JUNIPER NETWORKS | Engineering Simplicity

Junos® OS

High Availability User Guide

Published
2025-12-15

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

Junos® OS High Availability User Guide

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About This Guide

Use this guide to configure high availability features like ISSU, GRES, and BFD on a Junos OS device.

PART 1

Overview

Understanding High Availability Features on Juniper Networks Routers

IN THIS SECTION

For Juniper Networks routing platforms running the Junos operating system (Junos OS), *high availability* refers to the hardware and software components that provide redundancy and reliability for packet-based communications. This topic provides brief overviews of the following high availability features:

Routing Engine Redundancy

Redundant Routing Engines are two Routing Engines that are installed in the same routing platform. One functions as the primary, while the other stands by as a backup should the primary Routing Engine fail. On routing platforms with dual Routing Engines, network reconvergence takes place more quickly than on routing platforms with a single Routing Engine.

Graceful Routing Engine Switchover

Graceful Routing Engine switchover (GRES) enables a routing platform with redundant Routing Engines to continue forwarding packets, even if one Routing Engine fails. Graceful Routing Engine switchover preserves interface and kernel information. Traffic is not interrupted. However, graceful Routing Engine switchover does not preserve the control plane. Neighboring routers detect that the router has experienced a restart and react to the event in a manner prescribed by individual routing protocol specifications.

NOTE: To preserve routing during a switchover, graceful Routing Engine switchover must be combined with either graceful restart protocol extensions or *nonstop active routing*. For more information, see Understanding Graceful Routing Engine Switchover and Nonstop Active Routing Concepts.

Nonstop Bridging

Nonstop bridging enables a device to switch from a primary Routing Engine to a backup Routing Engine without losing Layer 2 Control Protocol (L2CP) information. Nonstop bridging uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. However, nonstop bridging also saves L2CP information by running the Layer 2 Control Protocol process (l2cpd) on the backup Routing Engine.

NOTE: To use nonstop bridging, you must first enable graceful Routing Engine switchover.

Nonstop bridging is supported for the following Layer 2 control protocols:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Multiple Spanning Tree Protocol (MSTP)
- VLAN Spanning Tree Protocol (VSTP)

Nonstop Active Routing

Nonstop active routing (NSR) enables a routing platform with redundant Routing Engines to switch from a primary Routing Engine to a backup Routing Engine without alerting peer nodes that a change has occurred. Nonstop active routing uses the same infrastructure as graceful Routing Engine switchover to preserve interface and kernel information. However, nonstop active routing also preserves routing information and protocol sessions by running the routing protocol process (rpd) on both Routing Engines. In addition, nonstop active routing preserves TCP connections maintained in the kernel.

NOTE: To use nonstop active routing, you must also configure graceful Routing Engine switchover.

For a list of protocols and features supported by nonstop active routing, see Nonstop Active Routing Protocol and Feature Support.

For more information about nonstop active routing, see Nonstop Active Routing Concepts.

Graceful Restart

With routing protocols, any service interruption requires an affected router to recalculate adjacencies with neighboring routers, restore routing table entries, and update other protocol-specific information. An unprotected restart of a router can result in forwarding delays, route flapping, wait times stemming from protocol reconvergence, and even dropped packets. To alleviate this situation, graceful restart provides extensions to routing protocols. These protocol extensions define two roles for a router—*restarting* and *helper*. The extensions signal neighboring routers about a router undergoing a restart and prevent the neighbors from propagating the change in state to the network during a graceful restart wait interval. The main benefits of graceful restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful restart enables a router to pass through intermediate convergence states that are hidden from the rest of the network.

When a router is running graceful restart and the router stops sending and replying to protocol liveness messages (hellos), the adjacencies assume a graceful restart and begin running a timer to monitor the restarting router. During this interval, helper routers do not process an adjacency change for the router that they assume is restarting, but continue active routing with the rest of the network. The helper routers assume that the router can continue stateful forwarding based on the last preserved routing state during the restart.

If the router was actually restarting and is back up before the graceful timer period expires in all of the helper routers, the helper routers provide the router with the routing table, topology table, or label table (depending on the protocol), exit the graceful period, and return to normal network routing.

If the router does not complete its negotiation with helper routers before the graceful timer period expires in all of the helper routers, the helper routers process the router's change in state and send routing updates, so that convergence occurs across the network. If a helper router detects a link failure from the router, the topology change causes the helper router to exit the graceful wait period and to send routing updates, so that network convergence occurs.

To enable a router to undergo a graceful restart, you must include the `graceful-restart` statement at the global `[edit routing-options]` or `[edit routing-instances` *`instance-name`* `routing-options]` hierarchy level. You can, optionally, modify the global settings at the individual protocol level. When a routing session is started, a router that is configured with graceful restart must negotiate with its neighbors to support it when it undergoes a graceful restart. A neighboring router will accept the negotiation and support helper mode without requiring graceful restart to be configured on the neighboring router.

NOTE: A Routing Engine switchover event on a helper router that is in graceful wait state causes the router to drop the wait state and to propagate the adjacency's state change to the network.

Graceful restart is supported for the following protocols and applications:

- BGP
- ES-IS
- IS-IS
- OSPF/OSPFv3
- PIM sparse mode
- RIP/RIPng
- MPLS-related protocols, including:
  - Label Distribution Protocol (LDP)
  - Resource Reservation Protocol (RSVP)
  - Circuit cross-connect (CCC)
  - Translational cross-connect (TCC)
- Layer 2 and Layer 3 virtual private networks (VPNs)

Nonstop Active Routing Versus Graceful Restart

Nonstop active routing and graceful restart are two different methods of maintaining high availability. Graceful restart requires a router restart. A router undergoing a graceful restart relies on its neighbors (or helpers) to restore its routing protocol information. The restart is the mechanism by which helpers are signaled to exit the wait interval and start providing routing information to the restarting router.

In contrast, nonstop active routing does not involve a router restart. Both the primary and backup Routing Engines are running the routing protocol process (rpd) and exchanging updates with neighbors. When one Routing Engine fails, the router simply switches to the active Routing Engine to exchange routing information with neighbors. Because of these feature differences, nonstop routing and graceful restart are mutually exclusive. Nonstop active routing cannot be enabled when the router is configured as a graceful restarting router. If you include the `graceful-restart` statement at any hierarchy level and the `nonstop-routing` statement at the `[edit routing-options]` hierarchy level and try to commit the configuration, the commit request fails. For more information, see Nonstop Active Routing Concepts.

Effects of a Routing Engine Switchover

Effects of a Routing Engine Switchover describes the effects of a Routing Engine switchover when no high availability features are enabled and when graceful Routing Engine switchover, graceful restart, and nonstop active routing features are enabled.

VRRP

The Virtual Router Redundancy Protocol (VRRP) enables hosts on a LAN to make use of redundant routing platforms (primary and backup pairs) on the LAN, requiring only the static configuration of a single default route on the hosts.

The VRRP routing platform pairs share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routing platforms is the primary (active) and the others are backups. If the primary fails, one of the backup routers or switches becomes the new primary router.

VRRP has advantages in ease of administration and network throughput and reliability:

- It provides a virtual default routing platform.
- It enables traffic on the LAN to be routed without a single point of failure.
- A virtual backup router can take over a failed default router:

- Within a few seconds.
- With a minimum of VRRP traffic.
- Without any interaction with the hosts.

Devices running VRRP dynamically elect primary and backup routers. You can also force assignment of primary and backup routers using priorities from 1 through 255, with 255 being the highest priority.

In VRRP operation, the default primary router sends advertisements to backup routers at regular intervals (default 1 second). If a backup router does not receive an advertisement for a set period, the backup router with the next highest priority takes over as primary and begins forwarding packets.

VRRP nonstop active routing (NSR) is enabled only when you configure the `nonstop-routing` statement at the `[edit routing-options]` or `[edit logical system *logical-system-name* routing-options]` hierarchy level.

For more information, see Understanding VRRP.

Unified ISSU

A unified in-service software upgrade (unified ISSU) enables you to upgrade between two different Junos OS Releases with no disruption on the control plane and with minimal disruption of traffic. Unified ISSU is only supported by dual Routing Engine platforms. In addition, graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) must be enabled.

With a unified ISSU, you can eliminate network downtime, reduce operating costs, and deliver higher services levels. For more information, see Getting Started with Unified In-Service Software Upgrade.

Interchassis Redundancy for MX Series Routers Using Virtual Chassis

Interchassis redundancy is a high availability feature that can span equipment located across multiple geographies to prevent network outages and protect routers against access link failures, uplink failures, and wholesale chassis failures without visibly disrupting the attached subscribers or increasing the network management burden for service providers. As more high-priority voice and video traffic is carried on the network, interchassis redundancy has become a requirement for providing stateful redundancy on broadband subscriber management equipment such as broadband services routers, broadband network gateways, and broadband remote access servers. Interchassis redundancy support enables service providers to fulfill strict service-level agreements (SLAs) and avoid unplanned network outages to better meet the needs of their customers.

To provide a stateful interchassis redundancy solution for MX Series routers, you can configure a *Virtual Chassis*. A *Virtual Chassis* configuration interconnects two MX Series routers into a logical system that you can manage as a single network element. The member routers in a Virtual Chassis are designated as the *primary router* (also known as the *protocol primary*) and the *backup router* (also known as the *protocol backup*). The member routers are interconnected by means of dedicated *Virtual Chassis ports* that you configure on Trio Modular Port Concentrator/Modular Interface Card (MPC/MIC) interfaces.

An MX Series Virtual Chassis is managed by the *Virtual Chassis Control Protocol* (*VCCP*), which is a dedicated control protocol based on IS-IS. VCCP runs on the Virtual Chassis port interfaces and is responsible for building the Virtual Chassis topology, electing the Virtual Chassis primary router, and establishing the interchassis routing table to route traffic within the Virtual Chassis.

Graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) must be enabled on both member routers in the Virtual Chassis.

RELATED DOCUMENTATION

High Availability-Related Features in Junos OS | **8**

High Availability-Related Features in Junos OS

Related redundancy and reliability features include:

- Redundant power supplies, host modules, host subsystems, and forwarding boards. For more information on your specific device, see the Junos Documentation by Product.
- Additional link-layer redundancy, including Multiplex Section Protection (MSP) for SDH interfaces, and DLSw redundancy for Ethernet interfaces. For more information, see the Ethernet Interfaces User Guide for Routing Devices.
- Bidirectional Forwarding Detection (BFD) works with other routing protocols to detect failures rapidly. For more information, see the Understanding How BFD Detects Network Failures.
- Redirection of Multiprotocol Label Switching (MPLS) label-switched path (LSP) traffic—Mechanisms such as link protection, node-link protection, and fast reroute recognize link and node failures, allowing MPLS LSPs to select a bypass LSP to circumvent failed links or devices. For more information, see the MPLS Applications User Guide.

RELATED DOCUMENTATION

Understanding High Availability Features on Juniper Networks Routers | 2

High Availability Features for EX Series Switches Overview

IN THIS SECTION

High availability refers to the hardware and software components that provide redundancy and reliability for network communications. This topic covers the following high availability features of Juniper Networks EX Series Ethernet Switches:

VRRP

You can configure Virtual Router Redundancy Protocol (VRRP) for IP and IPv6 on most switch interfaces, including Gigabit Ethernet interfaces, high-speed Gigabit Ethernet uplink interfaces, and logical interfaces. When VRRP is configured, the switches act as virtual routing platforms. VRRP enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routing platforms share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routing platforms is the primary (active) and the others are backups. If the primary routing platform fails,

one of the backup routing platforms becomes the new primary, providing a virtual default routing platform and enabling traffic on the LAN to be routed without relying on a single routing platform. Using VRRP, a backup switch can take over a failed default switch within a few seconds. This is done with minimum loss of VRRP traffic and without any interaction with the hosts.

Graceful Protocol Restart

With standard implementations of routing protocols, any service interruption requires an affected switch to recalculate adjacencies with neighboring switches, restore routing table entries, and update other protocol-specific information. An unprotected restart of a switch can result in forwarding delays, route flapping, wait times stemming from protocol reconvergence, and even dropped packets. Graceful protocol restart enables a restarting switch and its neighbors to continue forwarding packets without disrupting network performance. Because neighboring switches assist in the restart (these neighbors are called helper switches), the restarting switch can quickly resume full operation without recalculating algorithms from scratch.

On the switches, graceful protocol restart can be applied to aggregate and static routes and for routing protocols (BGP, IS-IS, OSPF, and RIP).

Graceful protocol restart works similarly for the different routing protocols. The main benefits of graceful protocol restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful protocol restart thus allows a switch to pass through intermediate convergence states that are hidden from the rest of the network. Most graceful restart implementations define two types of switches—the restarting switch and the helper switch. The restarting switch requires rapid restoration of forwarding state information so that it can resume the forwarding of network traffic. The helper switch assists the restarting switch in this process. Individual graceful restart configuration statements typically apply to either the restarting switch or the helper switch.

Redundant Routing Engines

Redundant Routing Engines are two Routing Engines that are installed in a switch or a *Virtual Chassis*. When a switch has two Routing Engines, one functions as the primary, while the other stands by as a backup in case the primary Routing Engine fails. When a Virtual Chassis has two Routing Engines, the switch in the primary role functions as the primary Routing Engine and the switch in the backup role functions as the backup Routing Engine. Redundant Routing Engines are supported on all EX Series Virtual Chassis configurations.

The primary Routing Engine receives and transmits routing information, builds and maintains routing tables, communicates with interfaces and Packet Forwarding Engine components of the switch, and has full control over the control plane of the switch.

The backup Routing Engine stays in sync with the primary Routing Engine in terms of protocol states, forwarding tables, and so forth. If the primary becomes unavailable, the backup Routing Engine takes over the functions that the primary Routing Engine performs.

Network reconvergence takes place more quickly on switches and on Virtual Chassis with redundant Routing Engines than on switches and on Virtual Chassis with a single Routing Engine.

Virtual Chassis

A Virtual Chassis is multiple switches connected together that operate as a single network entity. The advantages of connecting multiple switches into a Virtual Chassis include better-managed bandwidth at a network layer, simplified configuration and maintenance because multiple devices can be managed as a single device, a simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP), and improved fault tolerance and high availability. A Virtual Chassis improves high availability for the following reasons:

- Dual Routing Engine support. A Virtual Chassis automatically has two Routing Engines—the switches in the primary and backup `routing-engine` roles—and, therefore, provides more high availability options than standalone switches. Many high availability features, including graceful protocol restart, graceful Routing Engine switchover (GRES), nonstop software upgrade (NSSU), nonstop active routing (NSR), and nonstop bridging (NSB), are available for an EX Series Virtual Chassis that are not available on standalone EX Series switches.

- Increased fault tolerance. You increase your fault tolerance options when you configure your EX Series switches into a Virtual Chassis. You can, for instance, configure interfaces into a link aggregation group (LAG) with member interfaces on different member switches in the same Virtual Chassis to ensure network traffic is received by a Virtual Chassis even when a switch or physical interface in the Virtual Chassis fails.

Graceful Routing Engine Switchover

You can configure *graceful Routing Engine switchover* (GRES) on a switch with redundant Routing Engines or on a Virtual Chassis, allowing control to switch from the primary Routing Engine to the backup Routing Engine with minimal interruption to network communications. When you configure GRES, the backup Routing Engine automatically synchronizes with the primary Routing Engine to preserve kernel state information and forwarding state. Any updates to the primary Routing Engine are replicated to the backup Routing Engine as soon as they occur. If the kernel on the primary Routing Engine stops operating, the primary Routing Engine experiences a hardware failure, or the administrator initiates a manual switchover, primary role switches to the backup Routing Engine.

When the backup Routing Engine assumes primary role in a redundant failover configuration (that is, when GRES is not enabled), the Packet Forwarding Engines initialize their state to the boot state before they connect to the new primary Routing Engine. In contrast, in a GRES configuration, the Packet Forwarding Engines do not reinitialize their state, but resynchronize their state to that of the new primary Routing Engine. The interruption to traffic is minimal.

Link Aggregation

You can combine multiple physical Ethernet ports to form a logical point-to-point link, known as a link aggregation group (LAG) or bundle. A LAG provides more bandwidth than a single Ethernet link can provide. Additionally, link aggregation provides network redundancy by load-balancing traffic across all available links. If one of the links should fail, the system automatically load-balances traffic across all remaining links. In a Virtual Chassis, LAGs can be used to load-balance network traffic between member switches, which increases high availability by ensuring that network traffic is received by the Virtual Chassis even if a single interface fails for any reason.

The number of Ethernet interfaces you can include in a LAG and the number of LAGs you can configure on a switch depend on the switch model.

Nonstop Active Routing and Nonstop Bridging

Nonstop active routing (NSR) provides high availability in a switch with redundant Routing Engines by enabling transparent switchover of the Routing Engines without requiring restart of supported Layer 3 routing protocols. Both Routing Engines are fully active in processing protocol sessions, and so each can take over for the other. The switchover is transparent to neighbor routing devices, which do not detect that a change has occurred.

Nonstop bridging (NSB) provides the same mechanism for Layer 2 protocols. NSB provides high availability in a switch with redundant Routing Engines by enabling transparent switchover of the Routing Engines without requiring restart of supported Layer 2 protocols. Both Routing Engines are fully active in processing protocol sessions, and so each can take over for the other. The switchover is transparent to neighbor switching devices, which do not detect that a change has occurred.

