseconds;  
            threshold milliseconds;  
        }  
    }
```



```
    version (0 | 1 | automatic);
  }
  peer-ip peer-ip;
  routing-instance routing-instance;
  vpn-profile vpn-profile;
}
```

## Hierarchy Level

[edit chassis high-availability]

## Description

Configure the other node related information in a Multinode High Availability setup.

## Options

<b>ID</b>	Peer node identifier.
<b>desc</b>	Peer node description.
<b>interface</b>	Name of the interface used for communicating with the peer node using the interchassis link (ICL).
<b><a href="#">"liveness-detection" on page 238</a></b>	Liveness detection options for the peer node.
<b>peer-ip</b>	IPv4 address of the peer node.
<b>routing-instance</b>	Routing instance to locate the peer node route.
<b>vpn-profile</b>	VPN profile name for ICL encryption. You must configure a VPN profile for the HA traffic and apply the profile for both the nodes.



## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

[Multinode High Availability | 2](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

# monitor (Multinode High Availability)

### IN THIS SECTION

- [Syntax | 245](#)
- [Hierarchy Level | 246](#)
- [Description | 246](#)
- [Options | 246](#)
- [Required Privilege Level | 247](#)
- [Release Information | 247](#)

## Syntax

```
monitor {  
    bfd-liveliness destination-ip-address {
```



```

        interface interface-name;
        routing-instance routing-instance-name;
        session-type (multihop | singlehop);
        src-ip src-ip;
    }
    interface name;
    ip destination-ip-address {
        routing-instance routing-instance-name;
    }
}

```

## Hierarchy Level

```
[edit chassis high-availability services-redundancy-group id-number]
```

## Description

Configure monitoring options for the Multinode High Availability setup.

## Options

**bfd-liveliness** Configure Bidirectional Forwarding Detection (BFD) monitoring to detect failures in a network. You can configure Multinode High Availability to monitor one or more links using BFD. This configuration triggers a failover in the event of BFD failure. Configure BFD liveliness by specifying source and destination IP and the interface where the peer device is directly connected to.

- **interface**—Name of the interface for single-hop sessions
- **routing-instance**—Routing instance to locate the route



- **session-type** —(Optional) Configure BFD session mode to be single-hop or multihop. By default, BGP uses single-hop BFD sessions if the peer is directly connected to the router's interface.
- **src-ip**—Source IPv4 or IPv6 address for activeness probe.

**interface**  
**interface-name** Configure interface monitoring for SRG. The node which detects the interface monitoring failure transitions to ineligible state for the corresponding SRG and the other node (if healthy) takes over the active role or that SRG and the subsequent GARP ensures traffic switching and recovery.

**ip destination-**  
**ip-address** Configure IP monitoring. You can configure IP monitoring by specifying destination IP address (of upstream router) for probing.

## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

[Multinode High Availability Monitoring | 30](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

# services-redundancy-group

## IN THIS SECTION

- [Syntax | 248](#)
- [Hierarchy Level | 249](#)



- [Description | 249](#)
- [Options | 249](#)
- [Required Privilege Level | 251](#)
- [Release Information | 251](#)

## Syntax

```

services-redundancy-group name {
  activeness-priority activeness-priority;
  activeness-probe {
    dest-ip {
      ip-address;
      routing-instance routing-instance;
      src-ip src-ip;
    }
  }
  active-signal-route {
    ip-address;
    routing-instance routing-instance;
  }
  backup-signal-route {
    ip-address;
    routing-instance routing-instance;
  }
  deployment-type(cloud | hybrid | routing | switching);
  monitor {
    bfd-liveliness name {
      interface interface;
      routing-instance routing-instance;
      session-type(multihop | singlehop);
      src-ip src-ip;
    }
    interface name;

    ip name {
      routing-instance routing-instance;

```



```
    }
  }
  peer-id id;
  preemption;
  prefix-list name {
    routing-instance routing-instance;
  }
  process-packet-on-backup;
  shutdown-on-failure name;

  virtual-ip name {
    interface interface;
    ip ip;
    use-virtual-mac;
  }
}
```

## Hierarchy Level

[edit chassis high-availability]

## Description

You can define the services level failover domain and to configure the SRX nodes to work in the high availability mode using the `services-redundancy-group` parameter.

## Options

<b>name</b>	Services redundancy group identifier
<b>active-signal-route</b>	IP address used for route preference advertisement. You must specify the active signal route along with the <code>route-exists</code> policy in the <code>policy-options</code> statement.



Signal route required for active role enforcement. When you configure the active-signal-route with if-route-exists condition, the HA module adds this route to the routing table.

- ip-address—IP address for active signal route
- routing-instance—Routing instance of the active signal route.

#### activeness-priority

Specify priority for the SRG1 in a node to take up the active role in a case where both nodes initialize at the same time. The node where SRG1 is in active state is considered as active node.

If you prefer a certain node to take over as the active node on boot, you can do one of the followings:

- Configure the upstream routers to include preferences for the path where the node is located.
- Configure the activeness priority for SRG1 on the SRX Series device (higher activeness priority). You can configure a priority for each node. As long as the nodes can communicate with each other through the ICL, the priority is honored.
- Allow the node with higher node ID (in case above two options not configured) to take the active role.
- **Range:** 1 through 254

#### "activeness-probe" on page 230

Specify the probe destination IP address for activeness determination.

#### backup-signal-route

Specify the backup signal route to advertise a route with a medium priority. When the HA link is down or the current active node relinquishes active role after any failure, the active signal route is removed from the routing table. The backup overwrites the default routing preference toward the old active node with the medium priority.

- ip-address—IP address for backup signal route
- routing-instance—Routing instance of the backup signal route.

#### deployment-type

Deployment type of the Services Redundancy Group.

- cloud—Cloud deployment
- hybrid—Hybrid deployment



- routing—Routing deployment
- switching—Switching/default gateway deployment

"monitor" on  
page 245

Specify to configure the BFD and IP monitoring options.

**peer-id**

Allows you to choose a specific peer when multiple HA peers are configured globally to the service redundancy group.

**preemption**

Allow preemption of activeness based on priority. When you configure the activeness priority (1-254) for the SRG1 and enable the preemptive behavior on both nodes, the preempt option ensures that the node with higher activeness priority always remains active after a failover.

**prefix-list**

Define a named set of address prefixes.

**process-packet-on-backup**

Enable packet forward engine to forward packets on backup node for the corresponding service redundancy group. When you configure the process packet on back up option, the Packet Forwarding Engine forwards packets on backup node for the corresponding SRG. This configuration processes VPN packets on the backup node even when the node is not active.

**shutdown-on-failure**

Configure one or multiple Interfaces which are required to be shut down to isolate the node in case of internal failures or during software upgrades. During software upgrades, you can divert the traffic by closing down interfaces on the node.

"virtual-ip" on  
page 256

IP address used for activeness determination and enforcement on the switching side. Required for hybrid and default gateway deployments.

The remaining statements are explained separately. See [CLI Explorer](#).

## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 20.4R1.



## RELATED DOCUMENTATION

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)

[Example: Configure Multinode High Availability in a Hybrid Deployment | 134](#)

# software-upgrade

## IN THIS SECTION

- [Syntax | 252](#)
- [Hierarchy Level | 252](#)
- [Description | 253](#)
- [Required Privilege Level | 253](#)
- [Release Information | 253](#)

## Syntax

```
software-upgrade
```

## Hierarchy Level

```
[edit chassis high-availability]
```



## Description

Enable software upgrade mode. Use this configuration option when you want to upgrade your security device in Multinode High Availability. This statement marks SRG status as offline and diverts the transit traffic to the other node.

Once you complete the software upgrade, delete the statement on both nodes using the command `delete chassis high-availability software-upgrade`.

## Required Privilege Level

interface

## Release Information

Statement introduced in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

[Software Upgrade in Multinode High Availability](#) | 179

# traceoptions

### IN THIS SECTION

- [Syntax](#) | 254
- [Hierarchy Level](#) | 254
- [Description](#) | 254
- [Options](#) | 254
- [Required Privilege Level](#) | 256
- [Release Information](#) | 256



## Syntax