To use NSR or NSB, you must also configure GRES.

Nonstop Software Upgrade

Nonstop software upgrade (NSSU) allows you to upgrade the software on a switch with dual Routing Engines or on a Virtual Chassis in an automated manner with minimal traffic disruption. NSSU takes advantage of GRES and NSR to enable upgrading the Junos OS version with no disruption to the control plane. In addition, NSSU minimizes traffic disruption by:

- Upgrading line cards one at a time in a Virtual Chassis, permitting traffic to continue to flow through the line cards that are not being upgraded.
- Upgrading member switches one at a time in all other Virtual Chassis, permitting traffic to continue to flow through the members that are not being upgraded.

By configuring LAGs such that the member links reside on different line cards or Virtual Chassis members, you can achieve minimal traffic disruption when performing an NSSU.

Redundant Power System

Most Juniper Networks Ethernet Switches have a built-in capability for redundant power supplies—therefore if one power supply fails on those switches, the other power supply takes over. .

RELATED DOCUMENTATION

Junos OS High Availability Configuration Guide

Understanding EX Series Virtual Chassis

Understanding Nonstop Software Upgrade on EX Series Switches | **558**

2

PART

Configuring Switching Control Board Redundancy

Understanding Switching Control Board Redundancy

SUMMARY

Switching control board redundancy allows your device to continue routing and switching functions if a primary control board fails.

IN THIS SECTION

NOTE: In this section, the term *failover* refers to an automatic event, whereas *switchover* refers to either an automatic or a manual event.

Redundant CFEBs

On devices that support this feature, the CFEB performs the following functions:

- Route lookups—Performs route lookups using the forwarding table stored in synchronous SRAM (SSRAM).
- Management of shared memory—Uniformly allocates incoming data packets throughout the router's shared memory.
- Transfer of outgoing data packets—Passes data packets to the destination Fixed Interface Card (FIC) or *Physical Interface Card* (PIC) when the data is ready to be transmitted.
- Transfer of exception and control packets—Passes exception packets to the microprocessor on the CFEB, which processes almost all of them. The remainder are sent to the Routing Engine for further processing. Any errors originating in the Packet Forwarding Engine and detected by the CFEB are sent to the Routing Engine using system log messages.

Devices that support this feature have two CFEBs, one that is configured to act as the primary and the other that serves as a backup in case the primary fails. You can initiate a manual switchover by issuing the `request chassis cfeb master switch` command.

Redundant FEBs

Devices that support this feature supports up to six Forwarding Engine Boards (FEBs). Flexible PIC Concentrator (FPCs), which host PICs, are separate from the FEBs, which handle packet forwarding. FPCs are located on the front of the chassis and provide power and management to PICs through the midplane. FEBs are located on the back of the chassis and receive signals from the midplane, which the FEBs process for packet forwarding. The midplane allows any FEB to carry traffic for any FPC.

To configure the mapping of FPCs to FEBs, use the `fpc-feb-connectivity` statement. You cannot specify a connection between an FPC and a FEB configured as a backup. If an FPC is not specified to connect to a FEB, the FPC is assigned automatically to the FEB with the same slot number. For example, the FPC in slot 1 is assigned to the FEB in slot 1.

You can configure one FEB as a backup for one or more FEBs by configuring a FEB redundancy group. When a FEB fails, the backup FEB can quickly take over packet forwarding. A redundancy group must contain exactly one backup FEB and can optionally contain one primary FEB and multiple other FEBs. A FEB can belong to only one group. A group can provide backup on a one-to-one basis (primary-to-backup), a many-to-one basis (two or more other-FEBs-to-backup), or a combination of both (one primary-to-backup and one or more other-FEBs-to-backup).

When you configure a primary FEB in a redundancy group, the backup FEB mirrors the exact forwarding state of the primary FEB. If switchover occurs from a primary FEB, the backup FEB does not reboot. A manual switchover from the primary FEB to the backup FEB results in less than 1 second of traffic loss. Failover from the primary FEB to the backup FEB results in less than 10 seconds of traffic loss.

If a failover occurs from the other FEB and a primary FEB is specified for the group, the backup FEB reboots so that the forwarding state from the other FEB can be downloaded to the backup FEB and forwarding can continue. Automatic failover from a FEB that is not specified as a primary FEB results in higher packet loss. The duration of packet loss depends on the number of interfaces and on the size of the routing table, but it can be minutes.

If a failover from a FEB occurs when no primary FEB is specified in the redundancy group, the backup FEB does not reboot and the interfaces on the FPC connected to the previously active FEB remain online. The backup FEB must obtain the entire forwarding state from the Routing Engine after a switchover, and this update may take a few minutes. If you do not want the interfaces to remain online during the switchover for the other FEB, configure a primary FEB for the redundancy group.

Failover to a backup FEB occurs automatically if a FEB in a redundancy group fails. You can disable automatic failover for any redundancy group by including the `no-auto-failover` statement at the `[edit chassis redundancy feb redundancy-group` *`group-name`*`]` hierarchy level.

You can also initiate a manual switchover by issuing the `request chassis redundancy feb slot` *`slot-number`* `switch-to-backup` command, where ***slot-number*** is the number of the active FEB. For more information, see the CLI Explorer.

The following conditions result in failover as long as the backup FEB in a redundancy group is available:

- The FEB is absent.
- The FEB experienced a hard error while coming online.
- A software failure on the FEB resulted in a crash.
- Ethernet connectivity from a FEB to a Routing Engine failed.
- A hard error on the FEB, such as a power failure, occurred.
- The FEB was disabled when the offline button for the FEB was pressed.
- The software watchdog timer on the FEB expired.
- Errors occurred on the links between all the active fabric planes and the FEB. This situation results in failover to the backup FEB if it has at least one valid fabric link.
- Errors occurred on the link between the FEB and all of the FPCs connected to it.

After a switchover occurs, a backup FEB is no longer available for the redundancy group. You can revert from the backup FEB to the previously active FEB by issuing the *operational mode command* **request chassis redundancy feb slot *slot-number* revert-from-backup**, where ***slot-number*** is the number of the previously active FEB. For more information, see the CLI Explorer.

When you revert from the backup FEB, it becomes available again for a switchover. If the redundancy group does not have a primary FEB, the backup FEB reboots after you revert back to the previously active FEB. If the FEB to which you revert back is not a primary FEB, the backup FEB is rebooted so that it can aligned with the state of the primary FEB.

If you modify the configuration for an existing redundancy group so that a FEB connects to a different FPC, the FEB is rebooted unless the FEB was already connected to one or two Type 1 FPCs and the change only resulted in the FEB being connected either to one additional or one fewer Type 1 FPC. For more information about how to map a connection between an FPC and a FEB, see the Junos OS Administration Library for Routing Devices. If you change the primary FEB in a redundancy group, the backup FEB is rebooted. The FEB is also rebooted if you change a backup FEB to a nonbackup FEB or change an active FEB to a backup FEB.

To view the status of configured FEB redundancy groups, issue the `show chassis redundancy feb` operational mode command. For more information, see the CLI Explorer.

Redundant SSBs

The System and Switch Board (SSB) on devices that support this feature performs the following major functions:

- Shared memory management on the FPCs—The Distributed Buffer Manager ASIC on the SSB uniformly allocates incoming data packets throughout shared memory on the FPCs.
- Outgoing data cell transfer to the FPCs—A second Distributed Buffer Manager ASIC on the SSB passes data cells to the FPCs for packet reassembly when the data is ready to be transmitted.
- Route lookups—The Internet Processor ASIC on the SSB performs route lookups using the forwarding table stored in SSRAM. After performing the lookup, the Internet Processor ASIC informs the midplane of the forwarding decision, and the midplane forwards the decision to the appropriate outgoing interface.
- System component monitoring—The SSB monitors other system components for failure and alarm conditions. It collects statistics from all sensors in the system and relays them to the Routing Engine, which sets the appropriate alarm. For example, if a temperature sensor exceeds the first internally defined threshold, the Routing Engine issues a “high temp” alarm. If the sensor exceeds the second threshold, the Routing Engine initiates a system shutdown.
- Exception and control packet transfer—The Internet Processor ASIC passes exception packets to a microprocessor on the SSB, which processes almost all of them. The remaining packets are sent to the Routing Engine for further processing. Any errors that originate in the Packet Forwarding Engine and are detected by the SSB are sent to the Routing Engine using system log messages.
- FPC reset control—The SSB monitors the operation of the FPCs. If it detects errors in an FPC, the SSB attempts to reset the FPC. After three unsuccessful resets, the SSB takes the FPC offline and informs the Routing Engine. Other FPCs are unaffected, and normal system operation continues.

Devices that support this feature can hold up to two SSBs. One SSB is configured to act as the primary and the other is configured to serve as a backup in case the primary fails. You can initiate a manual switchover by issuing the `request chassis ssb master switch` command. For more information, see the CLI Explorer.

Redundant SFMs

Devices that support this feature have redundant Switching and Forwarding Modules (SFMs). The SFMs contain the Internet Processor II ASIC and two Distributed Buffer Manager ASICs. SFMs ensure that all traffic leaving the FPCs is handled properly. SFMs provide route lookup, filtering, and switching.

You can initiate a manual switchover by issuing the `request chassis sfm master switch` command. For more information, see the CLI Explorer.

RELATED DOCUMENTATION

- Understanding High Availability Features on Juniper Networks Routers | 2
- Understanding Routing Engine Redundancy | 128
- *show chassis redundancy feb*
- *request chassis cb*

Configuring Switching Control Board Redundancy

SUMMARY

Follow the steps below to configure switching control board redundancy.

IN THIS SECTION

Configuring CFEB Redundancy

The Compact Forwarding Engine Board (CFEB) on devices that support this feature provides route lookup, filtering, and switching on incoming data packets, and then directs outbound packets to the appropriate interface for transmission to the network. The CFEB communicates with the Routing Engine using a dedicated 100-Mbps Fast Ethernet link that transfers routing table data from the Routing Engine to the forwarding table in the integrated ASIC. The link is also used to transfer from the CFEB to the Routing Engine routing link-state updates and other packets destined for the router that have been received through the router interfaces.

To configure a CFEB redundancy group, include the following statements at the `[edit chassis redundancy]` hierarchy level:

```
[edit chassis redundancy]
cfeb slot-number (always | preferred);
```

slot-number can be 0 or 1.

always defines the CFEB as the sole device.

preferred defines a preferred CFEB.

To manually switch CFEB primary role, issue the `request chassis cfeb master switch` command. To view CFEB status, issue the `show chassis cfeb` command.

SEE ALSO

> Understanding Switching Control Board Redundancy | 15

Configuring FEB Redundancy

To configure a FEB redundancy group for devices that support this feature, include the following statements at the `[edit chassis redundancy feb]` hierarchy level:

```
[edit chassis redundancy feb]
redundancy-group group-name {
    description description;
    feb slot-number (backup | primary);
    no-auto-failover;
}
```

group-name is the unique name for the redundancy group. The maximum length is 39 alphanumeric characters.

slot-number is the slot number of each FEB you want to include in the redundancy group. The range is from 0 through 5. You must specify exactly one FEB as a backup FEB per redundancy group. Include the **backup** keyword when configuring the backup FEB and make sure that the FEB is not connected to an FPC.

Include the **primary** keyword to optionally specify one primary FEB per redundancy group. When the **primary** keyword is specified for a particular FEB, that FEB is configured for 1:1 redundancy. With 1:1 redundancy, the backup FEB contains the same forwarding state as the primary FEB. When no FEB in the redundancy group is configured as a primary FEB, the redundancy group is configured for *n*:1 redundancy. In this case, the backup FEB has no forwarding state. When a FEB fails, the forwarding state must be downloaded from the Routing Engine to the backup FEB before forwarding continues.

A combination of 1:1 and *n*:1 redundancy is possible when more than two FEBs are present in a group. The backup FEB contains the same forwarding state as the primary FEB, so that when the primary FEB fails, 1:1 failover is in effect. When a nonprimary FEB fails, the backup FEB must be rebooted so that the forwarding state from the nonprimary FEB is installed on the backup FEB before it can continue forwarding.

You can optionally include the `description` statement to describe a redundancy group.

Automatic failover is enabled by default. To disable automatic failover, include the `no-auto-failover` statement. If you disable automatic failover, you can perform only a manual switchover using the operational command **request chassis redundancy feb slot *slot-number* switch-to-backup**.

To view FEB status, issue the `show chassis feb` command. For more information, see the CLI Explorer.

SEE ALSO

> Understanding Switching Control Board Redundancy | **15**

Configuring SFM Redundancy

By default, the Switching and Forwarding Module (SFM) in slot 0 is the primary and the SFM in slot 1 is the backup. To modify the default configuration, include the `sfm` statement at the `[edit chassis redundancy]` hierarchy level:

```
[edit chassis redundancy]
sfm slot-number (always | preferred);
```

always defines the SFM as the sole device.

preferred defines a preferred SFM.

To manually switch primary role between SFMs, issue the `request chassis sfm master switch` command. To view SFM status, issue the `show chassis sfm` command. For more information, see the CLI Explorer.

SEE ALSO

Understanding Switching Control Board Redundancy | 15

Configuring SSB Redundancy

For devices that support this feature with two System and Switch Boards (SSBs), you can configure which SSB is the primary and which is the backup. By default, the SSB in slot 0 is the primary and the SSB in slot 1 is the backup. To modify the default configuration, include the `ssb` statement at the `[edit chassis redundancy]` hierarchy level:

```
[edit chassis redundancy]
ssb slot-number (always | preferred);
```

slot-number is 0 or 1.

always defines the SSB as the sole device.

preferred defines a preferred SSB.

To manually switch primary role between SSBs, issue the `request chassis ssb master switch` command.

To display SSB status information, issue the `show chassis ssb` command. The command output displays the number of times the primary role has changed, the SSB slot number, and the current state of the SSB: primary, backup, or empty. For more information, see the CLI Explorer.

SEE ALSO

Understanding Switching Control Board Redundancy | 15

Configuring the Junos OS to Support Redundancy on Routers Having Multiple Routing Engines or Switching Boards

For routers that have multiple Routing Engines or these multiple switching control boards: Switching and Forwarding Modules (SFMs), System and Switch Boards (SSBs), Forwarding Engine Boards (FEBs), or Compact Forwarding Engine Boards (CFEBs), you can configure redundancy properties.

To configure redundancy, include the following redundancy statements at the `[edit chassis]` hierarchy level:

```
redundancy {
    cfeb slot (always | preferred);
    failover {
        on-disk-failure
        on-loss-of-keepalives;
    }
    feb {
        redundancy-group group-name {
            feb slot-number (backup | primary);
            description description;
            no-auto-failover;
            }
        }
    graceful-switchover;
    keepalive-time seconds;
    routing-engine slot-number (master | backup | disabled);
    sfm slot-number (always | preferred);
    ssb slot-number (always | preferred);
}
```

SEE ALSO

Understanding Routing Engine Redundancy | 128

PART 3

Configuring Bidirectional Forwarding Detection (BFD)

Understanding How BFD Detects Network Failures

SUMMARY

This topic provides an overview of the Bidirectional Forwarding Detection (BFD) protocol and the different types of BFD sessions.

IN THIS SECTION

Understanding BFD

IN THIS SECTION

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. A pair of routing devices exchange BFD packets. The devices send hello packets at a specified, regular interval. The device detects a neighbor failure when the routing device stops receiving a reply after a specified interval.

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

Review the "Platform-Specific BFD Behavior" on page 50 section for notes related to your platform.

Benefits

- Use BFD to check the health of your network.
- BFD works with a wide variety of network environments and topologies.
- The BFD failure detection timers have short time limits, so they provide fast failure detection.
- BFD timers are adaptive. You can adjust them to be more or less aggressive.

Types of BFD Sessions

There are four types of BFD sessions that are based on the source from which BFD packets are sent to the neighbors. The different types of BFD sessions are:

Table 1: Types of BFD essions

Type of BFD Session	Description
Centralized (or non-distributed) BFD	BFD sessions run completely on the Routing Engine.
Distributed BFD	BFD sessions run completely on the FPC CPU.
Inline BFD	BFD sessions run on the FPC software
Hardware-assisted inline BFD	BFD sessions run on the ASIC firmware

Single-Hop and Multihop BFD

- Single-hop BFD—Single-hop BFD in Junos OS runs in centralized mode by default. Single-hop BFD control packets use UDP port 3784.
- Multihop BFD—One desirable application of BFD is to detect connectivity to routing devices that span multiple network hops and follow unpredictable paths. This is known as a multihop session. Multihop BFD control packets use UDP port 4784.

Consider the following when using multihop BFD:

- In a multichassis link aggregation group (MC-LAG) setup, Inter-Chassis Control Protocol (ICCP) uses BFD in multihop mode. Multihop BFD runs in centralized mode in this kind of setup.
- Junos OS does not execute firewall filters that you apply on a loopback interface (lo0) for a multihop BFD session with a delegated anchor FPC. There is an implicit filter on all ingress FPCs to forward packets to the anchor FPC. Therefore, the firewall filter on the loopback interface is not applied on these packets. If you do not want these packets to be forwarded to the anchor FPC, you can configure the `no-delegate-processing` option.

Centralized BFD

In *centralized BFD* mode (also called *non-distributed BFD* mode), the Routing Engine handles BFD.

For both single-hop BFD and multihop BFD, run BFD in non-distributed mode by enabling `routing-options ppm no-delegate-processing` and then running the `clear bfd session` command.

You can see what mode BDF is running in as follows:

```
user@device> show ppm adjacencies detail
Protocol: BFD, Hold time: 6000, IFL-index: 65
Distributed: FALSE
BFD discriminator: 18, BFD routing table index: 0
```

Distributed BFD

IN THIS SECTION

The term *distributed BFD* refers to BFD that runs on the FPC CPU. The Routing Engine creates the BFD sessions and the FPC CPU processes the sessions.

Starting in Junos OS Release 24.3R1, we've introduced distributed mode for BFD (Bidirectional Forwarding Detection) failure detection on vSRX 3.0. With this support, we've also added the dedicated offload CPU feature on vSRX 3.0. The dedicated offload CPU feature reschedules a flow thread and uses the DPDK flow filters on the NIC to move high priority packets (BGP, RIPv2, OSPF, PIM, Multicast, IGMP, Single-Hop BFD, and Multihop BFD) onto the dedicated flow thread. With this, feature packets are processed by their own dedicated flow thread or queue, even in cases where the forwarding plane is oversubscribed and dropping packets.

Because an entire flow thread is removed from forwarding traffic, you might observe a reduction in packet throughput and this performance impact is more pronounced in smaller vSRX 3.0 deployments.

To enable the dedicated offload CPU feature on vSRX 3.0, run the `set security forwarding-options dedicated-offload-cpu` command.

When you configure this feature, the following warning message is displayed in the syslog output:
Warning, you have enabled dedicated-offload-cpu, this will have a performance impact.

Without a dedicated offload CPU, in cases of oversubscription, where either memory or CPU thresholds are reached on the Packet Forwarding Engine and packets are being dropped, the Fast BFD packets might also be dropped, leading to BFD flapping.

To view the Packet Forwarding Engine's current dedicated offload CPU status use the `show security forward-options dedicated-offload-cpu` command. This command displays whether the Packet Forwarding Engine has dedicated offload CPU feature enabled or disabled.

Benefits

The benefits of distributed BFD are mainly in the scaling and performance areas. Distributed BFD:

- Allows for the creation of a larger number of BFD sessions.

- Runs BFD sessions with a shorter transfer/receive timer interval, which can be used to reduce the overall detection time.
- Separates the functionality of BFD from that of the Routing Engine
- A BFD session can stay up during graceful restart, even with an aggressive interval. The minimum interval for Routing Engine-based BFD sessions to survive *graceful Routing Engine switchover (GRES)* is 2500 ms. Distributed BFD sessions have a minimum interval of less than a second.
- Frees up the Routing Engine CPU, which improves scaling and performance for Routing Engine-based applications.
- BFD protocol packets flow even when the Routing Engine CPU is congested.

Dedicated Offload CPU Limitations for vSRX 3.0

- Dedicated offload CPU is supported by NICs using the mlx5 and iavf drivers and only in KVM and ESXi deployments.
- Only 800 series Intel NICs will support dedicated offload CPU
- NICs using the iavf driver currently only support BFD and BGP packets on the dedicated offload CPU.
- Dedicated offload CPU is disabled when using SWRSS due to queue scheduling complexity.
- Configuring dedicated offload CPU while traffic is flowing has a small chance of out of order packet processing, which might cause issues for the current network sessions.

Distributed Configuration and Support

Distributed BFD is not supported for chassis clusters.

To determine if a BFD peer is running distributed BFD, run the `show bfd sessions extensive` command and look for `Remote is control-plane independent` in the command output.

For distributed BFD to work, you need to configure the lo0 interface with unit 0 and the appropriate family.

```
# set interfaces lo0 unit 0 family inet
# set interfaces lo0 unit 0 family inet6
# set interfaces lo0 unit 0 family mpls
```

This is true for the following types of BFD sessions:

- BFD over aggregated Ethernet logical interfaces, both IPv4 and IPv6
- Multihop BFD, both IPv4 and IPv6
- BFD over VLAN interfaces in EX Series switches, both IPv4 and IPv6
- Virtual Circuit Connectivity Verification (VCCV) BFD (Layer 2 circuit, Layer 3 VPN, and VPLS) (MPLS)

NOTE: The distribution of adjacency entry (the IP addresses of adjacent routers) and transmit entry (the IP address of transmitting routers) for a BFD session is asymmetric. This is because an adjacency entry that requires rules might or might not be distributed based on the redirect rule, and the distribution of transmit entries is *not* dependent on the redirect rule.

The term *redirect rule* here denotes the capability of an interface to send protocol redirect messages. See *Disabling the Transmission of Redirect Messages on an Interface*.

Timer Guidelines for BFD

Depending on your network environment, these recommendations might apply:

- The recommended minimum interval for centralised BFD is 300 ms with a `multiplier` of 3, and the reccomended minimum interval for distributed BFD is 100 ms with a `multiplier` of 3.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.
- You can configure a hold-down interval to suppress state change notifications caused by brief session flapping. Use the `bfd-liveness-detection holddown-interval` *`milliseconds`* statement to specify a delay between 0 to 255,000 milliseconds before sending a state change notification. If the BFD session goes down and then comes back up during the hold-down interval, the timer is restarted.

Inline BFD

IN THIS SECTION

We support two types of inline BFD: inline BFD and hardware-assisted inline BFD. *Inline BFD* sessions run on the FPC software. *Hardware-assisted inline BFD* sessions run on the ASIC firmware. Support depends on your device and software version.

Benefits

- Inline BFD sessions can have keepalive intervals of less than a second, so you can detect errors in milliseconds.
- If you are running inline BFD and the Routing Engine crashes, the inline BFD sessions will continue without interruption for 15 seconds.
- Inline BFD has many of the same benefits as distributed BFD since it also separates the functionality of BFD from the Routing Engine.
- The Packet Forwarding Engine software and the ASIC firmware process the packets more quickly than the FPC CPU, so inline BFD is faster than distributed BFD.

Inline BFD

Inline BFD sessions run on the FPC software. The Routing Engine creates the BFD sessions and the Packet Forwarding Engine software processes the sessions. Integrated routing and bridging (IRB) interfaces support inline BFD sessions.

We support inline BFD sessions for both underlay and overlay. For example, you can run BFD sessions between EVPN overlay BGP peers.

We don't support inline BFD sessions over VXLAN tunnels. For example, you can't run inline BFD between BGP peers that are connected through a VXLAN tunnel. To use BFD sessions over a VXLAN tunnel, you must use either distributed mode or centralized mode.

Hardware-Assisted Inline BFD

Hardware-assisted inline BFD sessions run on the ASIC firmware. Hardware-assisted inline BFD is a hardware implementation of the inline BFD protocol. The Routing Engine creates BFD sessions and passes them to the ASIC firmware for processing. The device uses existing paths to forward any BFD events that need to be processed by protocol processes.

Regular inline BFD is a software approach. In hardware-assisted inline BFD, the firmware handles most of the BFD protocol processing. The ASIC firmware processes the packets more quickly than the software, so hardware-assisted inline BFD is faster than regular inline BFD. We support this feature for single-hop and multihop IPv4 and IPv6 BFD sessions.

We support hardware-assisted inline BFD sessions for both underlay and overlay. For example, you can run BFD sessions between EVPN overlay BGP peers.

We don't support hardware-assisted inline BFD sessions over VXLAN tunnels. For example, you can't run hardware-assisted inline BFD between BGP peers that are connected through a VXLAN tunnel. To use BFD sessions over a VXLAN tunnel, you must use either distributed mode or centralized mode.

Limitations

If the Packet Forwarding Engine process restarts or the system reboots, the BFD sessions will go down.

Hardware-assisted inline BFD:

- Does not support micro BFD.
- Is only supported on standalone devices.
- Does not support BFD authentication.
- Does not support IPv6 link local BFD sessions.
- Cannot be used with VXLAN encapsulation of BFD packets.
- Cannot be used with LAG.
- Cannot be used with ECMP on QFX5120 Series devices.

NOTE: When using hardware-assisted BFD with ECMP, if hardware recovery takes more time than the BFD timer, it can cause flapping in the BFD session.

Supported Platforms

The following platforms support hardware-assisted inline BFD:

Platforms	First Supported Release	Default Mode
QFX5120-32C QFX5120-48Y	21.2R1	Hardware-assisted Inline BFD
QFX-5220-32 QFX-5220-128c	23.2R1	Inline BFD
QFX5130-32CD QFX5700	23.4R1	Inline BFD

Configuration

Devices support either regular inline BFD or hardware-assisted inline BFD. Use the `set routing-options ppm inline-processing-enable` command to enable the type of inline BFD that your device supports. To return BFD to the default mode, delete the configuration.

Use the `set routing-options ppm no-delegate-processing` configuration statement to transition from inline mode to centralized mode. If there is a session over VXLAN tunnel, or any other tunnel, you need to set all BFD sessions to run in distributed mode or centralized mode.

BFD Session Damping Overview

IN THIS SECTION

BFD session damping lets you prevent excess BFD flap notifications by damping BFD session state changes for a configured time period if defined thresholds are exceeded.

NOTE: BFD session damping is currently supported for LACP protocol clients only.

Benefits

- Improve network stability by damping intermittent BFD session flaps that can disrupt services.
- Enhance network management by setting thresholds and timers for effective BFD damping control.
- Speed up convergence times by reducing false positives.

Overview

You can use BFD to quickly detect failures in reachabiltiy between devices. When BFD detects a failure, it sends a notification to relevant client and protocols. If an unstable link goes up and down repeatedly, this can cause excessive BFD notifications. You can use BFD session damping to avoid this by automatically damping BFD notifications for a configured time period when flap detection thresholds are exceeded.

If a BFD session transitions between `Up` and `Down` more frequently than the configured flap detection threshold, that session is considered flapping and placed in a dampened state. While dampened, all BFD notifications for that session are suppressed for the duration of the damping timeout period. After the timeout expires, notifications resume for that BFD session. You can configure the flap detection threshold and damping timeout period to suit network your needs. Lower timeout values result in faster restoration of notification for flapping sessions.

Session instablity is measured on a per-BFD-session basis as a value called merit value. Each time a BFD session transitions to a `Down` state, the merit value for that sessions is increased by a configured increment. When the merit value passes a configured threshold, that BFD session will be dampened.

Configuration

Use the `bfd-liveness-detection damping` configuration statement at the `[edit interfaces` *`name`* `aggregated-ether-option]` hierachy level to configure BFD session damping.

You can use a variety of configuration options to set values like the merit threshold for triggering damping,the maximum length of damping time, the value of incremental merit applied after each flap, and more.

BFD session damping happens independently on each router locally, so BFD session configuration values should match on both ends of a BFD session to ensure consistent behavior.

The key configuration options are as follows:

suppress	The merit value above which BFD notifications will be suppressed.
reuse	The merit value below which a suppressed BFD session will start notifications again.
increment	Increments applied to merit value for every flap.
max-suppress-time	The maximum time a BFD session can be suppressed.
half-life	The time duration in seconds after which the accumulated merit value of a BFD session will be reduced by half.

For more information on the default values and ranges for each option, see *damping (BFD Liveness Detection)*.

Understanding BFD for Static Routes for Faster Network Failure Detection

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. BFD works with a wide variety of network environments and topologies. A pair of routing devices exchanges BFD packets. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. The BFD failure detection timers have shorter time limits than the static route failure detection mechanisms, so they provide faster detection.

The BFD failure detection timers can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the `clear bfd adaptation` command to return BFD interval timers to their configured values. The `clear bfd adaptation` command is hitless, meaning that the command does not affect traffic flow on the routing device.

By default, BFD is supported on single-hop static routes.

NOTE: On MX Series devices, multihop BFD is not supported on a static route if the static route is configured with more than one next hop. It is recommended that you avoid using multiple next hops when a multihop BFD is required for a static route.

To enable failure detection, include the `bfd-liveness-detection` statement in the static route configuration.

NOTE: The `bfd-liveness-detection` command includes the description field. On devices that support this feature, the description is an attribute under the **bfd-liveness-detection** object. This field is applicable only for the static routes.

The BFD protocol is supported for IPv6 static routes. Global unicast and link-local IPv6 addresses are supported for static routes. The BFD protocol is not supported on multicast or anycast IPv6 addresses. For IPv6, the BFD protocol supports only static routes. IPv6 for BFD is also supported for the eBGP protocol.

To configure the BFD protocol for IPv6 static routes, include the `bfd-liveness-detection` statement at the `[edit routing-options rib inet6.0 static route` *`destination-prefix`*`]` hierarchy level.

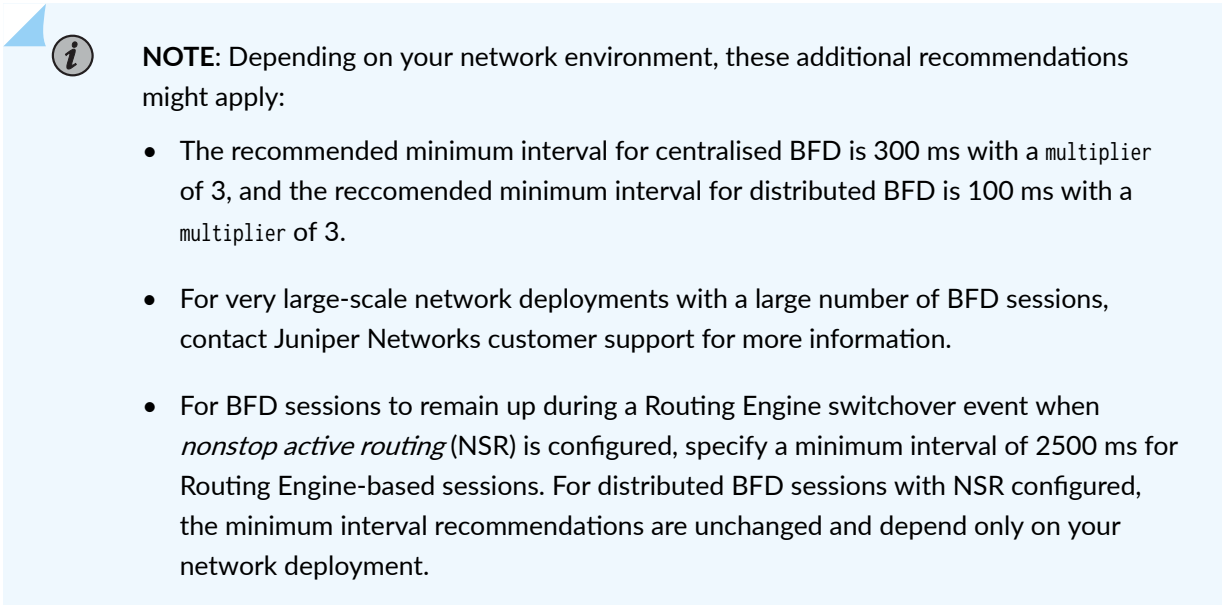
You can configure a hold-down interval to specify how long the BFD session must remain up before a state change notification is sent.

To specify the hold-down interval, include the `holddown-interval` statement in the BFD configuration. You can configure a number in the range from 0 through 255,000 milliseconds. The default is 0. If the BFD session goes down and then comes back up during the hold-down interval, the timer is restarted.

NOTE: If a single BFD session includes multiple static routes, the hold-down interval with the highest value is used.

To specify the minimum transmit and receive intervals for failure detection, include the `minimum-interval` statement in the BFD configuration.

This value represents both the minimum interval after which the local routing device transmits hello packets and the minimum interval after which the routing device expects to receive a reply from the neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds. Optionally, instead of using this statement, you can configure the minimum transmit and receive intervals separately using the **transmit-interval minimum-interval** and `minimum-receive-interval` statements.



NOTE: Depending on your network environment, these additional recommendations might apply:

- The recommended minimum interval for centralised BFD is 300 ms with a `multiplier` of 3, and the reccomended minimum interval for distributed BFD is 100 ms with a `multiplier` of 3.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when *nonstop active routing* (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

To specify the minimum receive interval for failure detection, include the `minimum-receive-interval` statement in the BFD configuration. This value represents the minimum interval after which the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds. Optionally, instead of using this statement, you can configure the minimum receive interval using the `minimum-interval` statement at the `[edit routing-options static route` *`destination-prefix`* `bfd-liveness-detection]` hierarchy level.

To specify the number of hello packets not received by the neighbor that causes the originating interface to be declared down, include the `multiplier` statement in the BFD configuration. The default value is 3. You can configure a number in the range from 1 through 255.

To specify a threshold for detecting the adaptation of the detection time, include the `threshold` statement in the BFD configuration.

When the BFD session detection time adapts to a value equal to or higher than the threshold, a single trap and a system log message are sent. The detection time is based on the multiplier of the **minimum-interval** or the **minimum-receive-interval** value. The threshold must be a higher value than the multiplier for either of these configured values. For example if the **minimum-receive-interval** is 300 ms and the **multiplier** is 3, the total detection time is 900 ms. Therefore, the detection time threshold must have a value higher than 900.

To specify the minimum transmit interval for failure detection, include the `transmit-interval minimum-interval` statement in the BFD configuration.

This value represents the minimum interval after which the local routing device transmits hello packets to the neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds. Optionally, instead of using this statement, you can configure the

minimum transmit interval using the `minimum-interval` statement at the `[edit routing-options static route` *`destination-prefix`* `bfd-liveness-detection]` hierarchy level.