```
traceoptions {
    file <filename> <files files> <match match> <size size> <(world-readable | no-world-
readable)>;
    flag name;
    level (alert | all | critical | debug | emergency | error | info | notice | warning);
    no-remote-trace;
}
```

## Hierarchy Level

```
[edit chassis high-availability]
```

## Description

Set Multinode High Availability traceoptions

## Options

**file *filename***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks

**file *name***—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory `/var/log`. We recommend that you place global routing protocol tracing output in the file **routing-log**.

**files *number***—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

- **Range:** 2 through 1000 files
- **Default:** 2 files



If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

**no-world-readable**—Restrict users from reading the log file.

**size *size***—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file.0***. When the ***trace-file*** again reaches its maximum size, ***trace-file.0*** is renamed ***trace-file.1*** and ***trace-file*** is renamed ***trace-file.0***. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

- **Syntax:** ***xk*** to specify KB, ***xm*** to specify MB, or ***xg*** to specify GB
- **Range:** 10 KB through the maximum file size supported on your system
- **Default:** 128 KB

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

**world-readable**—Allow users to read the log file.

**level**—Set a level of tracing.

- Values:
  - **alert**—Match alert conditions
  - **all**—Match all levels
  - **critical**—Match critical conditions
  - **debug**—Match debug messages
  - **emergency**—Match emergency conditions
  - **error**—Match error conditions
  - **info**—Match informational messages
  - **notice**—Match notice level messages
  - **warning**—Match warning messages

**no-remote-trace**—Disable remote tracing.



## Required Privilege Level

trace

## Release Information

Statement introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

[Multinode High Availability](#) | 2

# virtual-ip

### IN THIS SECTION

- [Syntax](#) | 256
- [Hierarchy Level](#) | 257
- [Description](#) | 257
- [Options](#) | 257
- [Required Privilege Level](#) | 257
- [Release Information](#) | 258

## Syntax

```
virtual-ip name {  
  interface interface;  
  ip ip;
```



```
use-virtual-mac;  
}
```

## Hierarchy Level

```
[edit chassis high-availability services-redundancy-group]
```

## Description

Specify IP address used for activeness determination and enforcement on the switching side of Multinode High Availability deployments. Required for hybrid and default gateway deployments.

## Options

<b>name</b>	Virtual IP Identifier.
<b>interface</b>	Name of the interface for virtual IP.
<b>ip</b>	IPV4/IPV6 prefix, should be in the same subnet as the interface IP
<b>use-virtual-mac</b>	Use virtual MAC (vMAC) address for services redundancy group role enforcement. Virtual MAC address dynamically assigned to the interface on active node that faces the switching side.

## Required Privilege Level

interface—To view this statement in the configuration.

interface-control—To add this statement to the configuration.



## Release Information

Statement introduced in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

[Multinode High Availability | 2](#)

---

[Example: Configure Multinode High Availability in a Default Gateway Deployment | 93](#)





## Operational Commands

---

`clear chassis high-availability data-plane statistics` | 261

`clear chassis high-availability information` | 262

`clear security pki node-local certificate-request` | 264

`clear security pki node-local local-certificate` | 266

`clear security pki node-local key-pair` | 268

`show chassis high-availability data-plane statistics` | 269

`show chassis high-availability information` | 275

`request chassis high-availability failover services-redundancy-group` | 281

`show chassis high-availability peer-info` | 283

`show chassis high-availability services-redundancy-group` | 285

`show security pki node-local local-certificate` | 290

`show security pki node-local certificate-request` | 296

`request security pki node-local local-certificate verify` | 300

`request security pki node-local local-certificate re-enroll` | 302

`request security pki node-local local-certificate load` | 304

`request security pki node-local local-certificate export` | 306

`request security pki node-local local-certificate enroll` | 308

`request security pki node-local key-pair export` | 312

`request security pki node-local generate-key-pair` | 314

`request security pki node-local generate-certificate-request` | 316







# clear chassis high-availability data-plane statistics

## IN THIS SECTION

- [Syntax | 261](#)
- [Description | 261](#)
- [Required Privilege Level | 261](#)
- [Output Fields | 261](#)
- [Sample Output | 262](#)
- [Release Information | 262](#)

## Syntax

```
clear chassis high-availability data-plane statistics
```

## Description

Clear the data plane statistics of Multinode High Availability.

## Required Privilege Level

clear

## Output Fields



## Sample Output

**clear chassis high-availability data-plane statistics**

```
user@host> clear chassis high-availability data-plane statistics
Cleared data-plane statistics
```

## Release Information

Command introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

| [show chassis high-availability data-plane statistics](#) | [269](#)

# clear chassis high-availability information

### IN THIS SECTION

- [Syntax](#) | [263](#)
- [Description](#) | [263](#)
- [Required Privilege Level](#) | [263](#)
- [Output Fields](#) | [263](#)
- [Sample Output](#) | [263](#)
- [Release Information](#) | [263](#)



## Syntax

```
clear chassis high-availability information
```

## Description

Clear Multinode High Availability information.

## Required Privilege Level

clear

## Output Fields

## Sample Output

```
clear chassis high-availability data-plane statistics
```

```
user@host> clear chassis high-availability information
```

```
Cleared chassis l3-ha information
```

## Release Information

Command introduced in Junos OS Release 20.4R1.



## RELATED DOCUMENTATION

[show chassis high-availability information](#) | [275](#)

# clear security pki node-local certificate-request

## IN THIS SECTION

- [Syntax](#) | [264](#)
- [Description](#) | [264](#)
- [Options](#) | [264](#)
- [Required Privilege Level](#) | [265](#)
- [Output Fields](#) | [265](#)
- [Sample Output](#) | [265](#)
- [Release Information](#) | [265](#)

## Syntax

```
clear security pki node-local certificate-request (all | certificate-id certificate-id-name)
```

## Description

Delete node-local digital certificate, certificate requests, and the corresponding public/private key pairs from the device in a Multinode High Availability setup.

## Options

**all** Delete all local digital certificate requests from the router.



**certificate-id** *certificate-id-name*

Delete the specified local digital certificate and corresponding public/private key pair.

## Required Privilege Level

clear

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

When you enter this command, you are provided feedback on the status of your request.

## Release Information

Command introduced in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

*Multinode High Availability*

*show pki security node-local certificate-request*



# clear security pki node-local local-certificate

## IN THIS SECTION

- [Syntax | 266](#)
- [Description | 266](#)
- [Options | 266](#)
- [Required Privilege Level | 267](#)
- [Output Fields | 267](#)
- [Sample Output | 267](#)
- [Release Information | 267](#)

## Syntax

```
clear security pki node-local local-certificate (all | certificate-id certificate-id | system-generated)
```

## Description

Clear public key infrastructure (PKI) information for local digital certificates on the local device in a Multinode High Availability system.

## Options

- `all`—Clear information for all the local digital certificates on the device.

You cannot clear the automatically generated self-signed certificate using `clear security pki local-certificate all` command. To clear the self-signed certificate you need to use `system-generated` as an option.



- `certificate-id certificate-id`—Clear the specified local digital certificate with this certificate ID.
- `system-generated`—Clear the existing automatically generated self-signed certificate and generate a new self-signed certificate.

## Required Privilege Level

clear and security

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

This command produces no output.

## Release Information

Command modified in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

*Multinode High Availability*

*show security pki node-local local-certificate*

*request security pki node-local local-certificate load*

[request security pki node-local local-certificate export](#)

*request security pki node-local local-certificate enroll*

*request security pki node-local local-certificate re-enroll*



# clear security pki node-local key-pair

## IN THIS SECTION

- [Syntax | 268](#)
- [Description | 268](#)
- [Options | 268](#)
- [Required Privilege Level | 269](#)
- [Output Fields | 269](#)
- [Release Information | 269](#)

## Syntax

```
clear security pki node-local key-pair (all | certificate-id certificate-id)
```

## Description

Clear public key infrastructure (PKI) key pair information for local digital certificates on the local device in a Multinode High Availability system.

## Options

- `all`—Clear key pair information for all local certificates.
- `certificate-id certificate-id` —Clear key pair information for the local certificate with this certificate ID.



## Required Privilege Level

clear and security

## Output Fields

This command produces no output.

## Release Information

Command introduced in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

---

*Multinode High Availability*

---

*request security pki node-local generate-key-pair*

---

*request security pki node-local key-pair export*

# show chassis high-availability data-plane statistics

### IN THIS SECTION

- [Syntax | 270](#)
- [Description | 270](#)
- [Required Privilege Level | 270](#)
- [Output Fields | 270](#)
- [Sample Output | 273](#)
- [Release Information | 275](#)



## Syntax

```
show chassis high-availability data-plane statistics
```

## Description

Display Multinode High Availability data plane statistics.

## Required Privilege Level

view

## Output Fields



**Table 7: show chassis cluster data-plane statistics Output Fields**

Field Name	Field Description
Services Synchronized	<ul style="list-style-type: none"> <li>• Service name—Name of the service.</li> <li>• Translation context—Messages synchronizing Network Address Translation (NAT) translation context.</li> <li>• Incoming NAT—Messages synchronizing incoming Network Address Translation (NAT) service.</li> <li>• Resource manager—Messages synchronizing resource manager groups and resources.</li> <li>• DS-LITE create—Messages synchronizing DS-LITE create sessions.</li> <li>• Session create—Messages synchronizing session creation.</li> <li>• IPv6 session create—Messages synchronizing IPv6 session create sessions.</li> <li>• IPv4/6 session RTO ACK—Messages synchronizing IPv4/6 session RTO ACK sessions.</li> <li>• Session close—Messages synchronizing session close.</li> <li>• IPv6 session close—Messages synchronizing IPv6 session close sessions.</li> <li>• Session change—Messages synchronizing session change.</li> <li>• IPv6 session change—Messages synchronizing IPv6 session change sessions.</li> <li>• ALG Support Library—Messages synchronizing ALG Support Library sessions.</li> <li>• Gate create—Messages synchronizing creation of pinholes (temporary openings in the firewall).</li> <li>• Session ageout refresh request—Messages synchronizing request session after age-out.</li> <li>• IPv6 session ageout refresh requests—Messages synchronizing IPv6 session ageout refresh requests.</li> <li>• Session ageout refresh replies—Messages synchronizing reply session after age-out.</li> </ul>



Table 7: show chassis cluster data-plane statistics Output Fields *(Continued)*

Field Name	Field Description
	<ul style="list-style-type: none"> <li>• IPv6 session ageout refresh replies—Messages synchronizing IPv6 session ageout refresh replies sessions.</li> <li>• IPsec VPN—Messages synchronizing VPN session.</li> <li>• Firewall user authentication—Messages synchronizing firewall user authentication session.</li> <li>• MGCP ALG—Messages synchronizing MGCP ALG sessions.</li> <li>• H323 ALG—Messages synchronizing H.323 ALG sessions.</li> <li>• SIP ALG—Messages synchronizing SIP ALG sessions.</li> <li>• SCCP ALG—Messages synchronizing SCCP ALG sessions.</li> <li>• PPTP ALG—Messages synchronizing PPTP ALG sessions.</li> <li>• JSF PPTP ALG—Messages synchronizing JSF PPTP ALG sessions.</li> <li>• RPC ALG—Messages synchronizing RPC ALG sessions.</li> <li>• RTSP ALG—Messages synchronizing RTSP ALG sessions.</li> <li>• RAS ALG—Messages synchronizing RAS ALG sessions.</li> <li>• MAC address learning—Messages synchronizing MAC address learning sessions.</li> <li>• GPRS GTP—Messages synchronizing GPRS GTP sessions.</li> <li>• GPRS SCTP—Messages synchronizing GPRS SCTP sessions.</li> <li>• GPRS FRAMEWORK—Messages synchronizing GPRS FRAMEWORK sessions.</li> <li>• JSF RTSP ALG—Messages synchronizing JSF RTSP ALG sessions.</li> <li>• JSF SUNRPC MAP—Messages synchronizing JSF SUNRPC MAP sessions.</li> <li>• JSF MSRPC MAP—Messages synchronizing JSF MSRPC MAP sessions.</li> <li>• DS-LITE delete—Messages synchronizing DS-LITE delete sessions.</li> <li>• JSF SLB—Messages synchronizing JSF SLB sessions.</li> </ul>



**Table 7: show chassis cluster data-plane statistics Output Fields (Continued)**

Field Name	Field Description
	<ul style="list-style-type: none"> <li>• APPID—Messages synchronizing APPID sessions.</li> <li>• JSF MGCP MAP—Messages synchronizing JSF MGCP MAP sessions.</li> <li>• JSF H323 ALG—Messages synchronizing JSF H323 ALG sessions.</li> <li>• JSF RAS ALG—Messages synchronizing JSF RAS ALG sessions.</li> <li>• JSF SCCP MAP—Messages synchronizing JSF SCCP MAP sessions.</li> <li>• JSF SIP MAP—Messages synchronizing JSF SIP MAP sessions.</li> <li>• PST_NAT_CREATE—Messages synchronizing PST NAT CREATE sessions.</li> <li>• PST_NAT_CLOSE—Messages synchronizing PST NAT CLOSE sessions.</li> <li>• PST_NAT_UPDATE—Messages synchronizing PST NAT UPDATE sessions.</li> <li>• JSF TCP STACK—Messages synchronizing JSF TCP STACK sessions.</li> <li>• JSF IKE ALG—Messages synchronizing JSF IKE ALG sessions.</li> </ul>

## Sample Output

### show chassis high-availability data-plane statistics

```

user@host> show chassis high-availability data-plane
statistics
Services Synchronized:
  Service name           RT0s sent  RT0s received
  Translation context    0          0
  Incoming NAT           0          0
  Resource manager       0          0
  DS-LITE create         0          0
  Session create         0          0
  IPv6 session create    0          0
  IPv4/6 session RT0 ACK 0          0

```



Session close	0	0
IPv6 session close	0	0
Session change	0	0
IPv6 session change	0	0
ALG Support Library	0	0
Gate create	0	0
Session ageout refresh requests	0	0
IPv6 session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPv6 session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
JSF PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0
GPRS SCTP	0	0
GPRS FRAMEWORK	0	0
JSF RTSP ALG	0	0
JSF SUNRPC MAP	0	0
JSF MSRPC MAP	0	0
DS-LITE delete	0	0
JSF SLB	0	0
APPID	0	0
JSF MGCP MAP	0	0
JSF H323 ALG	0	0
JSF RAS ALG	0	0
JSF SCCP MAP	0	0
JSF SIP MAP	0	0
PST_NAT_CREATE	0	0
PST_NAT_CLOSE	0	0
PST_NAT_UPDATE	0	0
JSF TCP STACK	0	0
JSF IKE ALG	0	0



## Release Information

Command introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

[clear chassis high-availability data-plane statistics](#) | [261](#)

# show chassis high-availability information

## IN THIS SECTION

- [Syntax](#) | [275](#)
- [Description](#) | [275](#)
- [Required Privilege Level](#) | [276](#)
- [Output Fields](#) | [276](#)
- [Sample Output](#) | [279](#)
- [Release Information](#) | [280](#)

## Syntax