To specify the threshold for the adaptation of the transmit interval, include the `transmit-interval threshold` statement in the BFD configuration.

The threshold value must be greater than the transmit interval. When the BFD session transmit time adapts to a value greater than the threshold, a single trap and a system log message are sent. The detection time is based on the multiplier of the value for the **minimum-interval** or the `minimum-receive-interval` statement at the `[edit routing-options static route` *`destination-prefix`* `bfd-liveness-detection]` hierarchy level. The threshold must be a higher value than the multiplier for either of these configured values.

To specify the BFD version, include the `version` statement in the BFD configuration. The default is to have the version detected automatically.

To include an IP address for the next hop of the BFD session, include the `neighbor` statement in the BFD configuration.

NOTE: You must configure the `neighbor` statement if the next hop specified is an interface name. If you specify an IP address as the next hop, that address is used as the neighbor address for the BFD session.

You can configure BFD sessions not to adapt to changing network conditions. To disable BFD adaptation, include the `no-adaptation` statement in the BFD configuration.

NOTE: We recommend that you not disable BFD adaptation unless it is preferable *not* to have BFD adaptation in your network.

NOTE: If BFD is configured only on one end of a static route, the route is removed from the routing table. BFD establishes a session when BFD is configured on both ends of the static route.

BFD is not supported on ISO address families in static routes. BFD does support IS-IS.

If you configure *graceful Routing Engine switchover* (GRES) at the same time as BFD, GRES does not preserve the BFD state information during a failover.

Understanding BFD for BGP

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. The failure detection timers for BFD have shorter time limits than default failure detection mechanisms for BGP, so they provide faster detection.

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

Review the *Platform-Specific BFD for BGP Behavior* section for notes related to your platform.

NOTE: Configuring both BFD and graceful restart for BGP on the same device is counterproductive. When an interface goes down, BFD detects this instantly, stops traffic forwarding and the BGP session goes down whereas graceful restart forwards traffic despite the interface failure, this behavior might cause network issues. Hence we do not recommend configuring both BFD and graceful restart on the same device.

The BFD failure detection timers can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds (15000 milliseconds). A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the `clear bfd adaptation` command to return BFD interval timers to their configured values. The `clear bfd adaptation` command is hitless, meaning that the command does not affect traffic flow on the routing device.

SEE ALSO

Enabling Dedicated and Real-Time BFD

Understanding BFD for OSPF

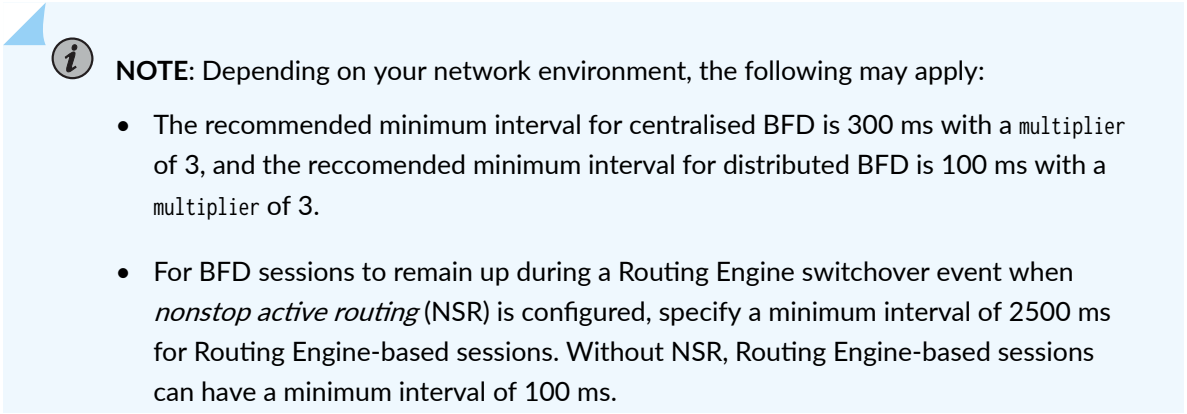
The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. BFD works with a wide variety of network environments and topologies. A pair of routing devices exchange BFD packets. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. The BFD failure

detection timers have shorter time limits than the OSPF failure detection mechanisms, so they provide faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the `clear bfd adaptation` command to return BFD interval timers to their configured values. The `clear bfd adaptation` command is hitless, meaning that the command does not affect traffic flow on the routing device.

You can configure the following BFD protocol settings:

- `detection-time threshold`—Threshold for the adaptation of the detection time. When the BFD session detection time adapts to a value equal to or greater than the configured threshold, a single trap and a single system log message are sent.
- `full-neighbors-only`—Ability to establish BFD sessions only for OSPF neighbors with full neighbor adjacency. The default behavior is to establish BFD sessions for all OSPF neighbors.
- `minimum-interval`—Minimum transmit and receive interval for failure detection. This setting configures both the minimum interval after which the local routing device transmits hello packets and the minimum interval after which the routing device expects to receive a reply from the neighbor with which it has established a BFD session. Both intervals are in milliseconds. You can also specify the minimum transmit and receive intervals separately using the `transmit-interval minimum-interval` and `minimum-receive-interval` statements.



NOTE: Depending on your network environment, the following may apply:

- The recommended minimum interval for centralised BFD is 300 ms with a `multiplier` of 3, and the reccomended minimum interval for distributed BFD is 100 ms with a `multiplier` of 3.
- For BFD sessions to remain up during a Routing Engine switchover event when *nonstop active routing* (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. Without NSR, Routing Engine-based sessions can have a minimum interval of 100 ms.

- For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.
- BFD is not distributed prior to Junos 21.2 (because for OSPFv3, BFD is based in the Routing Engine).

- `minimum-receive-interval`—Minimum receive interval for failure detection. This setting configures the minimum receive interval, in milliseconds, after which the routing device expects to receive a hello packet from a neighbor with which it has established a BFD session. You can also specify the minimum receive interval using the `minimum-interval` statement.
- `multiplier`—Multiplier for hello packets. This setting configures the number of hello packets that are not received by a neighbor, which causes the originating interface to be declared down. By default, three missed hello packets cause the originating interface to be declared down.
- `no-adaptation`—Disables BFD adaption. This setting disables BFD sessions from adapting to changing network conditions.

  NOTE: We recommend that you do not disable BFD adaptation unless it is preferable not to have BFD adaptation in your network.

- `transmit-interval minimum-interval`—Minimum transmit interval for failure detection. This setting configures the minimum transmit interval, in milliseconds, at which the local routing device transmits hello packets to the neighbor with which it has established a BFD session. You can also specify the minimum transmit interval using the `minimum-interval` statement.
- `transmit-interval threshold`—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 log message are sent. The threshold value must be greater than the minimum transmit interval. If you attempt to commit a configuration with a threshold value less than the minimum transmit interval, the routing device displays an error and does not accept the configuration.
- `version`—BFD version. This setting configures the BFD version used for detection. You can explicitly configure BFD version 1, or the routing device can automatically detect the BFD version. By default, the routing device automatically detects the BFD version automatically, which is either 0 or 1.

You can also trace BFD operations for troubleshooting purposes.

Understanding BFD for IS-IS

The Bidirectional Forwarding Detection (BFD) protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. The failure detection timers for BFD have shorter time limits than the failure detection mechanisms of IS-IS, providing faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. For example, the timers can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (RX) interval by two if the local BFD instance is the reason for the session flap. The transmission (TX) interval is increased by two if the remote BFD instance is the reason for the session flap.

You can use the `clear bfd adaptation` command to return BFD interval timers to their configured values. The `clear bfd adaptation` command is hitless, meaning that the command does not affect traffic flow on the routing device.

NOTE: You can configure IS-IS BFD sessions for IPv6 by including the `bfd-liveness-detection` statement at the `[edit protocols isis interface` *`interface-name`* `family inet|inet6]` hierarchy level.

- For interfaces that support both IPv4 and IPv6 routing, the `bfd-liveness-detection` statement must be configured separately for each inet family.
- BFD over IPv6 link local address is currently not distributed because IS-IS uses link local addresses for forming adjacencies.
- BFD sessions over IPv6 must not have the same aggressive detection intervals as IPv4 sessions.
- BFD IPv6 sessions with detection intervals less than 2.5 seconds are currently not supported when nonstop active routing (NSR) is enabled.

To detect failures in the network, the set of statements in Table 2 on page 43 are used in the configuration.

Table 2: Configuring BFD for IS-IS

Statement	Description
`bfd-liveness-detection`	Enable failure detection.
`minimum-interval` *`milliseconds`*	Specify the minimum transmit and receive intervals for failure detection. This value represents the minimum interval at which the local router transmits hellos packets as well as the minimum interval at which the router expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number from 1 through 255,000 milliseconds. You can also specify the minimum transmit and receive intervals separately. **NOTE**: Depending on your network environment, these additional recommendations might apply: • The recommended minimum interval for centralised BFD is 300 ms with a `multiplier` of 3, and the reccomended minimum interval for distributed BFD is 100 ms with a `multiplier` of 3. • For very large-scale network deployments with a large number of BFD sessions, please contact Juniper Networks customer support for more information. • For BFD sessions to remain up during a Routing Engine switchover event when *nonstop active routing* (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with nonstop active routing configured, the minimum interval recommendations are unchanged and depend only on your network deployment.
`minimum-receive-interval` *`milliseconds`*	Specify only the minimum receive interval for failure detection. This value represents the minimum interval at which the local router expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number from 1 through 255,000 milliseconds.
`multiplier` *`number`*	Specify the number of hello packets not received by the neighbor that causes the originating interface to be declared down. The default is 3, and you can configure a value from 1 through 225.

Table 2: Configuring BFD for IS-IS *(Continued)*

Statement	Description
`no-adaptation`	Disable BFD adaptation. You can specify that the BFD sessions not adapt to changing network conditions. **NOTE**: We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.
`threshold`	Specify the threshold for the following: • Adaptation of the detection time When the BFD session detection time adapts to a value equal to or greater than the threshold, a single trap and a system log message are sent. • Transmit interval **NOTE**: The threshold value must be greater than the minimum transmit interval multiplied by the multiplier number.
`transmit-interval minimum-interval`	Specify the minimum transmit interval for failure detection. This value represents the minimum interval at which the local routing device transmits hello packets to the neighbor with which it has established a BFD session. You can configure a value from 1 through 255,000 milliseconds.
`version`	Specify the BFD version used for detection. The default is to have the version detected automatically.

NOTE: You can trace BFD operations by including the `traceoptions` statement at the `[edit protocols bfd]` hierarchy level.

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

Understanding BFD for RIP

The Bidirectional Forwarding Detection (BFD) Protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. BFD failure detection times are shorter than RIP detection times, providing faster reaction times to various kinds of failures in the network. Instead of waiting for the routing protocol neighbor timeout, BFD provides rapid detection of link failures. BFD timers are adaptive and can be adjusted to be more or less aggressive. For example, a timer can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the one configured. Note that the functionality of configuring BFD for RIP described in this topic is not supported in Junos OS Releases 15.1X49, 15.1X49-D30, or 15.1X49-D40.

NOTE: EX4600 and QFX5000 Series switches running Junos OS or Junos OS Evolved do not support minimum interval values of less than 1 second in centralized and distributed mode.

BFD enables quick failover between a primary and a secondary routed path. The protocol tests the operational status of the interface multiple times per second. BFD provides for configuration timers and thresholds for failure detection. For example, if the minimum interval is set for 50 milliseconds and the threshold uses the default value of three missed messages, a failure is detected on an interface within 200 milliseconds of the failure.

Intervening devices (for example, an Ethernet LAN switch) hide link-layer failures from routing protocol peers, such as when two routers are connected by way of a LAN switch, where the local interface status remains up even when a physical fault happens on the remote link. Link-layer failure detection times vary, depending on the physical media and the Layer 2 encapsulation. BFD can provide fast failure detection times for all media types, encapsulations, topologies, and routing protocols.

To enable BFD for RIP, both sides of the connection must receive an update message from the peer. By default, RIP does not export any routes. Therefore, you must enable update messages to be sent by configuring an export policy for routes before a BFD session is triggered.

Understanding Independent Micro BFD Sessions for LAG

IN THIS SECTION

The Bidirectional Forwarding Detection (BFD) protocol is a simple detection protocol that quickly detects failures in the forwarding paths. To enable failure detection for aggregated Ethernet interfaces in a LAG, you can configure an independent, asynchronous-mode BFD session on every LAG member link in a LAG bundle. Instead of a single BFD session monitoring the status of the UDP port, independent micro-BFD sessions monitor the status of individual member links.

When you configure micro-BFD sessions on every member link in a LAG bundle, each individual session determines the Layer 2 and Layer 3 connectivity of each member link in a LAG.

After the individual session is established on a particular link, member links are attached to the LAG and then load balanced by either one of the following:

- Static configuration—The device control process acts as the client to the micro-BFD session.
- Link Aggregation Control Protocol (LACP)—LACP acts as the client to the micro-BFD session.

When the micro-BFD session is up, a LAG link is established and data is transmitted over that LAG link. If the micro-BFD session on a member link is down, that particular member link is removed from the load balancer, and the LAG managers stop directing traffic to that link. These micro-BFD sessions are independent of each other despite having a single client that manages the LAG interface.

Micro-BFD sessions run in the following modes:

- Distribution mode—In this mode, the Packet Forwarding Engine (PFE) sends and receives the packets at Layer 3. By default, micro-BFD sessions are distributed at Layer 3.
- Non-distribution mode—In this mode, the Routing Engine sends and receives the packets at Layer 2. You can configure the BFD session to run in this mode by including the `no-delegate-processing` statement under periodic packet management (PPM).

A pair of routing devices in a LAG exchange BFD packets at a specified, regular interval. The routing device detects a neighbor failure when it stops receiving a reply after a specified interval. This allows the quick verification of member link connectivity with or without LACP. A UDP port distinguishes BFD over LAG packets from BFD over single-hop IP packets. The Internet Assigned Numbers Authority (IANA) has allocated 6784 as the UDP destination port for micro-BFD.

Benefits

- Failure detection for LAG—Enables failure detection between devices that are in point-to-point connections.
- Multiple BFD sessions—Enables you to configure multiple micro-BFD sessions for each member link instead of a single BFD session for the entire bundle.

Configuration Guidelines for Micro-BFD Sessions

Consider the following guidelines as you configure individual micro-BFD sessions on an aggregated Ethernet bundle.

- This feature works only when both the devices support BFD. If BFD is configured at one end of the LAG, this feature does not work.
- IANA has allocated 01-00-5E-90-00-01 as the dedicated MAC address for micro BFD. Dedicated MAC mode is used by default for micro BFD sessions.
- In Junos OS, micro-BFD control packets are always untagged by default. For Layer 2 aggregated interfaces, the configuration must include `vlan-tagging` or `flexible-vlan-tagging` options when you configure Aggregated Ethernet with BFD. Otherwise, the system will throw an error while committing the configuration.
- When you enable micro-BFD on an aggregated Ethernet interface, the aggregated interface can receive micro-BFD packets. In Junos OS Release 19.3 and later, for MPC10E and MPC11E MPCs, you cannot apply firewall filters on the micro-BFD packets received on the aggregated Ethernet interface. For MPC1E through MPC9E, you can apply firewall filters on the micro-BFD packets received on the aggregated Ethernet interface only if the aggregated Ethernet interface is configured as an untagged interface.
- Junos OS checks and validates the configured micro-BFD `local-address` against the interface or loopback IP address before the configuration commit. Junos OS performs this check on both IPv4 and IPv6 micro-BFD address configurations, and if they do not match, the commit fails. The configured micro-BFD local address should match with the micro-BFD neighbour address that you have configured on the peer router.
- For the IPv6 address family, disable duplicate address detection before configuring this feature with aggregated Ethernet interface addresses. To disable duplicate address detection, include the `dad-disable` statement at the `[edit interface` *aex* `unit` *y* `family inet6]` hierarchy level.
- AFT-based line cards (MPC10 and newer) use a different hardware design. If micro BFD is activated on an interface, the received packets won't be part of the interface group for the AE interface and won't match filter terms on lo0.0 with the interface group. To ensure terms match, you can set up a separate filter on lo0.0 using port 6784.

CAUTION: Deactivate `bfd-liveness-detection` at the `[edit interfaces aex aggregated-ether-options]` hierarchy level or deactivate the aggregated Ethernet interface before changing the neighbor address from the loopback IP address to the aggregated Ethernet interface IP address. Modifying the local and neighbor address without deactivating `bfd-liveness-detection` or the aggregated Ethernet interface first might cause micro-BFD sessions failure.

SEE ALSO

authentication

bfd-liveness-detection

detection-time

transmit-interval

Understanding Static Route State When BFD is in Admin Down State

The Bidirectional Forwarding Detection (BFD) Admin Down state is used to bring down a BFD session administratively (applicable for normal BFD session and micro BFD session), to protect client applications from BFD configuration removal, license issues, and clearing of BFD sessions.

When BFD enters the Admin Down state, BFD notifies the new state to its peer for a failure detection time and after the time expires, the client stops transmitting packets.

For the Admin Down state to work, the peer, which receives the Admin Down state notification, must have the capability to distinguish between administratively down state and real link failure.