```
show chassis high-availability information
```

## Description

Display details of the Multinode High Availability status on your security device including health status of the peer node.



## Required Privilege Level

view

## Output Fields

Table 8 on page 276 lists the output fields for the show chassis high-availability information command. Output fields are listed in the approximate order in which they appear.

Table 8: show chassis high-availability information

Field Name	Field Description
Node failure codes	<p>Node failure codes are:</p> <ul style="list-style-type: none"><li>• HW Hardware monitoring</li><li>• MB Mbuf monitoring</li><li>• CS Cold Sync monitoring</li><li>• LB Loopback monitoring</li><li>• SP SPU monitoring</li></ul>
Node Status	<ul style="list-style-type: none"><li>• status of the node</li><li>• Local-id—local identifier</li><li>• Local-IP—local IP address of the node</li></ul>



**Table 8: show chassis high-availability information (Continued)**

Field Name	Field Description
HA Peer Information	<ul style="list-style-type: none"> <li>• Description—peer information</li> <li>• Peer Id—peer identifier</li> <li>• IP address—peer IP address</li> <li>• Interface—interface name</li> <li>• Routing Instance—routing instance name</li> <li>• Encrypted—data encrypted status</li> <li>• Cold Sync Status—cold sync status of the node.</li> </ul>
HA Hardware Upgrade Events	Message on additional SPC3 installation status.
Services Redundancy Group	<ul style="list-style-type: none"> <li>• Current State—current state of the node</li> <li>• Peer Information—peer information</li> <li>• Peer Id—peer identifier</li> </ul>
SRG failure event codes	<ul style="list-style-type: none"> <li>• BF BFD monitoring—monitor Bidirectional Forwarding Detection.</li> <li>• IP IP monitoring—monitor IP address</li> <li>• CP Control Plane monitoring—monitor control plane state</li> </ul>



Table 8: show chassis high-availability information (Continued)

Field Name	Field Description
Services Redundancy Group	<ul style="list-style-type: none"> <li>• Status—node status</li> <li>• Activeness Priority—the node with the higher activeness priority become active for the service redundancy group.</li> <li>• Process Packet In Backup State—packet processing in Backup state.</li> <li>• Control Plane State—Displays the Control plane cold sync readiness after taking over the backup role. Means the control plane finished syncing all the control plane data from the new active node.  The options are: <ul style="list-style-type: none"> <li>• READY—Active mode</li> <li>• READY/NOT READY—Backup node based on the actual state</li> <li>• N/A—in other states</li> </ul> </li> <li>• System Integrity Check—Displays if the hold timer is running and system integrity check is under progress. Applicable only in BACKUP and INELIGIBLE state.</li> <li>• Failure Events—Displays local service redundancy group related attribute failure events.</li> <li>• Peer Information—Displays peer node information. <ul style="list-style-type: none"> <li>• Peer Id— Peer node identification number.</li> <li>• Status—Displays status of the peer node. The options are Active/Backup/Ineligible/Unknown.</li> <li>• Health Status—Displays health status of the peer node as advertised last <ul style="list-style-type: none"> <li>• HEALTHY—Peer node SRG state is healthy</li> <li>• UNHEALTHY—Peer node SRG state is unhealthy</li> <li>• UNKNOWN—Peer link is down</li> <li>• SRG NOT CONFIGURED—SRG not configured at peer node.</li> </ul> </li> </ul> </li> </ul>



Table 8: show chassis high-availability information (*Continued*)

Field Name	Field Description
	<ul style="list-style-type: none"> <li>Failover Readiness—Displays Failover readiness of the peer node and applicable only in Active state.</li> </ul> <p>The options are:</p> <ul style="list-style-type: none"> <li>READY—Peer's healthy, control and data plane cold sync is complete and hence peer is ready for a failover. A manual failover can be done</li> <li>NOT READY—Peer is either unhealthy, or data/control plane cold sync is pending</li> <li>UNKNOWN—Peer's failover readiness is awaited or if peer link is down</li> <li>N/A—In all states other than Active.</li> </ul>

## Sample Output

### show chassis high-availability information

```

user@host> 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: 10.22.0.2
HA Peer Information:

  Peer Id: 1      IP address: 10.22.0.1    Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: DOWN
  Cold Sync Status: IN PROGRESS

Services Redundancy Group: 0
  Current State: ONLINE

```



Peer Information:

Peer Id: 1

SRG failure event codes:

BF BFD monitoring

IP IP monitoring

IF Interface monitoring

CP Control Plane monitoring

Services Redundancy Group: 1

Deployment Type: ROUTING

Status: ACTIVE

Activeness Priority: 1

Preemption: DISABLED

Process Packet In Backup State: NO

Control Plane State: READY

System Integrity Check: N/A

Failure Events: NONE

Peer Information:

Peer Id: 1

Status : BACKUP

Health Status: HEALTHY

Failover Readiness: READY

## Release Information

Command introduced in Junos OS Release 20.4R1.

### RELATED DOCUMENTATION

[Multinode High Availability Monitoring | 30](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)



# request chassis high-availability failover services-redundancy-group

## IN THIS SECTION

- [Syntax | 281](#)
- [Description | 281](#)
- [Options | 281](#)
- [Required Privilege Level | 282](#)
- [Output Fields | 282](#)
- [Sample Output | 282](#)
- [Release Information | 282](#)

## Syntax

```
request chassis high-availability failover services-redundancy-group group-id peer-id peer-id
```

## Description

Initiate a manual failover on service redundancy group of the peer node. Use the command from the active node of the service redundancy group.

## Options

**services-redundancy-group** Service redundancy group on which to initiate manual failover.

- **Range:** 1 through 256



**peer-id** Node identifier of the SRG. After the failover, this node transitions as the new active node.

- **Range:** 1 through 10

## Required Privilege Level

maintenance

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**request chassis high-availability failover services-redundancy-group 1**

```
user@host> request chassis high-availability failover services-redundancy-group 1 peer-id 2
Initiated manual failover for redundancy group 1
```

## Release Information

Command introduced in Junos OS 20.4R1.

### RELATED DOCUMENTATION

[Hardware Upgrade for SRX5000-Line SPC3 in a Multinode High Availability Setup](#)



# show chassis high-availability peer-info

## IN THIS SECTION

- [Syntax | 283](#)
- [Description | 283](#)
- [Required Privilege Level | 283](#)
- [Output Fields | 283](#)
- [Sample Output | 284](#)
- [Release Information | 285](#)

## Syntax

```
show chassis high-availability peer-info
```

## Description

Display details of the peer node in a Multinode High Availability setup. Use this command to gather details of peer node, connection details, and packet statistics.

## Required Privilege Level

view

## Output Fields

[Table 9 on page 284](#) lists the output fields for the `show chassis high-availability information` command. Output fields are listed in the approximate order in which they appear.



**Table 9: show chassis high-availability peer-info**

Field Name	Field Description
HA Peer Information	<ul style="list-style-type: none"> <li>• Description—peer information</li> <li>• Peer Id—peer identifier</li> <li>• IP address—peer IP address</li> <li>• Interface—interface name</li> <li>• Routing Instance—routing instance name</li> </ul>
Internal connection details	Details related to internal traffic including internal interface, internal local IP address, internal peer IP address, and internal routing instance)
Packet Statistics	Details of packets sent and received. The detail includes: number of inbound/outbound errors and number of packets sent and received (SRG messages and attribute messages).

## Sample Output

### show chassis high-availability peer-info

```

user@host> show chassis high-availability peer-info
HA Peer Information:

  Peer-ID: 2      IP address: 10.22.0.2      Interface: ge-0/0/2.0
  Routing Instance: default
  Encrypted: YES   Conn State: UP
  Cold Sync Status: COMPLETE
  Internal Interface: st0.16000
  Internal Local-IP: 180.100.1.1
  Internal Peer-IP: 180.100.1.2
  Internal Routing-instance: __juniper_private1__
Packet Statistics:
  Receive Error : 0      Send Error : 0

```



Packet-type	Sent	Received
SRG Status Msg	4	3
SRG Status Ack	3	4
Attribute Msg	1	1
Attribute Ack	1	1

## Release Information

Command introduced in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

- [Multinode High Availability Monitoring | 30](#)
- [Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

# show chassis high-availability services-redundancy-group

### IN THIS SECTION

- [Syntax | 286](#)
- [Description | 286](#)
- [Required Privilege Level | 286](#)
- [Output Fields | 286](#)
- [Sample Output | 288](#)
- [Release Information | 289](#)



## Syntax

```
show chassis high-availability services-redundancy-group <services-redundancy-group-id>
```

## Description

Display the service redundancy group information in a Multinode High Availability setup.

## Required Privilege Level

view

## Output Fields

[Table 10 on page 286](#) lists the output fields for the `show chassis high-availability services-redundancy-group` command. Output fields are listed in the approximate order in which they appear.

**Table 10: show chassis high-availability services-redundancy-group**

Field Name	Field Description
SRG failure event codes	SRG failure event codes are: <ul style="list-style-type: none"><li>• BF BFD monitoring</li><li>• IP IP monitoring</li><li>• CP Control Plane monitoring</li></ul>



Table 10: show chassis high-availability services-redundancy-group *(Continued)*

Field Name	Field Description
Services Redundancy Group	<ul style="list-style-type: none"><li>• Status</li><li>• Activeness Priority</li><li>• Process Packet In Backup State</li><li>• Control Plane State</li><li>• System Integrity Check</li><li>• Failure Events</li><li>• Peer Information<ul style="list-style-type: none"><li>• Peer Id</li><li>• Status</li><li>• Health Status</li><li>• Failover Readiness</li></ul></li><li>• Signal Route Info<ul style="list-style-type: none"><li>• Active Signal Route</li><li>• IP</li><li>• Routing Instance</li><li>• Status</li><li>• Backup Signal Route</li><li>• IP</li><li>• Routing Instance</li><li>• Status</li></ul></li><li>• IP Monitoring</li><li>• DST-IP</li></ul>



Table 10: show chassis high-availability services-redundancy-group (Continued)

Field Name	Field Description
	<ul style="list-style-type: none"> <li>• Routing Instance</li> <li>• Status</li> <li>• Reason</li> </ul>

## Sample Output

### show chassis high-availability services-redundancy-group

For SRG0:

```
user@host> show chassis high-availability services-redundancy-group 0
Services Redundancy Group: 0
    Current State: ONLINE
    Peer Information:
        Peer Id: 2
```

For SRG1:

```
user@host> show chassis high-availability services-redundancy-group 1
SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 1
    Deployment Type: ROUTING
    Status: ACTIVE
    Activeness Priority: 200
    Preemption: ENABLED
    Process Packet In Backup State: NO
    Control Plane State: READY
```



System Integrity Check: N/A

Failure Events: NONE

Peer Information:

Peer Id: 2

Status : BACKUP

Health Status: HEALTHY

Failover Readiness: READY

Signal Route Info:

Active Signal Route:

IP: 10.39.1.1

Routing Instance: default

Status: INSTALLED

Backup Signal Route:

IP: 10.39.1.2

Routing Instance: default

Status: NOT INSTALLED

Split-brain Prevention Probe Info:

DST-IP: 10.111.0.1

SRC-IP: 10.11.0.1

Routing Instance: default

Status: NOT RUNNING

Result: N/A Reason: N/A

BFD Monitoring:

Status: UP

SRC-IP: 10.4.0.1 DST-IP: 10.4.0.2

Routing Instance: default

Type: SINGLE-HOP

IFL Name: ge-0/0/4.0

State: UP

## Release Information

Command introduced in Junos OS Release 20.4R1.



## RELATED DOCUMENTATION

[Multinode High Availability Monitoring | 30](#)

[Example: Configure Multinode High Availability in a Layer 3 Network | 48](#)

# show security pki node-local local-certificate

## IN THIS SECTION

- [Syntax | 290](#)
- [Description | 290](#)
- [Options | 291](#)
- [Required Privilege Level | 291](#)
- [Output Fields | 291](#)
- [Sample Output | 294](#)
- [Release Information | 295](#)

## Syntax

```
show security pki node-local local-certificate  
<brief/detail>  
<certificate-id certificate-id-name>  
<system-generated>
```

## Description

Display information about the local digital certificates, corresponding public keys, and the automatically generated self-signed certificate configured on the local device in a Multinode High Availability setup.



## Options

- none—Display basic information about all configured local digital certificates, corresponding public keys, and the automatically generated self-signed certificate.
- brief | detail—(Optional) Display the specified level of output.
- certificate-id *certificate-id-name* —(Optional) Display information about only the specified local digital certificates and corresponding public keys.
- system-generated—Display information about the automatically generated self-signed certificate.

## Required Privilege Level

view

## Output Fields

[Table 11 on page 291](#) lists the output fields for the `show security pki node-local local-certificate` command. Output fields are listed in the approximate order in which they appear.

**Table 11: show security pki node-local local-certificate Output Fields**

Field Name	Field Description
Certificate identifier	Name of the digital certificate.
Certificate version	Revision number of the digital certificate.
Serial number	<p>Unique serial number of the digital certificate. Starting in Junos OS Release 20.1R1, PKI local certificate serial number is displayed with <b>0x</b> as prefix to indicate that the PKI local certificate is in the hexadecimal format.</p> <p>Starting in Junos OS Release 21.4R1, you can view the serial number of the digital certificate in both hexadecimal and decimal formats.</p>
Issued to	Device that was issued the digital certificate.



**Table 11: show security pki node-local local-certificate Output Fields (Continued)**

Field Name	Field Description
Issued by	Authority that issued the digital certificate.
Issuer	<p>Authority that issued the digital certificate, including details of the authority organized using the distinguished name format. Possible subfields are:</p> <ul style="list-style-type: none"> <li>• Organization—Organization of origin.</li> <li>• Organizational unit—Department within an organization.</li> <li>• Country—Country of origin.</li> <li>• Locality—Locality of origin.</li> <li>• Common name—Name of the authority.</li> </ul>
LSYS	Name of the logical systems.
Subject	<p>Details of the digital certificate holder organized using the distinguished name format. Possible subfields are:</p> <ul style="list-style-type: none"> <li>• Organization—Organization of origin.</li> <li>• Organizational unit—Department within an organization.</li> <li>• Country—Country of origin.</li> <li>• Locality—Locality of origin.</li> <li>• Common name—Name of the authority.</li> <li>• Serial number—Serial number of the device.</li> </ul> <p>If the certificate contains multiple subfield entries, all entries are displayed.</p>
Subject string	Subject field as it appears in the certificate.
Alternate subject	Domain name or IP address of the device related to the digital certificate.



**Table 11: show security pki node-local local-certificate Output Fields (Continued)**

Field Name	Field Description
Cert-Chain	Starting in Junos OS Release 21.4R1, you can view the certificate chain for a given local certificate.
Validity	Time period when the digital certificate is valid. Values are: <ul style="list-style-type: none"> <li>• Not before—Start time when the digital certificate becomes valid.</li> <li>• Not after—End time when the digital certificate becomes invalid.</li> </ul>
Public key algorithm	Encryption algorithm used with the private key, such as rsa Encryption(1024 bits).
Public key verification status	Public key verification status: Failed or Passed. The detail output also provides the verification hash.
Signature algorithm	Encryption algorithm that the CA used to sign the digital certificate, such as sha1WithRSAEncryption.
Fingerprint	Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate.  Starting in Junos OS Release 21.4R1, you can also view the SHA-256 fingerprint for a local certificate along with SHA-1 and MD-5 fingerprints.
Distribution CRL	Distinguished name information and URL for the certificate revocation list (CRL) server.
Use for key	Use of the public key, such as Certificate signing, CRL signing, Digital signature, or Data encipherment.



## Sample Output

**show security pki node-local local-certificate certificate-id hello**

```
user@host> show security pki node-local local-certificate certificate-id cert-1234
LSYS: root-logical-system
Certificate identifier: cert-1234
  Issued to: tc5-5-1, Issued by: DC = Juniper, CN = root-551-AAA
  Validity:
    Not before: 10-14-2021 21:41 UTC
    Not after: 02-13-2026 14:27 UTC
  Public key algorithm: rsaEncryption(1024 bits)
  Keypair Location: Keypair generated locally
```

**show security pki node-local local-certificate system-generated**