A BFD session moves to the Admin Down state under the following conditions:

- If BFD configuration is removed for the last client tied to a BFD session, BFD moves to Admin Down state and communicates the change to the peer, to enable the client protocols without going down.
- If BFD license is removed on the client, BFD moves to Admin Down state and communicates the change to the remote system to enable the client protocols without going down.
- When `clear bfd session` command is executed, the BFD sessions move to Admin Down state before restarting. This `clear bfd session` command also ensures that the client applications are not impacted.

Starting from Junos OS 16.1R1 release, you can set the state of static route in BFD Admin Down state by configuring one of the following commands:

- `set routing-options static static-route bfd-admin-down active`—BFD Admin Down state pulls down the static route.
- `set routing-options static static-route bfd-admin-down passive`—BFD Admin Down state does not pull down the static route.

SEE ALSO

Understanding BFD for Static Routes for Faster Network Failure Detection

Understanding Seamless BFD

IN THIS SECTION

Seamless BFD (S-BFD) reduces the time that is required to detect and respond to path failures by speeding up BFD initiation time. Each node in the network is assigned a unique S-BFD discriminator. Nodes operate as either initiators that actively check the reachability of paths, or as responders that listen and respond to reachability requests. When an S-BFD initiator sends a request packet, the responder replies by swapping discriminators and immediately setting the state to `UP`. This stateless operation allows for the quick establishment of multiple sessions, making it ideal for environments requiring rapid connectivity checks.

To enable sbfd, configure the `bfd-liveness-detection sbfd remote-discriminator` *`value`* statement on initator nodes, and the `bfd sbfd local-discriminator` *`value`* on reponder nodes.

Benefits of S-BFD

- **Rapid Failure Detection**: S-BFD allows for immediate connectivity verification without the need for initial handshake messages, enabling network operators to detect and respond to failures at a much faster rate compared to traditional BFD.
- **Reduced Session State Overhead**: The responder in S-BFD does not maintain any session state, which simplifies the network architecture and reduces the overhead associated with maintaining multiple sessions, thus improving the scalability of the network.
- **Fast Failure Detection and Recovery:** S-BFD's ability to quickly detect unidirectional path failures and support fast re-route (FRR) capabilities ensures minimal downtime and rapid recovery, which is crucial for maintaining high network reliability.

S-BFD Triggered Fast Re-Route

Starting in Junos OS and Junos OS Evolved release 23.2R1, S-BFD supports fast re-route (FRR), a feature designed to enhance the reliability and resilience of Segment Routing Traffic Engineering (SR-TE) tunnels. S-BFD monitors end-to-end paths within SR-TE tunnels and promptly initiates local repair mechanisms when failures are detected, rerouting traffic to alternate paths to minimize disruption. The core principle behind FRR is to ensure that network traffic continues to flow seamlessly, even in the event of path disruptions, minimizing downtime and maintaining service continuity.

To enable S-BFD triggered FRR, use the `source-packet-routing sbfd-frr` configuration statement.

Understanding BFD Echo and Echo-Lite Modes

Starting in Junos OS Release 22.4R1, you can configure BFD to send echo packets back and forth from a neighboring device to ensure that a forwarding path is available. Use the `bfd-liveness-detection echo minimum-interval` *`milliseconds`* configuration statement to enable BFD echo mode and set the minimum interval for echo packets. BFD echo mode only works if the neighboring device supports BFD.

If the neighboring device does not support BFD, you can use BFD echo-lite mode. To enable BFD echo-lite mode, use the `bfd-liveness-detection echo-lite minimum-interval` *`milliseconds`* configuration statement. BFD echo-lite mode performs the same function as BFD echo mode without requiring BFD configuration on the neighbor device.

By default, echo and echo-lite modes only support single-hop sessions in centralized BFD mode. Starting in Junos OS Release 24.2R1, PRPD BFD APIs support echo-lite mode for multihop sessions in distributed and inline BFD modes. For more information about PRPD APIs, see Overview of JET APIs. Starting in Junos OS Release 25.4R1, you can configure single-hop BFD echo-lite sessions in inline and distribued mode.

Platform-Specific BFD Behavior

IN THIS SECTION

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

Use the following tables to review platform-specific behaviors for your platform:

Platform-Specific Distributed BFD Behavior

Platform	Difference
ACX Series	• ACX7000 Series devices do not support distributed BFD for OSPFv3, IS-IS IPv6, PIM IPv6, RSVP, LDP, or S-BFD. • ACX7000 Series devices support a `minimum-interval` of 1000ms or higher for distributed and centralized BFD.
MX Series	• MX Series routers only support inline BFD if the router is static and has MPCs/MICs with `enhanced-ip` configured.
PTX Series	• Flapping occurs during the BFD session when the lo0 interface is not configured on PTX Series routers.
QFX Series	• QFX5110, QFX5120, QFX5200, and QFX5210 switches support 10 multihop inline BFD sessions. You can configure them with a timer of 150 x 3 milliseconds. Single-hop sessions are also supported. • Devices support either regular inline BFD or hardware-assisted inline BFD. Starting in Junos OS Release 21.2R1, QFX5120-32C and QFX5120-48Y switches support hardware-assisted inline BFD. They support a timer of 100 x 3 milliseconds. They can run up to 128 hardware-assisted inline BFD sessions, which can be a mix of single-hop and multihop BFD sessions.

(Continued)

Platform	Difference
SRX Series	• Distributed BFD is not supported for chassis clusters. Standalone SRX Series Firewalls support a BFD failure detection time of 3 x 100 ms. • Enable distributed mode on the SRX5000 line of devices with SPC3 line cards and SRX1500, SRX4100, SRX4200, and SRX4600 devices by configuring the BFD failure detection time to a value less than 500 ms. SRX1500 devices run in dedicated mode if you've configured `set chassis dedicated-ukern-cpu`, regardless of the BFD failure detection time. You can enable distributed mode on SRX1500 devices only when dedicated mode is not enabled.

Platform-Specific Inline BFD Behavior

Platform	Difference
ACX Series	• ACX7000 Series devices do not support inline BFD for OSPFv3, IS-IS IPv6, PIM IPv6, RSVP, LDP, S-BFD, or VCCV BFD. • ACX7000 Series devices do not support multihop inline sessions. Single-hop sessions over LAG are supported. • ACX7000 Series devices support a `minimum-interval` of 4ms to 1000ms for inline BFD.
MX Series	• MX Series routers only support inline BFD if the router is static and has MPCs/MICs with `enhanced-ip` configured.

(Continued)

Platform	Difference

QFX Series	• QFX5110, QFX5120, QFX5200, and QFX5210 switches support 10 multihop inline BFD sessions. You can configure them with a timer of 150 x 3 milliseconds. Single-hop sessions are also supported.
	• Devices support either regular inline BFD or hardware-assisted inline BFD. Starting in Junos OS Release 21.2R1, QFX5120-32C and QFX5120-48Y switches support hardware-assisted inline BFD. They support a timer of 100 x 3 milliseconds. They can run up to 128 hardware-assisted inline BFD sessions, which can be a mix of single-hop and multihop BFD sessions.
	• QFX5120-48T switches support inline BFD, but do not support hardware-assisted inline BFD.
	• QFX5130-32CD, QFX5130E-32CD, QFX5130-48C, QFX5700, and QFX5700E switches—Starting in Junos OS Evolved Releases 25.2X100-D10 and 25.4R1, we support hardware-assisted inline BFD with 100 x 3 millisecond timers over VXLAN tunnels on the listed platforms only. This support applies to:
	  • Type 2 IPv4 or IPv6 L2 and L3 multihop BFD with ECMP or multihomed VTEPs with these requirements:
	  • 100 x 3 ms timers
	  • Overlay BGP peering between loopbacks
	  • BFD configured on the overlay BGP sessions
	  • Type 5 IPv4 or IPv6 multihop BFD with ECMP
	  • Pure Type 5 routing instances

Platform-Specific BFD for Static Routes Behavior

Platform	Difference
ACX Series	• ACX7000 Series devices support inline single-hop BFD for static routes with a `minimum-interval` between 4ms and 1000ms. • ACX7000 Series devices do not support inline multihop BFD for static routes.

(Continued)

Platform	Difference
EX Series	• EX4600 switches do not support minimum interval values of less than 1 second.
MX Series	• On MX Series devices, multihop BFD is not supported on a static route if the static route is configured with more than one next hop. It is recommended that you avoid using multiple next hops when a multihop BFD is required for a static route.
SRX Series	• The `bfd-liveness-detection` command includes the description field. The description is an attribute under the **bfd-liveness-detection** object. This field is applicable only for the static routes.

Platform-Specific BFD for BGP Behavior

Platform	Difference
ACX Series	• ACX7000 Series devices support inline single-hop BFD for BGP with a `minimum-interval` between 4ms and 1000ms. • ACX7000 Series devices do not support inline multihop BFD for BGP.
EX Series	• EX4600 switches do not support minimum interval values of less than 1 second.
MX Series	• On MX Series devices, multihop BFD is not supported on a static route if the static route is configured with more than one next hop. It is recommended that you avoid using multiple next hops when a multihop BFD is required for a static route.
QFX Series	• QFX5110, QFX5120, QFX5200, and QFX5210 switches support multihop Bidirectional Forwarding Detection (BFD) inline keep alive support which will enable sessions to be configured at less than 1 second. Performance may vary depending on the system load. 10 inline BFD sessions are supported and can be configured with a timer of 150 x 3 milliseconds. Single-hop sessions are also supported.

(Continued)

Platform	Difference
SRX Series	• On all SRX Series Firewalls that support this feature, high CPU utilization triggered for reasons such as CPU intensive commands and SNMP walks causes the BFD protocol to flap while processing large BGP updates. (Platform support depends on the Junos OS release in your installation.) • Starting with Junos OS Release 15.1X49-D100, SRX340, SRX345, and SRX1500 devices support dedicated BFD. • Starting with Junos OS Release 15.1X49-D100, SRX300 and SRX320 devices support real-time BFD. • Starting with Junos OS Release 15.1X49-D110, SRX550M devices support dedicated BFD.

Platform-Specific BFD for OSPF Behavior

Platform	Difference
ACX Series	• ACX7000 Series devices do not support distributed or inline BFD for OSPFv3.
EX Series	• EX4600 switches do not support minimum interval values of less than 1 second.
MX Series	• Junos OS 21.2R1 and later support distributed OSPFv3 and ISIS BFD sessions with IPv6 link local addresses on MX series routers running MPCs 1 through 9 (it is not supported on MPC 10 or MPC 11). The default for IPv6 link local BFD is inline mode.
QFX Series	• QFX5000 switches running do not support minimum interval values of less than 1 second in centralized and distributed mode. • On a single QFX5100 switches, when you add a QFX-EM-4Q expansion module, specify a minimum interval higher than 1000 ms.

(Continued)

Platform	Difference
SRX Series	• For SRX Series Firewalls that support this feature, we recommend 1000 ms as the minimum keepalive time interval for BFD packets. • For vSRX 3.0, we recommend 300 ms as the minimum keepalive time interval for BFD packets.

Platform-Specific BFD for IS-IS Behavior

Platform	Difference
ACX Series	• ACX7000 Series devices do not support distributed or inline BFD for IS-IS IPv6.
EX Series	• EX4600 switches do not support minimum interval values of less than 1 second.

Platform-Specific BFD for RIP Behavior

Platform	Difference
EX Series	• EX4600 switches do not support minimum interval values of less than 1 second.

Platform-Specific Micro-BFD Behavior

Platform	Difference
MX Series	• Starting with Junos OS Release 14.1, specify the neighbor in a BFD session. In releases before Junos OS Release 16.1, you must configure the loopback address of the remote destination as the neighbor address. Beginning with Junos OS Release 16.1, you can also configure this feature on MX Series routers that support this feature with aggregated Ethernet interface address of the remote destination as the neighbor address.

(Continued)

Platform	Difference
PTX Series	• Starting in Junos OS 21.4R1, LACP minimum link with sync reset and microBFD configuration is supported on PTX10001-36MR, PTX10003, PTX10004, PTX10008, and PTX10016 routers.

Change History Table

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

Release	Description
24.3R1	Starting in Junos OS Release 24.3R1, we've introduced distributed mode for Bidirectional Forwarding Detection (BFD) on vSRX 3.0.
24.2R1	Starting in Junos OS Release 24.2R1, PRPD BFD APIs support echo-lite mode for multihop sessions in distributed and inline BFD modes. For more information about PRPD APIs, see Overview of JET APIs.
change-completed	
15.1X49-D100	Starting with Junos OS Release 15.1X49-D100, SRX340, SRX345, and SRX1500 devices support dedicated BFD.
15.1X49-D100	Starting with Junos OS Release 15.1X49-D100, SRX300 and SRX320 devices support real-time BFD.
8.3	In Junos OS Release 8.3 and later, BFD is supported on internal BGP (IBGP) and multihop external BGP (EBGP) sessions as well as on single-hop EBGP sessions.
9.1	In Junos OS Release 9.1 through Junos OS Release 11.1, BFD supports IPv6 interfaces in static routes only.
11.2	In Junos OS Release 11.2 and later, BFD supports IPv6 interfaces with BGP.
15.1X49	Note that the functionality of configuring BFD for RIP described in this topic is not supported in Junos OS Releases 15.1X49, 15.1X49-D30, or 15.1X49-D40.

19.3	Starting with Junos OS Release 19.3 and later, for MPC10E and MPC11E MPCs, you cannot apply firewall filters on the MicroBFD packets received on the aggregated Ethernet Interface. For MPC1E through MPC9E, you can apply firewall filters on the MicroBFD packets received on the aggregated Ethernet Interface only if the aggregated Ethernet Interface is configured as an untagged Interface.

Configuring BFD

SUMMARY

Use the following examples to configure Bidirectional Forwarding Detection (BFD) on your device.

IN THIS SECTION

Example: Configuring BFD for Static Routes for Faster Network Failure Detection

IN THIS SECTION

This example shows how to configure Bidirectional Forwarding Detection (BFD) for static routes.

Requirements

In this example, no special configuration beyond device initialization is required.

Overview

IN THIS SECTION

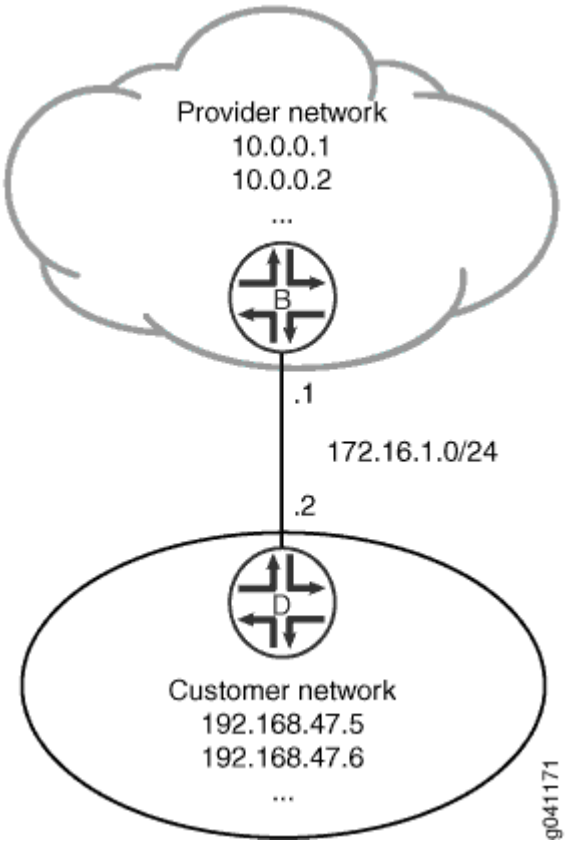
There are many practical applications for static routes. Static routing is often used at the network edge to support attachment to stub networks, which, given their single point of entry and egress, are well suited to the simplicity of a static route. In Junos OS, static routes have a global preference of 5. Static routes are activated if the specified next hop is reachable.

In this example, you configure the static route 192.168.47.0/24 from the provider network to the customer network, using the next-hop address of 172.16.1.2. You also configure a static default route of 0.0.0.0/0 from the customer network to the provider network, using a next-hop address of 172.16.1.1.

For demonstration purposes, some loopback interfaces are configured on Device B and Device D. These loopback interfaces provide addresses to ping and thus verify that the static routes are working.

Figure 1 on page 60 shows the sample network.