```
user@host> show security pki node-local local-certificate system-generated
LSYS: root-logical-system
Certificate identifier: system-generated
  Issued to: 4a505bb373d7, Issued by: CN = 4a505bb373d7, CN = system generated, CN = self-signed
  Validity:
    Not before: 07-12-2019 22:23 UTC
    Not after: 07-10-2024 22:23 UTC
  Public key algorithm: rsaEncryption(2048 bits)
  Keypair Location: Keypair generated locally
```

**show security pki node-local local-certificate system-generated detail**

```
user@host> show security pki node-local local-certificate system-generated detail
LSYS: root-logical-system
Certificate identifier: system-generated
  Certificate version: 3

  Serial number:
    hexadecimal: 0x23171f4f104463e2847bc792c39eb614
    decimal: 46643037698975347221422984685160412692
  Issuer:
    Common name: 4a505bb373d7, Common name: system generated, Common name: self-signed
```



**Subject:**

Common name: 4a505bb373d7, Common name: system generated, Common name: self-signed

**Subject string:**

CN=4a505bb373d7, CN=system generated, CN=self-signed

**Validity:**

Not before: 07-12-2019 22:23 UTC

Not after: 07-10-2024 22:23 UTC

**Public key algorithm:** rsaEncryption(2048 bits)

30:82:01:0a:02:82:01:01:00:d5:7e:5e:7a:15:90:e3:23:07:8e:e3  
 4b:40:0e:95:33:31:8c:17:0b:d1:78:48:2e:b5:e8:cb:44:03:f1:fd  
 00:57:af:e9:d9:2c:78:96:04:37:3c:4a:65:d9:f1:fb:72:14:7f:b2  
 d3:42:d3:84:be:e8:c5:6c:e2:f5:91:8a:41:02:30:a7:8b:2f:10:5e  
 ab:5e:4e:d7:d6:f1:e7:ad:e3:6c:16:8d:6b:3c:0e:11:e9:26:8a:38  
 99:78:0a:57:67:cc:0a:ea:fa:35:2b:f3:51:4e:cc:30:ee:e9:a7:0a  
 26:14:42:fc:1b:22:ec:2d:0c:3b:10:d5:fb:e3:e6:ae:c6:cc:e7:de  
 0f:cf:4d:a7:87:11:e1:4e:7f:33:69:c0:16:4e:80:c8:57:b4:9a:f8  
 90:15:d8:e6:3e:06:7a:1c:a3:34:91:92:a6:88:9f:14:f5:89:39:da  
 0f:88:1c:b0:bd:7d:46:23:b2:42:e8:6f:d2:34:9e:f2:bd:00:34:23  
 99:4e:bb:39:0e:e4:bb:b2:9b:53:02:36:30:10:b7:28:e3:c4:8c:0e  
 4c:fd:cf:4f:58:81:72:91:b4:82:18:cf:ba:f6:76:59:f2:d5:36:e1  
 3a:29:20:72:02:5b:26:45:6f:92:0c:8e:dc:6c:d4:1c:78:55:db:66  
 3a:e9:9a:9c:81:02:03:01:00:01

**Signature algorithm:** sha256WithRSAEncryption**Fingerprint:**

0b:08:f8:bc:c6:a3:c1:41:75:2b:48:da:5d:a7:0f:d8:99:45:cd:8a (sha1)

8a:1b:b9:79:19:c6:c3:88:05:a8:05:28:3c:f2:b0:e9 (md5)

a3:9b:c1:c4:55:a8:f8:79:6f:a9:27:fc:f8:5a:af:45:37:dd:42:5f:2f:2b:bb:85:e3:f0:d7:99:9d:93:65:b1  
 (sha256)

## Release Information

Command modified in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

*Multinode High Availability*

*request security pki node-local local-certificate verify*



---

*request security pki node-local local-certificate re-enroll*

---

*request security pki node-local local-certificate load*

---

[request security pki node-local local-certificate export](#)

---

*request security pki node-local local-certificate enroll*

---

## show security pki node-local certificate-request

### IN THIS SECTION

- [Syntax | 296](#)
- [Description | 296](#)
- [Options | 297](#)
- [Required Privilege Level | 297](#)
- [Output Fields | 297](#)
- [Sample Output | 298](#)
- [Sample Output | 299](#)
- [Release Information | 299](#)

### Syntax

```
show security pki node-local certificate-request
<brief|detail>
<certificate-id certificate-id-name>
```

### Description

Display information about manually generated local digital certificate requests that are stored on the local device in your Multinode High Availability setup.



## Options

- none—Display basic information about all local digital certificate requests.
- brief / detail—(Optional) Display the specified level of output.
- certificate-id *certificate-id-name* —(Optional) Display information about only the specified local digital certificate requests.

## Required Privilege Level

view

## Output Fields

Table 12 on page 297 lists the output fields for the `show security pki node-local certificate-request` command. Output fields are listed in the approximate order in which they appear.

**Table 12: show security pki node-local certificate-request Output Fields**

Field Name	Field Description
Certificate identifier	Name of the digital certificate.
Certificate version	Revision number of the digital certificate.
Issued to	Device that was issued the digital certificate.



**Table 12: show security pki node-local certificate-request Output Fields (Continued)**

Field Name	Field Description
Subject	<p>Details of the digital certificate holder organized using the distinguished name format. Possible subfields are:</p> <ul style="list-style-type: none"> <li>• Organization—Organization of origin.</li> <li>• Organizational unit—Department within an organization.</li> <li>• Country—Country of origin.</li> <li>• Locality—Locality of origin.</li> <li>• Common name—Name of the authority.</li> </ul>
Alternate subject	Domain name or IP address of the device related to the digital certificate.
Public key algorithm	Encryption algorithm used with the private key, such as rsaEncryption(1024 bits).
Public key verification status	Public key verification status: Failed or Passed. The detail output also provides the verification hash.
Fingerprint	Secure Hash Algorithm (SHA1) and Message Digest 5 (MD5) hashes used to identify the digital certificate.
Use for key	Use of the public key, such as Certificate signing, CRL signing, Digital signature, or Data encipherment.

## Sample Output

**show security pki node-local certificate-request certificate-id user brief**

```
user@host> show security pki node-local certificate-request certificate-id user-1
brief
Certificate identifier: user-1
```



```

Issued to: user@example.net
Public key algorithm: rsaEncryption(1024 bits)

```

## Sample Output

**show security pki node-local certificate-request certificate-id user detail**

```

user@host> show security pki node-local certificate-request certificate-id user-1
detail
Certificate identifier: user
Certificate version: 3
Subject:
  Organization: example, Organizational unit: example, Country: IN,
  Common name: user1
Alternate subject: 192.168.72.124
Public key algorithm: rsaEncryption(1024 bits)
Public key verification status: Passed
c7:a4:fb:e7:8c:4f:31:e7:eb:01:d8:32:65:21:f2:eb:6f:7d:49:1a:c3:9b
63:47:e2:4f:f6:db:f6:c8:75:dd:e6:ec:0b:35:0a:62:32:45:6b:35:1f:65
c9:66:b7:40:b2:f9:2a:ab:5b:60:f7:c7:73:36:da:68:25:fc:40:4b:12:3c
d5:c8:c6:66:f6:10:1e:86:67:a8:95:9b:7f:1c:ae:a7:55:b0:28:95:a7:9a
a2:24:28:e4:5a:b2:a9:06:7a:69:37:20:15:e1:b6:66:eb:22:b5:b6:77:f6
65:88:b0:94:2b:91:4b:99:78:4a:e3:56:cc:14:45:d7:97:fd
Fingerprint:
  8f:22:1a:f2:9f:27:b0:21:6c:da:46:64:31:34:1f:68:42:5a:39:e0 (sha1)
  09:15:11:aa:ea:f9:5a:b5:70:d7:0b:8e:be:a6:d3:cb (md5)
Use for key: Digital signature

```

## Release Information

Command modified in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

*request security pki node-local generate-certificate-request*



# request security pki node-local local-certificate verify

## IN THIS SECTION

- [Syntax | 300](#)
- [Description | 300](#)
- [Options | 300](#)
- [Required Privilege Level | 301](#)
- [Output Fields | 301](#)
- [Sample Output | 301](#)
- [Release Information | 301](#)

## Syntax

```
request security pki node-local local-certificate verify certificate-id certificate-id-name
```

## Description

Verify the validity of the local digital certificate identifier on the local node in Multinode High Availability setup.

## Options

`certificate-id` *certificate-id-name* — Name of the local digital certificate identifier.



## Required Privilege Level

maintenance and security

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**request security pki node-local local-certificate verify certificate-id bme1 (not downloaded)**

You receive the following response before the certificate revocation list (CRL) is downloaded:

```
user@host> request security pki node-local local-certificate verify certificate-id
bme1
Local certificate bme1: CRL verification in progress. Please check the PKId debug logs for
completion status
```

**request security pki local-certificate verify certificate bme1 (downloaded)**

You receive the following response after the certificate revocation list (CRL) is downloaded:

```
user@host> request security pki node-local local-certificate verify certificate-id
bme1
Local certificate bme1 verification success
```

## Release Information

Command introduced in Junos OS Release 22.3R1.



## RELATED DOCUMENTATION

*Multinode High Availability*

*show security pki node-local local-certificate*

*request security pki node-local local-certificate re-enroll*

*request security pki node-local local-certificate load*

*request security pki node-local local-certificate export*

*request security pki node-local local-certificate enroll*

# request security pki node-local local-certificate re-enroll

## IN THIS SECTION

- [Syntax | 302](#)
- [Description | 303](#)
- [Options | 303](#)
- [Required Privilege Level | 303](#)
- [Output Fields | 303](#)
- [Release Information | 304](#)

## Syntax

```
request security pki node-local local-certificate re-enroll (cmpv2 | scep) certificate-id
certificate-id ca-profile profile-name challenge-password password
<re-generate-keypair>
<scep-digest-algorithm>
<scep-encryption-algorithm>
```



## Description

Manually reenroll an end-entity (EE) certificate with Certificate Management Protocol version 2 (CMPv2) or with Simple Certificate Enrollment Protocol (SCEP). This command initiates renewal of the EE certificate using the selected protocol and you can use the command in conjunction with the `set security pki auto-re-enrollment` command for automatic enrollment.

## Options

<code>scep</code>	Enroll end-entity certificate using SCEP protocol
<code>cmpv2</code>	Enroll certificate using CMPv2 protocol
<code>ca-profile-name</code> <i>ca-profile-name</i>	(Optional) CA profile name.
<code>certificate-id</code> <i>certificate-id-name</i>	Name of the local digital certificate.
<code>challenge-password</code>	Password used by CA for enrollment and revocation
<code>re-generate-keypair</code>	(Optional) Generate a PKI public/private key pair for the EE certificate.  Key generation might take a few seconds.
<code>scep-digest-algorithm</code>	Hash algorithm used for SCEP-PKCS7
<code>scep-encryption-algorithm</code>	Encryption algorithm used for SCEP-PKCS7

## Required Privilege Level

maintenance and security

## Output Fields

This command produces no output.



## Release Information

Command introduced in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

*Multinode High Availability*

*show security pki node-local local-certificate*

*request security pki node-local local-certificate enroll*

*request security pki node-local local-certificate load*

[request security pki node-local local-certificate export](#)

# request security pki node-local local-certificate load

### IN THIS SECTION

- [Syntax | 304](#)
- [Description | 305](#)
- [Options | 305](#)
- [Required Privilege Level | 305](#)
- [Output Fields | 305](#)
- [Sample Output | 305](#)
- [Release Information | 306](#)

## Syntax

```
request security pki node-local local-certificate load certificate-id certificate-id
filename file-name key key-string
<passphrase passphrase-string>
```



## Description

Manually load a local digital certificate from a specified location on the local device in a Multinode High Availability setup.

## Options

<b>certificate-id</b>	Name of the certificate identifier
<b>filename</b>	Filename that contains the certificate to load
<b>key</b>	File pathname that contains the private key/key-pair to loaded
<b>passphrase</b>	Passphrase of the private key/key-pair (PEM) file

## Required Privilege Level

maintenance and security

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**request security pki node-local local-certificate load**

```
user@host> request security pki node-local local-certificate load filename cert_name.crt key
key_name.key certificate-id test
Local certificate cert_name.crt loaded successfully
```



## Release Information

Command introduced in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

*Multinode High Availability*

*show security pki node-local local-certificate*

*request security pki node-local local-certificate enroll*

[request security pki node-local local-certificate export](#)

*request security pki node-local local-certificate re-enroll*

# request security pki node-local local-certificate export

### IN THIS SECTION

- [Syntax | 307](#)
- [Description | 307](#)
- [Options | 307](#)
- [Required Privilege Level | 307](#)
- [Output Fields | 307](#)
- [Sample Output | 308](#)
- [Release Information | 308](#)



## Syntax

```
request security pki node-local local-certificate export certificate id certificate-id-name  
filename path/filename  
<type (der | pem)>
```

## Description

Export a generated self-signed certificate from the default location (var/db/certs/common/local) to a specific location within the local device in a Multinode High Availability setup.

## Options

<b>certificate id</b> <i>certificate-id-name</i>	Name of the local digital certificate.
<b>filename</b> <i>path/filename</i>	Target directory location and filename of the CA digital certificate.
<b>type</b> (der   pem)	Certificate format: DER (distinguished encoding rules) or PEM (privacy-enhanced mail).

## Required Privilege Level

maintenance

## Output Fields

When you enter this command, you are provided feedback on the status of your request.



## Sample Output

**request security pki node-local local-certificate export**

```
user@host> request security pki node-local local-certificate export filename /var/tmp/my-
cert.pem certificate-id nss-cert type pem
certificate exported successfully
```

## Release Information

Command introduced in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

*Multinode High Availability*

*show security pki node-local local-certificate*

*request security pki node-local local-certificate enroll*

*request security pki node-local local-certificate load*

*request security pki node-local local-certificate re-enroll*

# request security pki node-local local-certificate enroll

### IN THIS SECTION

- [Syntax | 309](#)
- [Description | 309](#)
- [Options | 310](#)
- [Required Privilege Level | 311](#)



- [Output Fields | 311](#)
- [Sample Output | 311](#)
- [Release Information | 311](#)

## Syntax

```
request security pki node-local local-certificate enroll
  ca-dn subject-dn
  ca-profile ca-profile name
  ca-reference reference
  ca-secret shared-secret
  certificate-id certificate-id-name
  challenge-password password
  cmpv2
  digest
  domain-name domain-name
  email email-address
  ip-address ip-address
  ipv6-address ipv6-address
  scep
  scep-digest-algorithm
  scep-encryption-algorithm
  subject subject-distinguished-name
```

## Description

Enroll and install a local digital certificate online by using CMPv2 or Simple Certificate Enrollment Protocol (SCEP). This command loads both end-entity (EE) and CA certificates based on the CA server configuration. Certificate revocation list (CRL) or Online Certificate Status Protocol (OCSP) can be used to check the revocation status of a certificate.



## Options

<b>ca-profile</b> <i>ca-profile-name</i>	CA profile name.
<b>certificate-id</b> <i>certificate-id-name</i>	Name of the local digital certificate and the public/private key pair.
<b>challenge-password</b> <i>password</i>	Password set by the administrator and normally obtained from the SCEP enrollment webpage of the CA. The password is 16 characters in length
<b>cmpv2</b>	Enroll certificate using CMPv2 protocol.
<b>domain-name</b> <i>domain-name</i>	Fully qualified domain name (FQDN). The FQDN provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.
<b>email</b> <i>email-address</i>	E-mail address of the certificate holder.
<b>ip-address</b> <i>ip-address</i>	IP address of the router.
<b>ipv6-address</b> <i>ipv6-address</i>	IPv6 address of the router for the alternate subject.
<b>scep</b>	Enroll certificate using Simple Certificate Enrollment Protocol (SCEP) protocol.
<b>scep-digest-algorithm</b>	Hash algorithm digest, either MD5 or SHA-1; SHA-1 is the default.
<b>scep-encryption-algorithm</b>	Encryption algorithm, either DES or DES3; DES3 is the default.
<b>subject</b> <i>subject-distinguished-name</i>	Distinguished Name (DN) format that contains the domain component, common name, department, serial number, company name, state, and country in the following format: DC, CN, OU, O, SN, L, ST, C. <ul style="list-style-type: none"> <li>• DC—Domain component</li> <li>• CN—Common name</li> <li>• OU—Organizational unit name</li> <li>• O—Organization name</li> <li>• SN—Serial number of the device</li> </ul>



If you define SN in the subject field without the serial number, then the serial number is read directly from the device and added to the certificate signing request (CSR).

- ST—State
- C—Country

## Required Privilege Level

maintenance and security

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

command-name