Figure 1: Customer Routes Connected to a Service Provider



Topology

Configuration

IN THIS SECTION

CLI Quick Configuration

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

Device B

```
set interfaces ge-1/2/0 unit 0 description B->D
set interfaces ge-1/2/0 unit 0 family inet address 172.16.1.1/24
set interfaces lo0 unit 57 family inet address 10.0.0.1/32
set interfaces lo0 unit 57 family inet address 10.0.0.2/32
set routing-options static route 192.168.47.0/24 next-hop 172.16.1.2
set routing-options static route 192.168.47.0/24 bfd-liveness-detection minimum-interval 1000
set routing-options static route 192.168.47.0/24 bfd-liveness-detection description Site-xxx
set protocols bfd traceoptions file bfd-trace
set protocols bfd traceoptions flag all
```

Device D

```
set interfaces ge-1/2/0 unit 1 description D->B
set interfaces ge-1/2/0 unit 1 family inet address 172.16.1.2/24
set interfaces lo0 unit 2 family inet address 192.168.47.5/32
set interfaces lo0 unit 2 family inet address 192.168.47.6/32
set routing-options static route 0.0.0.0/0 next-hop 172.16.1.1
set routing-options static route 0.0.0.0/0 bfd-liveness-detection minimum-interval 1000
set protocols bfd traceoptions file bfd-trace
set protocols bfd traceoptions flag all
```

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure BFD for static routes:

1. On Device B, configure the interfaces.

```
[edit interfaces]
user@B# set ge-1/2/0 unit 0 description B->D
user@B# set ge-1/2/0 unit 0 family inet address 172.16.1.1/24
user@B# set lo0 unit 57 family inet address 10.0.0.1/32
user@B# set lo0 unit 57 family inet address 10.0.0.2/32
```

2. On Device B, create a static route and set the next-hop address.

```
[edit routing-options]
user@B# set static route 192.168.47.0/24 next-hop 172.16.1.2
```

3. On Device B, configure BFD for the static route.

```
[edit routing-options]
user@B# set static route 192.168.47.0/24 bfd-liveness-detection minimum-interval 1000
set routing-options static route 192.168.47.0/24 bfd-liveness-detection description Site-
xxx
```

4. On Device B, configure tracing operations for BFD.

```
[edit protocols]
user@B# set bfd traceoptions file bfd-trace
user@B# set bfd traceoptions flag all
```

5. If you are done configuring Device B, commit the configuration.

```
[edit]
user@B# commit
```

6. On Device D, configure the interfaces.

```
[edit interfaces]
user@D# set ge-1/2/0 unit 1 description D->B
user@D# set ge-1/2/0 unit 1 family inet address 172.16.1.2/24
```

```
user@D# set lo0 unit 2 family inet address 192.168.47.5/32
user@D# set lo0 unit 2 family inet address 192.168.47.6/32
```

7. On Device D, create a static route and set the next-hop address.

```
[edit routing-options]
user@D# set static route 0.0.0.0/0 next-hop 172.16.1.1
```

8. On Device D, configure BFD for the static route.

```
[edit routing-options]
user@D# set static route 0.0.0.0/0 bfd-liveness-detection minimum-interval 1000
```

9. On Device D, configure tracing operations for BFD.

```
[edit protocols]
user@D# set bfd traceoptions file bfd-trace
user@D# set bfd traceoptions flag all
```

10. If you are done configuring Device D, commit the configuration.

```
[edit]
user@D# commit
```

Results

Confirm your configuration by issuing the `show interfaces`, `show protocols`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Device B

```
user@B# show interfaces
ge-1/2/0 {
    unit 0 {
        description B->D;
        family inet {
            address 172.16.1.1/24;
```

```
        }
    }
}
lo0 {
    unit 57 {
        family inet {
            address 10.0.0.1/32;
            address 10.0.0.2/32;
        }
    }
}
```

```
user@D# show protocols
bfd {
    traceoptions {
        file bfd-trace;
        flag all;
    }
}
```

```
user@B# show routing-options
static {
    route 192.168.47.0/24 {
        next-hop 172.16.1.2;
        bfd-liveness-detection {
            description Site- xxx;
            minimum-interval 1000;
        }
    }
}
```

Device D

```
user@D# show interfaces
ge-1/2/0 {
    unit 1 {
        description D->B;
        family inet {
            address 172.16.1.2/24;
```

```
        }
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.168.47.5/32;
            address 192.168.47.6/32;
        }
    }
}
```

```
user@D# show routing-options
static {
    route 0.0.0.0/0 {
        next-hop 172.16.1.1;
        bfd-liveness-detection {
            description Site - xxx;
            minimum-interval 1000;
        }
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That BFD Sessions Are Up

Purpose

Verify that the BFD sessions are up, and view details about the BFD sessions.

Action

From operational mode, enter the `show bfd session extensive` command.

```
user@B> show bfd session extensive
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
172.16.1.2              Up        lt-1/2/0.0    3.000     1.000        3
 Client Static, description Site-xxx, TX interval 1.000, RX interval 1.000
 Session up time 00:14:30
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated, routing table index 172
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 2, remote discriminator 1
 Echo mode disabled/inactive

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps
```

NOTE: The **description Site- <*xxx*>** is supported only on the SRX Series Firewalls.
If each client has more than one description field, then it displays "and more" along with the first description field.

```
user@D> show bfd session extensive
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
172.16.1.1              Up        lt-1/2/0.1    3.000     1.000        3
 Client Static, TX interval 1.000, RX interval 1.000
 Session up time 00:14:35
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Replicated, routing table index 170
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
```

```
 Local discriminator 1, remote discriminator 2
 Echo mode disabled/inactive

1 sessions, 1 clients
Cumulative transmit rate 1.0 pps, cumulative receive rate 1.0 pps
```

Meaning

The `TX interval 1.000, RX interval 1.000` output represents the setting configured with the `minimum-interval` statement. All of the other output represents the default settings for BFD. To modify the default settings, include the optional statements under the `bfd-liveness-detection` statement.

Viewing Detailed BFD Events

Purpose

View the contents of the BFD trace file to assist in troubleshooting, if needed.

Action

From operational mode, enter the `file show /var/log/bfd-trace` command.

```
user@B> file show /var/log/bfd-trace
Nov 23 14:26:55    Data (9) len 35: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20 78 6d 69 74 20
72
Nov 23 14:26:55 PPM Trace: BFD periodic xmit rt tbl index 172
Nov 23 14:26:55 Received Downstream TraceMsg (22) len 108:
Nov 23 14:26:55    IfIndex (3) len 4: 0
Nov 23 14:26:55    Protocol (1) len 1: BFD
Nov 23 14:26:55    Data (9) len 83: (hex) 70 70 6d 64 5f 62 66 64 5f 73 65 6e 64 6d 73 67 20 3a
20
Nov 23 14:26:55 PPM Trace: ppmd_bfd_sendmsg : socket 12 len 24, ifl 78 src 172.16.1.1 dst
172.16.1.2 errno 65
Nov 23 14:26:55 Received Downstream TraceMsg (22) len 93:
Nov 23 14:26:55    IfIndex (3) len 4: 0
Nov 23 14:26:55    Protocol (1) len 1: BFD
Nov 23 14:26:55    Data (9) len 68: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20 78 6d 69 74 20
74
```

Meaning

BFD messages are being written to the trace file.

Example: Configuring BFD on Internal BGP Peer Sessions

IN THIS SECTION

This example shows how to configure internal BGP (IBGP) peer sessions with the Bidirectional Forwarding Detection (BFD) protocol to detect failures in a network.

Requirements

No special configuration beyond device initialization is required before you configure this example.

Overview

The minimum configuration to enable BFD on IBGP sessions is to include the `bfd-liveness-detection minimum-interval` statement in the BGP configuration of all neighbors participating in the BFD session. The `minimum-interval` statement specifies the minimum transmit and receive intervals for failure detection. Specifically, this value represents the minimum interval after which the local routing device transmits hello packets as well as the minimum interval that the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value from 1 through 255,000 milliseconds.

Optionally, you can specify the minimum transmit and receive intervals separately using the `transmit-interval minimum-interval` and `minimum-receive-interval` statements. For information about these and other optional BFD configuration statements, see `bfd-liveness-detection`.

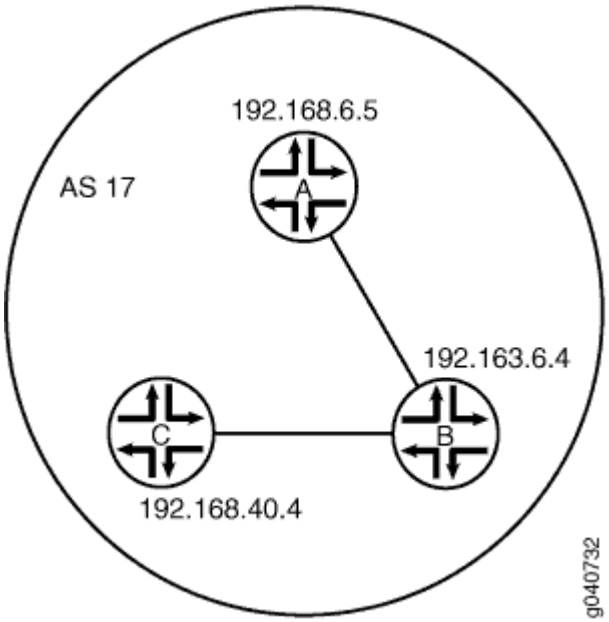
NOTE: Depending on your network environment, these additional recommendations might apply:

- To prevent BFD flapping during the general Routing Engine switchover event, specify a minimum interval of 5000 milliseconds for Routing Engine-based sessions. This minimum value is required because, during the general Routing Engine switchover event, processes such as RPD, MIBD, and SNMPD utilize CPU resources for more than the specified threshold value. Hence, BFD processing and scheduling is affected because of this lack of CPU resources.
- For BFD sessions to remain up during the dual chassis cluster control link scenario, when the first control link fails, specify the minimum interval of 6000 milliseconds to prevent the LACP from flapping on the secondary node for Routing Engine-based sessions.
- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 milliseconds for Routing Engine-based sessions and 100 milliseconds for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 milliseconds for Routing Engine-based sessions. For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

BFD is supported on the default routing instance (the main router), routing instances, and logical systems. This example shows BFD on logical systems.

Figure 2 on page 70 shows a typical network with internal peer sessions.

Figure 2: Typical Network with IBGP Sessions



Configuration

IN THIS SECTION

CLI Quick Configuration

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

Device A

```
set logical-systems A interfaces lt-1/2/0 unit 1 description to-B
set logical-systems A interfaces lt-1/2/0 unit 1 encapsulation ethernet
set logical-systems A interfaces lt-1/2/0 unit 1 peer-unit 2
set logical-systems A interfaces lt-1/2/0 unit 1 family inet address 10.10.10.1/30
set logical-systems A interfaces lo0 unit 1 family inet address 192.168.6.5/32
```

```
set logical-systems A protocols bgp group internal-peers type internal
set logical-systems A protocols bgp group internal-peers traceoptions file bgp-bfd
set logical-systems A protocols bgp group internal-peers traceoptions flag bfd detail
set logical-systems A protocols bgp group internal-peers local-address 192.168.6.5
set logical-systems A protocols bgp group internal-peers export send-direct
set logical-systems A protocols bgp group internal-peers bfd-liveness-detection minimum-interval
1000
set logical-systems A protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems A protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems A protocols ospf area 0.0.0.0 interface lo0.1 passive
set logical-systems A protocols ospf area 0.0.0.0 interface lt-1/2/0.1
set logical-systems A policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems A policy-options policy-statement send-direct term 2 then accept
set logical-systems A routing-options router-id 192.168.6.5
set logical-systems A routing-options autonomous-system 17
```

Device B

```
set logical-systems B interfaces lt-1/2/0 unit 2 description to-A
set logical-systems B interfaces lt-1/2/0 unit 2 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 2 peer-unit 1
set logical-systems B interfaces lt-1/2/0 unit 2 family inet address 10.10.10.2/30
set logical-systems B interfaces lt-1/2/0 unit 5 description to-C
set logical-systems B interfaces lt-1/2/0 unit 5 encapsulation ethernet
set logical-systems B interfaces lt-1/2/0 unit 5 peer-unit 6
set logical-systems B interfaces lt-1/2/0 unit 5 family inet address 10.10.10.5/30
set logical-systems B interfaces lo0 unit 2 family inet address 192.163.6.4/32
set logical-systems B protocols bgp group internal-peers type internal
set logical-systems B protocols bgp group internal-peers local-address 192.163.6.4
set logical-systems B protocols bgp group internal-peers export send-direct
set logical-systems B protocols bgp group internal-peers bfd-liveness-detection minimum-interval
1000
set logical-systems B protocols bgp group internal-peers neighbor 192.168.40.4
set logical-systems B protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems B protocols ospf area 0.0.0.0 interface lo0.2 passive
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.2
set logical-systems B protocols ospf area 0.0.0.0 interface lt-1/2/0.5
set logical-systems B policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems B policy-options policy-statement send-direct term 2 then accept
set logical-systems B routing-options router-id 192.163.6.4
set logical-systems B routing-options autonomous-system 17
```

Device C

```
set logical-systems C interfaces lt-1/2/0 unit 6 description to-B
set logical-systems C interfaces lt-1/2/0 unit 6 encapsulation ethernet
set logical-systems C interfaces lt-1/2/0 unit 6 peer-unit 5
set logical-systems C interfaces lt-1/2/0 unit 6 family inet address 10.10.10.6/30
set logical-systems C interfaces lo0 unit 3 family inet address 192.168.40.4/32
set logical-systems C protocols bgp group internal-peers type internal
set logical-systems C protocols bgp group internal-peers local-address 192.168.40.4
set logical-systems C protocols bgp group internal-peers export send-direct
set logical-systems C protocols bgp group internal-peers bfd-liveness-detection minimum-interval
1000
set logical-systems C protocols bgp group internal-peers neighbor 192.163.6.4
set logical-systems C protocols bgp group internal-peers neighbor 192.168.6.5
set logical-systems C protocols ospf area 0.0.0.0 interface lo0.3 passive
set logical-systems C protocols ospf area 0.0.0.0 interface lt-1/2/0.6
set logical-systems C policy-options policy-statement send-direct term 2 from protocol direct
set logical-systems C policy-options policy-statement send-direct term 2 then accept
set logical-systems C routing-options router-id 192.168.40.4
set logical-systems C routing-options autonomous-system 17
```

Configuring Device A

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure Device A:

1. Set the CLI to Logical System A.

```
user@host> set cli logical-system A
```

2. Configure the interfaces.

```
[edit interfaces lt-1/2/0 unit 1]
user@host:A# set description to-B
user@host:A# set encapsulation ethernet
```

```
user@host:A# set peer-unit 2
user@host:A# set family inet address 10.10.10.1/30
[edit interfaces lo0 unit 1]
user@host:A# set family inet address 192.168.6.5/32
```

3. Configure BGP.

   The `neighbor` statements are included for both Device B and Device C, even though Device A is not directly connected to Device C.

```
[edit protocols bgp group internal-peers]
user@host:A# set type internal
user@host:A# set local-address 192.168.6.5
user@host:A# set export send-direct
user@host:A# set neighbor 192.163.6.4
user@host:A# set neighbor 192.168.40.4
```

4. Configure BFD.

```
[edit protocols bgp group internal-peers]
user@host:A# set bfd-liveness-detection minimum-interval 1000
```

   You must configure the same minimum interval on the connecting peer.

5. (Optional) Configure BFD tracing.

```
[edit protocols bgp group internal-peers]
user@host:A# set traceoptions file bgp-bfd
user@host:A# set traceoptions flag bfd detail
```

6. Configure OSPF.

```
[edit protocols ospf area 0.0.0.0]
user@host:A# set interface lo0.1 passive
user@host:A# set interface lt-1/2/0.1
```

7. Configure a policy that accepts direct routes.

Other useful options for this scenario might be to accept routes learned through OSPF or local routes.

```
[edit policy-options policy-statement send-direct term 2]
user@host:A# set from protocol direct
user@host:A# set then accept
```

8. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@host:A# set router-id 192.168.6.5
user@host:A# set autonomous-system 17
```

9. If you are done configuring the device, enter `commit` from configuration mode.
   Repeat these steps to configure Device B and Device C.

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show policy-options`, `show protocols`, and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host:A# show interfaces
lt-1/2/0 {
    unit 1 {
        description to-B;
        encapsulation ethernet;
        peer-unit 2;
        family inet {
            address 10.10.10.1/30;
        }
    }
}
lo0 {
    unit 1 {
        family inet {
            address 192.168.6.5/32;
        }
```

```
    }
}
```

```
user@host:A# show policy-options
policy-statement send-direct {
    term 2 {
        from protocol direct;
        then accept;
    }
}
```

```
user@host:A# show protocols
bgp {
    group internal-peers {
        type internal;
        traceoptions {
            file bgp-bfd;
            flag bfd detail;
        }
        local-address 192.168.6.5;
        export send-direct;
        bfd-liveness-detection {
            minimum-interval 1000;
        }
        neighbor 192.163.6.4;
        neighbor 192.168.40.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.1 {
            passive;
        }
        interface lt-1/2/0.1;
```

```
    }
}
```

```
user@host:A# show routing-options
router-id 192.168.6.5;
autonomous-system 17;
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That BFD Is Enabled

Purpose

Verify that BFD is enabled between the IBGP peers.

Action

From operational mode, enter the `show bgp neighbor` command. You can use the `| match bfd` filter to narrow the output.

```
user@host:A> show bgp neighbor | match bfd
Options: <BfdEnabled>
  BFD: enabled, up
  Trace file: /var/log/A/bgp-bfd size 131072 files 10
  Options: <BfdEnabled>
```

```
  BFD: enabled, up
  Trace file: /var/log/A/bgp-bfd size 131072 files 10
```

Meaning

The output shows that Logical System A has two neighbors with BFD enabled. When BFD is not enabled, the output displays `BFD: disabled, down`, and the `<BfdEnabled>` option is absent. If BFD is enabled and the session is down, the output displays `BFD: enabled, down`. The output also shows that BFD-related events are being written to a log file because trace operations are configured.

Verifying That BFD Sessions Are Up

Purpose

Verify that the BFD sessions are up, and view details about the BFD sessions.