```
user@host> request security pki node-local local-certificate enroll cmpv2 ca-profile root-552 ca-
dn DC=example,CN=root-552 certificate-id tc552 email tc552-root@example.net domain-name
example.net ip-address 10.192.0.22 ca-secret example ca-reference 51892 subject
CN=example,OU=SBU,O=552-22
```

Certificate enrollment has started. To view the status of your enrollment, check the public key infrastructure log (pkid) log file at /var/log/pkid.

## Release Information

Command introduced in Junos OS Release 22.3R1.



## RELATED DOCUMENTATION

*Multinode High Availability*

*show security pki node-local local-certificate*

*request security pki node-local local-certificate load*

[request security pki node-local local-certificate export](#)

*request security pki node-local local-certificate re-enroll*

# request security pki node-local key-pair export

## IN THIS SECTION

- [Syntax | 312](#)
- [Description | 312](#)
- [Options | 313](#)
- [Required Privilege Level | 313](#)
- [Output Fields | 313](#)
- [Release Information | 313](#)

## Syntax

```
request security pki node-local key-pair export certificate-id certificate-id filename filename
<passphrase string>
< type (der | pem)>
```

## Description

Export the keypair for an end-entity (EE) certificate. Junos OS encrypts the exported keypair.

You can export the PKI key-pairs file as a backup or to check the file for troubleshooting purposes.



We recommend providing permission to the `request security pki node-local key-pair export` command only to the privileged users.

## Options

<b>certificate-id</b> <i>certificate-id</i>	Name of the local digital certificate.
<b>filename</b> <i>filename</i>	Target directory location and filename of the CA digital certificate.
<b>passphrase</b> <i>passphrase</i>	(Optional) Passphrase to protect the keypair data for PEM format. The passphrase can be up to 64 characters. If specified, the passphrase must be used when importing the keypair.
<b>type</b> ( <i>der   pem</i> )	(Optional) Type of format, either DER or PEM. PEM is the default.

## Required Privilege Level

maintenance

## Output Fields

This command produces no output.

## Release Information

Command introduced in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

---

*Multinode High Availability*

---

*request security pki node-local generate-key-pair*

---

*clear security pki node-local key-pair*



# request security pki node-local generate-key-pair

## IN THIS SECTION

- Syntax | 314
- Description | 314
- Options | 314
- Required Privilege Level | 315
- Output Fields | 315
- Sample Output | 316
- Release Information | 316

## Syntax

```
request security pki node-local generate-key-pair certificate-id certificate-id-name
<size (256 | 384 | 521 | 1024 | 2048 | 4096)>
<type (dsa | ecdsa | rsa)>
```

## Description

Generate a public key infrastructure (PKI) public/private key pair for a local digital certificate on the local node in a Multinode High Availability setup.

## Options

- |   |  |
|---|--|
| <b>certificate-id</b><br><i>certificate-id-name</i> | Name of the local digital certificate and the public/private key pair.           |
| <b>size</b>   | Key pair size. The key pair size can be 256, 384, 521, 1024, 2048, or 4096 bits. |



#### Key size compatibility

- ECDSA-256, 384, and 521
- DSA and RSA - 1024, 2048, or 4096. The default key pair size is 1024 for DSA and 2048 for RSA.

When you use ECDSA-521 signatures, you can:

- Load a complete certificate, which is generated using an external tool like OpenSSL into PKI.
- Manually generate a Certificate Signing Request (CSR) for a local certificate and sending the CSR to a (Certificate Authority) CA server to enroll.
- Automatic enroll with CA server.

#### type

The algorithm to be used for encrypting the public/private key pair:

- `ecdsa`—ECDSA encryption
- `dsa`— DSA encryption
- `rsa`—RSA encryption (default)

## Required Privilege Level

maintenance

## Output Fields

When you enter this command, you are provided feedback on the status of your request.



## Sample Output

**request security pki generate-key-pair**

```
user@host> request security pki node-local generate-key-pair certificate-id cert-123 type rsa
size 4096
Generated key pair cert-123, key size [4096] bits
```

## Release Information

Command introduced in Junos OS Release 22.3R1.

### RELATED DOCUMENTATION

---

*Multinode High Availability*

---

*clear security pki node-local key-pair*

---

*request security pki node-local key-pair export*

# request security pki node-local generate-certificate-request

### IN THIS SECTION

- [Syntax | 317](#)
- [Description | 317](#)
- [Options | 317](#)
- [Required Privilege Level | 318](#)
- [Output Fields | 318](#)
- [Sample Output | 318](#)



## Syntax

```
request security pki node-local generate-certificate-request certificate-id certificate-id-name
domain-name domain-name subject subject-distinguished-name
<digest (sha1 | sha256)>
<email email-address>
<filename (path | terminal)>
<ip-address ip-address>
```

## Description

Manually generate a local digital certificate request in the Public-Key Cryptography Standards #10 (PKCS-10) format on the local device in a Multinode High Availability setup.

## Options

<b>certificate-id</b> <i>certificate-id-name</i>	Name of the local digital certificate and the public/private key pair.
<b>domain-name</b> <i>domain-name</i>	Fully qualified domain name (FQDN) provides the identity of the certificate owner for Internet Key Exchange (IKE) negotiations and provides an alternative to the subject name.
<b>subject</b> <i>subject-distinguished-name</i>	Distinguished name format contains the following information: <ul style="list-style-type: none"> <li>• DC—Domain component</li> <li>• CN—Common name</li> <li>• OU—Organizational unit name</li> <li>• O—Organization name</li> </ul>



- L—Locality
- ST—State
- C—Country

<b>digest</b>	(Optional) Hash algorithm used to sign the certificate request.
	<ul style="list-style-type: none"> <li>• sha-1—SHA-1 digests (default value for RSA or DSA only).</li> <li>• sha-256—SHA-256 digests for RSA or ECDSA only (default value for ECDSA).</li> <li>• sha-384—SHA-384 digests for ECDSA only.</li> </ul>
<b>email</b> <i>email-address</i>	(Optional) E-mail address of the certificate holder.
<b>filename</b> ( <i>path</i>   <b>terminal</b> )	(Optional) Location where the local digital certificate request should be placed or the login terminal.
<b>ip-address</b> <i>ip-address</i>	(Optional) IP address of the router.

## Required Privilege Level

maintenance

## Output Fields

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

**request security pki generate-certificate-request**

```
user@host> request security pki node-local generate-certificate-request certificate-id local-entrust2 domain-name router2.example.net filename entrust-req2 subject cn=router2.example.net
```



Generated certificate request

-----BEGIN CERTIFICATE REQUEST-----

```
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHxLmp1bm1wZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFk1Qws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8o0Xyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjA0BgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AA0BgQBC2rq1v5SQXh7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH406gQ3G
3iH0Zfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ70f09Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
```

-----END CERTIFICATE REQUEST-----

Fingerprint:

0d:90:b8:d2:56:74:fc:84:59:62:b9:78:71:9c:e4:9c:54:ba:16:97 (sha1)

1b:08:d4:f7:90:f1:c4:39:08:c9:de:76:00:86:62:b8 (md5)

## Release Information

Command introduced in Junos OS Release 22.3R1

### RELATED DOCUMENTATION

*Multinode High Availability*