Action

From operational mode, enter the `show bfd session extensive` command.

```
user@host:A> show bfd session extensive
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
192.163.6.4             Up                      3.000     1.000        3
 Client BGP, TX interval 1.000, RX interval 1.000
 Session up time 00:54:40
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 12, routing table index 25
 Min async interval 1.000, min slow interval 1.000
 Adaptive async TX interval 1.000, RX interval 1.000
 Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
 Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
 Local discriminator 10, remote discriminator 9
 Echo mode disabled/inactive
 Multi-hop route table 25, local-address 192.168.6.5

                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
192.168.40.4            Up                      3.000     1.000        3
 Client BGP, TX interval 1.000, RX interval 1.000
```

```
  Session up time 00:48:03
  Local diagnostic None, remote diagnostic None
  Remote state Up, version 1
  Logical system 12, routing table index 25
  Min async interval 1.000, min slow interval 1.000
  Adaptive async TX interval 1.000, RX interval 1.000
  Local min TX interval 1.000, minimum RX interval 1.000, multiplier 3
  Remote min TX interval 1.000, min RX interval 1.000, multiplier 3
  Local discriminator 14, remote discriminator 13
  Echo mode disabled/inactive
  Multi-hop route table 25, local-address 192.168.6.5

2 sessions, 2 clients
Cumulative transmit rate 2.0 pps, cumulative receive rate 2.0 pps
```

Meaning

The `TX interval 1.000, RX interval 1.000` output represents the setting configured with the `minimum-interval` statement. All of the other output represents the default settings for BFD. To modify the default settings, include the optional statements under the `bfd-liveness-detection` statement.

Viewing Detailed BFD Events

Purpose

View the contents of the BFD trace file to assist in troubleshooting, if needed.

Action

From operational mode, enter the `file show /var/log/A/bgp-bfd` command.

```
user@host:A> file show /var/log/A/bgp-bfd
Aug 15 17:07:25 trace_on: Tracing to "/var/log/A/bgp-bfd" started
Aug 15 17:07:26.492190 bgp_peer_init: BGP peer 192.163.6.4 (Internal AS 17) local address
192.168.6.5 not found.  Leaving peer idled
Aug 15 17:07:26.493176 bgp_peer_init: BGP peer 192.168.40.4 (Internal AS 17) local address
192.168.6.5 not found.  Leaving peer idled
Aug 15 17:07:32.597979 task_connect: task BGP_17.192.163.6.4+179 addr 192.163.6.4+179: No route
to host
Aug 15 17:07:32.599623 bgp_connect_start: connect 192.163.6.4 (Internal AS 17): No route to host
Aug 15 17:07:36.869394 task_connect: task BGP_17.192.168.40.4+179 addr 192.168.40.4+179: No
```

```
route to host
Aug 15 17:07:36.870624 bgp_connect_start: connect 192.168.40.4 (Internal AS 17): No route to host
Aug 15 17:08:04.599220 task_connect: task BGP_17.192.163.6.4+179 addr 192.163.6.4+179: No route
to host
Aug 15 17:08:04.601135 bgp_connect_start: connect 192.163.6.4 (Internal AS 17): No route to host
Aug 15 17:08:08.869717 task_connect: task BGP_17.192.168.40.4+179 addr 192.168.40.4+179: No
route to host
Aug 15 17:08:08.869934 bgp_connect_start: connect 192.168.40.4 (Internal AS 17): No route to host
Aug 15 17:08:36.603544 advertising receiving-speaker only capabilty to neighbor 192.163.6.4
(Internal AS 17)
Aug 15 17:08:36.606726 bgp_read_message: 192.163.6.4 (Internal AS 17): 0 bytes buffered
Aug 15 17:08:36.609119 Initiated BFD session to peer 192.163.6.4 (Internal AS 17):
address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3 ver=255
Aug 15 17:08:36.734033 advertising receiving-speaker only capabilty to neighbor 192.168.40.4
(Internal AS 17)
Aug 15 17:08:36.738436 Initiated BFD session to peer 192.168.40.4 (Internal AS 17):
address=192.168.40.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3 ver=255
Aug 15 17:08:40.537552 BFD session to peer 192.163.6.4 (Internal AS 17) up
Aug 15 17:08:40.694410 BFD session to peer 192.168.40.4 (Internal AS 17) up
```

Meaning

Before the routes are established, the `No route to host` message appears in the output. After the routes are established, the last two lines show that both BFD sessions come up.

Viewing Detailed BFD Events After Deactivating and Reactivating a Loopback Interface

Purpose

Check to see what happens after bringing down a router or switch and then bringing it back up. To simulate bringing down a router or switch, deactivate the loopback interface on Logical System B.

Action

1. From configuration mode, enter the `deactivate logical-systems B interfaces lo0 unit 2 family inet` command.

```
user@host:A# deactivate logical-systems B interfaces lo0 unit 2 family inet
user@host:A# commit
```

2. From operational mode, enter the `file show /var/log/A/bgp-bfd` command.

```
user@host:A> file show /var/log/A/bgp-bfd
...
Aug 15 17:20:55.995648 bgp_read_v4_message:9747: NOTIFICATION received from 192.163.6.4
(Internal AS 17): code 6 (Cease) subcode 6 (Other Configuration Change)
Aug 15 17:20:56.004508 Terminated BFD session to peer 192.163.6.4 (Internal AS 17)
Aug 15 17:21:28.007755 task_connect: task BGP_17.192.163.6.4+179 addr 192.163.6.4+179: No
route to host
Aug 15 17:21:28.008597 bgp_connect_start: connect 192.163.6.4 (Internal AS 17): No route to
host
```

3. From configuration mode, enter the `activate logical-systems B interfaces lo0 unit 2 family inet` command.

```
user@host:A# activate logical-systems B interfaces lo0 unit 2 family inet
user@host:A# commit
```

4. From operational mode, enter the `file show /var/log/A/bgp-bfd` command.

```
user@host:A> file show /var/log/A/bgp-bfd
...
Aug 15 17:25:53.623743 advertising receiving-speaker only capabilty to neighbor 192.163.6.4
(Internal AS 17)
Aug 15 17:25:53.631314 Initiated BFD session to peer 192.163.6.4 (Internal AS 17):
address=192.163.6.4 ifindex=0 ifname=(none) txivl=1000 rxivl=1000 mult=3 ver=255
Aug 15 17:25:57.570932 BFD session to peer 192.163.6.4 (Internal AS 17) up
```

Example: Configuring BFD for OSPF

IN THIS SECTION

This example shows how to configure the Bidirectional Forwarding Detection (BFD) protocol for OSPF.

Requirements

Before you begin:

- Configure the device interfaces. See the Junos OS Network Interfaces Library for Routing Devices.
- Configure the router identifiers for the devices in your OSPF network. See *Example: Configuring an OSPF Router Identifier*.
- Control OSPF designated router election. See *Example: Controlling OSPF Designated Router Election*.
- Configure a single-area OSPF network. See *Example: Configuring a Single-Area OSPF Network*.
- Configure a multiarea OSPF network. See *Example: Configuring a Multiarea OSPF Network*.
- Configure a multiarea OSPF network. See *Example: Configuring a Multiarea OSPF Network*.

Overview

IN THIS SECTION

An alternative to adjusting the OSPF hello interval and dead interval settings to increase route convergence is to configure BFD. The BFD protocol is a simple hello mechanism that detects failures in a network. The BFD failure detection timers have shorter timer limits than the OSPF failure detection mechanisms, thereby providing faster detection.

BFD is useful on interfaces that are unable to detect failure quickly, such as Ethernet interfaces. Other interfaces, such as SONET interfaces, already have built-in failure detection. Configuring BFD on those interfaces is unnecessary.

You configure BFD on a pair of neighboring OSPF interfaces. Unlike the OSPF hello interval and dead interval settings, you do not have to enable BFD on all interfaces in an OSPF area.

In this example, you enable failure detection by including the `bfd-liveness-detection` statement on the neighbor OSPF interface **fe-0/1/0** in area 0.0.0.0 and configure the BFD packet exchange interval to 300 milliseconds, configure 4 as the number of missed hello packets that causes the originating interface to be declared down, and configure BFD sessions only for OSPF neighbors with full neighbor adjacency by including the following settings:

- **full-neighbors-only**—In Junos OS Release 9.5 and later, configures the BFD protocol to establish BFD sessions only for OSPF neighbors with full neighbor adjacency. The default behavior is to establish BFD sessions for all OSPF neighbors.

- **minimum-interval**—Configures the minimum interval, in milliseconds, after which the local routing device transmits hello packets as well as the minimum interval after which the routing device expects to receive a reply from the neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds. You can also specify the minimum transmit and receive intervals separately using the **transmit-interval minimum-interval** and `minimum-receive-interval` statements.

  NOTE: Depending on your network environment, these additional recommendations might apply:

  - For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of no less than 500 ms. An interval of 1000 ms is recommended to avoid any instability issues.

    NOTE:

    - For the bfdd process, the detection time interval set is lower than 300 ms. If there is a high priority process such as ppmd running on the system, the CPU might spend time on the ppmd process rather than the bfdd process.

    - For branch SRX Series Firewalls, we recommend 1000 ms as the minimum keepalive time interval for BFD packets.

    - For vSRX 3.0, we recommend 300 ms as the minimum keepalive time interval for BFD packets.

  - For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.

- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with NSR configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

- **multiplier**—Configures the number of hello packets not received by a neighbor that causes the originating interface to be declared down. By default, three missed hello packets cause the originating interface to be declared down. You can configure a value in the range from 1 through 255.

Topology

Configuration

IN THIS SECTION

Procedure

CLI Quick Configuration

To quickly configure the BFD protocol for OSPF, 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.

```
[edit]
set protocols ospf area 0.0.0.0 interface fe-0/0/1 bfd-liveness-detection minimum-interval 300
set protocols ospf area 0.0.0.0 interface fe-0/0/1 bfd-liveness-detection multiplier 4
set protocols ospf area 0.0.0.0 interface fe-0/0/1 bfd-liveness-detection full-neighbors-only
```

Step-by-Step Procedure

To configure the BFD protocol for OSPF on one neighboring interface:

1. Create an OSPF area.

NOTE: To specify OSPFv3, include the `ospf3` statement at the `[edit protocols]` hierarchy level.

```
[edit]
user@host# edit protocols ospf area 0.0.0.0
```

2. Specify the interface.

```
[edit protocols ospf area 0.0.0.0]
user@host# set interface fe-0/0/1
```

3. Specify the minimum transmit and receive intervals.

```
[edit protocols ospf area 0.0.0.0 ]
user@host# set interface fe-0/0/1 bfd-liveness-detection minimum-interval 300
```

4. Configure the number of missed hello packets that cause the originating interface to be declared down.

```
[edit protocols ospf area 0.0.0.0 ]
user@host# set interface fe-0/0/1 bfd-liveness-detection multiplier 4
```

5. Configure BFD sessions only for OSPF neighbors with full neighbor adjacency.

```
[edit protocols ospf area 0.0.0.0 ]
user@host# set interface fe-0/0/1 bfd-liveness-detection full-neighbors-only
```

6. If you are done configuring the device, commit the configuration.

```
[edit protocols ospf area 0.0.0.0 ]
user@host# commit
```

NOTE: Repeat this entire configuration on the other neighboring interface.

Results

Confirm your configuration by entering the `show protocols ospf` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols ospf
area 0.0.0.0 {
    interface fe-0/0/1.0 {
        bfd-liveness-detection {
            minimum-interval 300;
            multiplier 4;
            full-neighbors-only;
        }
    }
}
```

To confirm your OSPFv3 configuration, enter the `show protocols ospf3` command.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the BFD Sessions

Purpose

Verify that the OSPF interfaces have active BFD sessions, and that session components have been configured correctly.

Action

From operational mode, enter the `show bfd session detail` command.

Meaning

The output displays information about the BFD sessions.

- The Address field displays the IP address of the neighbor.
- The Interface field displays the interface you configured for BFD.
- The State field displays the state of the neighbor and should show Full to reflect the full neighbor adjacency that you configured.
- The Transmit Interval field displays the time interval you configured to send BFD packets.
- The Multiplier field displays the multiplier you configured.

Example: Configuring BFD for IS-IS

IN THIS SECTION

This example describes how to configure the Bidirectional Forwarding Detection (BFD) protocol to detect failures in an IS-IS network.

NOTE: BFD is not supported with ISIS for IPV6 on QFX10000 series switches.

Requirements

Before you begin, configure IS-IS on both routers. See *Example: Configuring IS-IS* for information about the required IS-IS configuration.

NOTE: We provide the IS-IS configuration in the CLI quick configuration section but do not cover the IS-IS configuration in the step-by-step.

This example uses the following hardware and software components:

- Junos OS Release 7.3 or later
  - Updated and revalidated using Junos OS Release 22.4
- M Series, MX Series, and T Series routers

Overview

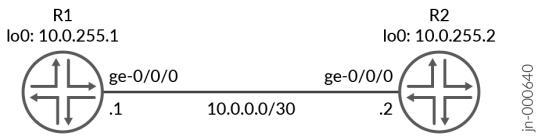
IN THIS SECTION

This example shows two routers connected to each other. A loopback interface is configured on each router. IS-IS and BFD protocols are configured on both routers.

Topology

Figure 3 on page 87 shows the sample network.

Figure 3: Configuring BFD for IS-IS



Configuration

IN THIS SECTION

CLI Quick Configuration

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

Router R1

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.1/30
set interfaces ge-0/0/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 10.0.255.1/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.0255.0001.00
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection version automatic
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection minimum-interval 200
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection minimum-receive-
interval 100
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection multiplier 2
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection no-adaptation
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection transmit-interval
minimum-interval 100
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection transmit-interval
threshold 300
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection detection-time
threshold 500
set protocols isis interface lo0.0
```

Router R2

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.2/30
set interfaces ge-0/0/0 unit 0 family iso
set interfaces lo0 unit 0 family inet address 10.0.255.2/32
set interfaces lo0 unit 0 family iso address 49.0001.0010.0255.0002.00
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection version automatic
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection minimum-interval 200
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection minimum-receive-
interval 100
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection multiplier 2
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection no-adaptation
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection transmit-interval
minimum-interval 100
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection transmit-interval
threshold 300
```

```
set protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection detection-time
threshold 500
set protocols isis interface lo0.0
```

Procedure

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

NOTE: To simply configure BFD for IS-IS, only the `minimum-interval` statement is required. The BFD protocol selects default parameters for all the other configuration statements when you use the `bfd-liveness-detection` statement without specifying any parameters.

NOTE: You can change parameters at any time without stopping or restarting the existing session. BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each BFD peer.

To configure BFD for IS-IS on Routers R1 and R2:

NOTE: We are only showing the steps for R1.

1. Configure the threshold for the adaptation of the detection time, which must be greater than the multiplier number multiplied by the minimum interval.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set detection-time threshold 500
```

2. Configure the minimum transmit and receive intervals for failure detection.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set minimum-interval 200
```

3. Configure only the minimum receive interval for failure detection.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set minimum-receive-interval 100
```

4. Disable BFD adaptation.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set no-adaptation
```

5. Configure the threshold for the transmit interval, which must be greater than the minimum transmit interval.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set transmit-interval threshold 300
```

6. Configure the minimum transmit interval for failure detection.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set transmit-interval minimum-interval 100
```

7. Configure the multiplier number, which is the number of hello packets not received by the neighbor that causes the originating interface to be declared down.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set multiplier 2
```

8. Configure the BFD version used for detection.

   The default is to have the version detected automatically.

```
[edit protocols isis interface ge-0/0/0.0 family inet bfd-liveness-detection]
user@R1# set version automatic
```

Results

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

```
user@R1# show protocols isis interface ge-0/0/0.0 family inet
bfd-liveness-detection {
    version automatic;
    minimum-interval 200;
    minimum-receive-interval 100;
    multiplier 2;
    no-adaptation;
    transmit-interval {
        minimum-interval 100;
        threshold 300;
    }
    detection-time {
        threshold 500;
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Connection Between Routers R1 and R2

Purpose

Make sure that Routers R1 and R2 can reach each other.

Action

Ping the other router to check the connectivity between the two routers as per the network topology.

```
user@R1> ping 10.0.0.2 count 2
PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=64 time=2.148 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.923 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.923/2.035/2.148/0.113 ms
```

Meaning

Routers R1 and R2 are able to ping each other.

Verifying That IS-IS Is Configured

Purpose

Make sure that the IS-IS instance is running on both routers.

Action

Use the `show isis database` statement to check if the IS-IS instance is running on both routers, R1 and R2.

```
user@R1> show isis database
IS-IS level 1 link-state database:
LSP ID                  Sequence Checksum Lifetime Attributes
R1.00-00                    0x1b   0xa2d5      552 L1 L2
R1.02-00                    0x2b   0x8da3      545 L1 L2
R2.00-00                    0x1a   0x628d      543 L1 L2
  3 LSPs

IS-IS level 2 link-state database:
LSP ID                  Sequence Checksum Lifetime Attributes
R1.00-00                    0x1e   0xb9ba      552 L1 L2
R1.02-00                    0x2b   0x8da3      545 L1 L2
```

```
R2.00-00                      0x1d   0x877e      543 L1 L2
  3 LSPs
```

Meaning

IS-IS is configured on both routers, R1 and R2.

Verifying That BFD Is configured

Purpose

Make sure that the BFD instance is running on both routers, R1 and R2.

Action

Use the `show bfd session detail` statement to check if BFD instance is running on the routers.

```
user@R1> show bfd session detail
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
10.0.0.2                Up        ge-0/0/0.0    0.200     0.100        2
 Client ISIS L1, TX interval 0.100, RX interval 0.100
 Client ISIS L2, TX interval 0.100, RX interval 0.100
 Session up time 00:02:41, previous down time 00:00:09
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Session type: Single hop BFD

1 sessions, 2 clients
Cumulative transmit rate 10.0 pps, cumulative receive rate 10.0 pps
```

Meaning

BFD is configured on Routers R1 and R2 for detecting failures in the IS-IS network.

Example: Configuring BFD for RIP

IN THIS SECTION

This example shows how to configure Bidirectional Forwarding Detection (BFD) for a RIP network.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

IN THIS SECTION

To enable failure detection, include the `bfd-liveness-detection` statement:

```
bfd-liveness-detection {
    detection-time {
        threshold milliseconds;
    }
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    multiplier number;
    no-adaptation;
    transmit-interval {
        threshold milliseconds;
        minimum-interval milliseconds;
    }
```

```
    version (1 | automatic);
}
```

Optionally, you can specify the threshold for the adaptation of the detection time by including the `threshold` statement. When the BFD session detection time adapts to a value equal to or greater than the threshold, a single trap and a system log message are sent.

To specify the minimum transmit and receive interval for failure detection, include the `minimum-interval` statement. This value represents the minimum interval at which the local routing device transmits hello packets as well as the minimum interval at which the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds. This examples sets a minimum interval of 600 milliseconds.

NOTE: Depending on your network environment, these additional recommendations might apply:

- The reccomended minimum interval for distributed BFD is 100 ms with a `multiplier` of 3.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with nonstop active routing configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

You can optionally specify the minimum transmit and receive intervals separately.

To specify only the minimum receive interval for failure detection, include the `minimum-receive-interval` statement. This value represents the minimum interval at which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,00 milliseconds.

To specify only the minimum transmit interval for failure detection, include the `transmit-interval minimum-interval` statement. This value represents the minimum interval at which the local routing device transmits hello packets to the neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds.

To specify the number of hello packets not received by a neighbor that causes the originating interface to be declared down, include the `multiplier` statement. The default is 3, and you can configure a value in the range from 1 through 255.

To specify the threshold for detecting the adaptation of the transmit interval, include the `transmit-interval threshold` statement. The threshold value must be greater than the transmit interval.

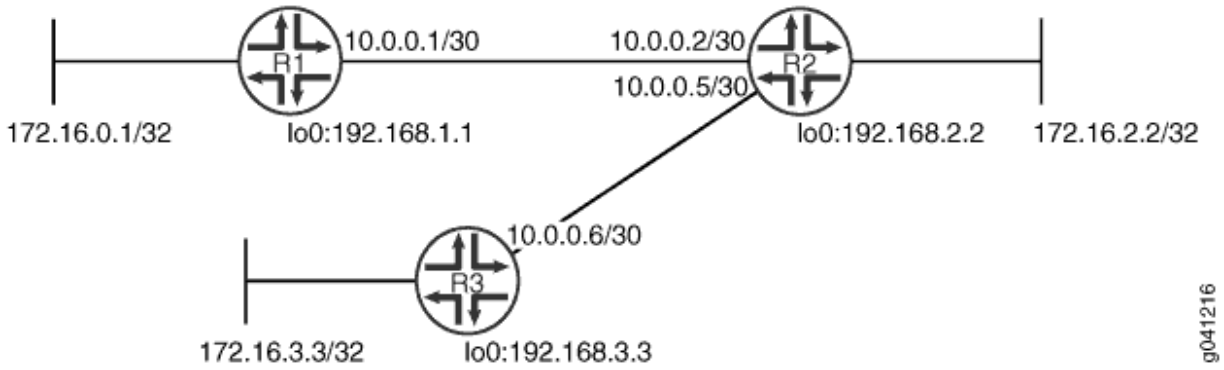
To specify the BFD version used for detection, include the `version` statement. The default is to have the version detected automatically.

You can trace BFD operations by including the `traceoptions` statement at the `[edit protocols bfd]` hierarchy level.

In Junos OS Release 9.0 and later, you can configure BFD sessions not to adapt to changing network conditions. To disable BFD adaptation, include the `no-adaptation` statement. We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

Figure 4 on page 96 shows the topology used in this example.

Figure 4: RIP BFD Network Topology



"CLI Quick Configuration" on page 97 shows the configuration for all of the devices in Figure 4 on page 96. The section "Step-by-Step Procedure" on page 98 describes the steps on Device R1.

Topology

Configuration

IN THIS SECTION

- Procedure | 97

Procedure

CLI Quick Configuration

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

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set protocols bfd traceoptions file bfd-trace
set protocols bfd traceoptions flag all
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct
```

```
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure a BFD for a RIP network:

1. Configure the network interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
```

2. Create the RIP group and add the interface.

   To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

   In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

5. Enable BFD.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection minimum-interval 600
```

6. Configure tracing operations to track BFD messages.

```
[edit protocols bfd traceoptions]
user@R1# set file bfd-trace
user@R1# set flag all
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols`, and `show policy-options` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
    unit 1 {
        family inet {
            address 10.0.0.1/30;
        }
    }
}
```

```
user@R1# show protocols
bfd {
    traceoptions {
        file bfd-trace;
        flag all;
    }
}
rip {
    group rip-group {
        export advertise-routes-through-rip;
        bfd-liveness-detection {
            minimum-interval 600;
```

```
        }
        neighbor fe-1/2/0.1;
    }
}
```

```
user@R1# show policy-options
policy-statement advertise-routes-through-rip {
    term 1 {
        from protocol [ direct rip ];
        then accept;
    }
}
```

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

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That the BFD Sessions Are Up

Purpose

Make sure that the BFD sessions are operating.

Action

From operational mode, enter the `show bfd session` command.

```
user@R1> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
```

```
10.0.0.2                Up        fe-1/2/0.1     1.800     0.600        3

1 sessions, 1 clients
Cumulative transmit rate 1.7 pps, cumulative receive rate 1.7 pps
```

Meaning

The output shows that there are no authentication failures.

Checking the BFD Trace File

Purpose

Use tracing operations to verify that BFD packets are being exchanged.

Action

From operational mode, enter the `show log` command.

```
user@R1> show log bfd-trace
Feb 16 10:26:32 PPM Trace: BFD periodic xmit to 10.0.0.2 (IFL 124, rtbl 53, single-hop port)
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 86:
Feb 16 10:26:32    IfIndex (3) len 4: 0
Feb 16 10:26:32    Protocol (1) len 1: BFD
Feb 16 10:26:32    Data (9) len 61: (hex) 42 46 44 20 70 61 63 6b 65 74 20 66 72 6f 6d 20 31 30
2e
Feb 16 10:26:32 PPM Trace: BFD packet from 10.0.0.1 (IFL 73, rtbl 56, ttl 255) absorbed
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 60:
Feb 16 10:26:32    IfIndex (3) len 4: 0
Feb 16 10:26:32    Protocol (1) len 1: BFD
Feb 16 10:26:32    Data (9) len 35: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20 78 6d 69 74 20
6f
...
```

Meaning

The output shows the normal functioning of BFD.

Configuring Micro BFD Sessions for LAG

The Bidirectional Forwarding Detection (BFD) protocol is a simple detection protocol that quickly detects failures in the forwarding paths. A link aggregation group (LAG) combines multiple links between devices that are in point-to-point connections, thereby increasing bandwidth, providing reliability, and allowing load balancing. To run a BFD session on LAG interfaces, configure an independent, asynchronous mode BFD session on every LAG member link in a LAG bundle. Instead of a single BFD session monitoring the status of the UDP port, independent micro BFD sessions monitor the status of individual member links.

NOTE: Starting in Junos OS Evolved Release 20.1R1, independent micro Bidirectional Forwarding Detection (BFD) sessions are enabled on a per member link basis of a Link Aggregation Group (LAG) bundle.

To enable failure detection for aggregated Ethernet interfaces:

1. Include the following statement in the configuration at the `[edit interfaces` *aex* `aggregated-ether-options]` hierarchy level:

```
bfd-liveness-detection
```

2. Configure the authentication criteria of the BFD session for LAG.

   To specify the authentication criteria, include the `authentication` statement:

```
bfd-liveness-detection {
    authentication {
        algorithm algorithm-name;
        key-chain key-chain-name;
        loose-check;
    }
}
```

   - Specify the algorithm to be used to authenticate the BFD session. You can use one of the following algorithms for authentication:
     - keyed-md5
     - keyed-sha-1
     - meticulous-keyed-md5
     - meticulous-keyed-sha-1

- simple-password

- To configure the key chain, specify the name that is associated with the security key for the BFD session. The name you specify must match one of the key chains configured in the `authentication-key-chains` *`key-chain`* statement at the `[edit security]` hierarchy level.

- Configure loose authentication checking on the BFD session. Use only for transitional periods when authentication might not be configured at both ends of the BFD session.

3. Configure BFD timers for aggregated Ethernet interfaces.

   To specify the BFD timers, include the `detection-time` statement:

```
bfd-liveness-detection {
    detection-time {
        threshold  milliseconds;
    }
}
```

   Specify the threshold value. This is the maximum time interval for detecting a BFD neighbor. If the transmit interval is greater than this value, the device triggers a trap.

4. Configure a hold-down interval value to set the minimum time that the BFD session must remain up before a state change notification is sent to the other members in the LAG network.

   To specify the hold-down interval, include the `holddown-interval` statement:

```
bfd-liveness-detection {
    holddown-interval milliseconds;
}
```

   You can configure a number in the range from 0 through 255,000 milliseconds, and the default is 0. If the BFD session goes down and then comes back up during the hold-down interval, the timer is restarted.

   This value represents the minimum interval at which the local routing device transmits BFD packets, as well as the minimum interval in which the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds. You can also specify the minimum transmit and receive intervals separately.

5. Configure the source address for the BFD session.

To specify a local address, include the `local-address` statement:

```
bfd-liveness-detection {
    local-address bfd-local-address;
}
```

The BFD local address is the loopback address of the source of the BFD session.

NOTE: Beginning with Junos OS Release 16.1, you can also configure this feature with the AE interface address as the local address in a micro BFD session. For the IPv6 address family, disable duplicate address detection before configuring this feature with the AE interface address. To disable duplicate address detection, include the `dad-disable` statement at the `[edit interface aex unit y family inet6]` hierarchy level.

Beginning with Release 16.1R2, Junos OS checks and validates the configured micro BFD `local-address` against the interface or loopback IP address before the configuration commit. Junos OS performs this check on both IPv4 and IPv6 micro BFD address configurations, and if they do not match, the commit fails. The configured micro-BFD `local-address` should match with the micro-BFD `neighbour-address` configured on the peer router.

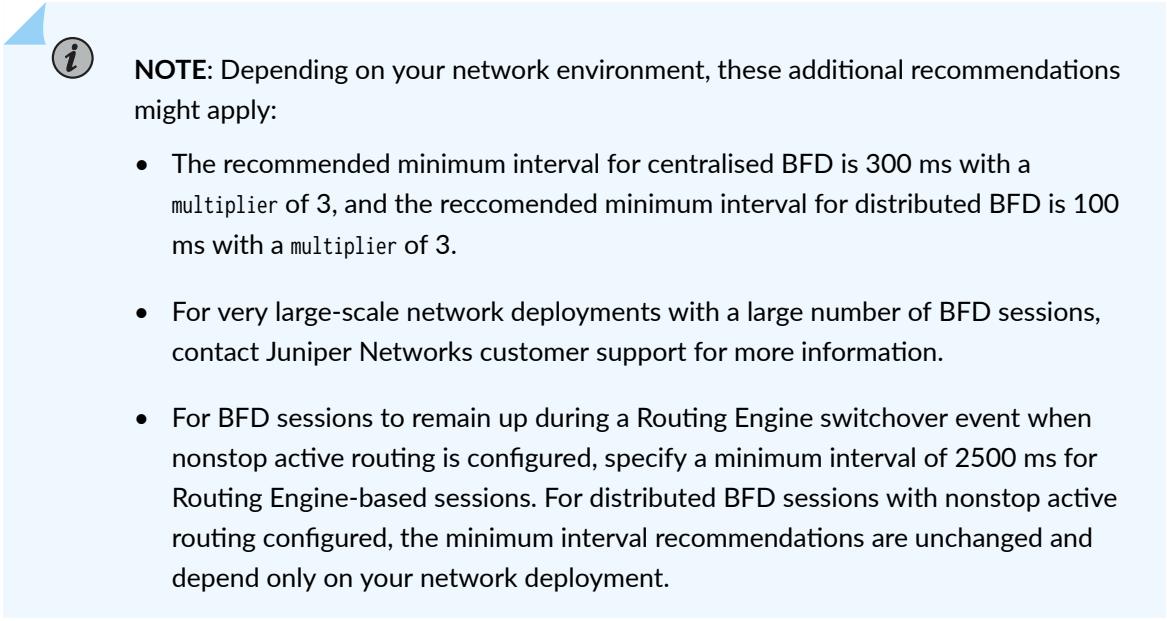
AFT-based Trio line cards (MPC10 and newer) use a different hardware design. If micro BFD is activated on an interface, the received packets won't be part of the interface group for the AE interface and won't match filter terms on lo0.0 with the interface group. To ensure terms match, you can set up a separate filter on lo0.0 using port 6784.

6. Specify the minimum interval that indicates the time interval for transmitting and receiving data.

   This value represents the minimum interval at which the local routing device transmits BFD packets, as well as the minimum interval in which the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds. You can also specify the minimum transmit and receive intervals separately.

   To specify the minimum transmit and receive intervals for failure detection, include the `minimum-interval` statement:

   ```
   bfd-liveness-detection {
       minimum-interval milliseconds;
   }
   ```



NOTE: Depending on your network environment, these additional recommendations might apply:

- The recommended minimum interval for centralised BFD is 300 ms with a `multiplier` of 3, and the reccomended minimum interval for distributed BFD is 100 ms with a `multiplier` of 3.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with nonstop active routing configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

7. Specify only the minimum receive interval for failure detection by including the `minimum-receive-interval` statement:

```
bfd-liveness-detection {
    minimum-receive-interval milliseconds;
}
```

This value represents the minimum interval in which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a number in the range from 1 through 255,000 milliseconds.

8. Specify the number of BFD packets that were not received by the neighbor that causes the originating interface to be declared down by including the `multiplier` statement:

```
bfd-liveness-detection {
    multiplier number;
}
```

The default value is 3. You can configure a number in the range from 1 through 255.

9. Configure the neighbor in a BFD session.

The neighbor address can be either an IPv4 or an IPv6 address.

To specify the next hop of the BFD session, include the `neighbor` statement:

```
bfd-liveness-detection {
    neighbor bfd-neighbor-address;
}
```

The BFD neighbor address is the loopback address of the remote destination of the BFD session.

NOTE: Beginning with Junos OS Release 16.1, you can also configure the AE interface address of the remote destination as the BFD neighbor address in a micro BFD session.

10. (Optional) Configure BFD sessions not to adapt to changing network conditions.

    To disable BFD adaptation, include the `no-adaptation` statement:

```
bfd-liveness-detection {
    no-adaptation;
}
```

NOTE: We recommend that you do not disable BFD adaptation unless it is preferable not to have BFD adaptation in your network.

11. Specify a threshold for detecting the adaptation of the detection time by including the `threshold` statement:

```
bfd-liveness-detection {
    detection-time {
        threshold milliseconds;
    }
}
```

When the BFD session detection time adapts to a value equal to or greater than the threshold, a single trap and a system log message are sent. The detection time is based on the multiplier of the minimum-interval or the minimum-receive-interval value. The threshold must be a higher value than the multiplier for either of these configured values. For example, if the minimum-receive-interval is 300 ms and the multiplier is 3, the total detection time is 900 ms. Therefore, the detection time threshold must have a value greater than 900.

12. Specify only the minimum transmit interval for failure detection by including the `transmit-interval minimum-interval` statement:

```
bfd-liveness-detection {
    transmit-interval {
        minimum-interval milliseconds;
    }
}
```

This value represents the minimum interval at which the local routing device transmits BFD packets to the neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds.

13. Specify the transmit threshold for detecting the adaptation of the transmit interval by including the `transmit-interval threshold` statement:

```
bfd-liveness-detection {
    transmit-interval {
        threshold milliseconds;
    }
}
```

The threshold value must be greater than the transmit interval. When the BFD session detection time adapts to a value greater than the threshold, a single trap and a system log message are sent. The detection time is based on the multiplier of the minimum-interval or the minimum-receive-interval value. The threshold must be a higher value than the multiplier for either of these configured values.

14. Specify the BFD version by including the `version` statement:

```
bfd-liveness-detection {
    version (1 | automatic);
}
```

The default is to have the version detected automatically.

NOTE:

- The `version` option is not supported on the QFX Series. Starting in Junos OS Release 17.2R1, a warning will appear if you attempt to use this command.

- This feature works when both the devices support BFD. If BFD is configured at only one end of the LAG, this feature does not work.

Example: Configuring Independent Micro BFD Sessions for LAG

IN THIS SECTION

This example shows how to configure an independent micro BFD session for aggregated Ethernet interfaces.

Requirements

This example uses the following hardware and software components:

- MX Series routers with Junos Trio chipset

  NOTE: AFT-based Trio line cards (MPC10 and newer) use a different hardware design. If micro BFD is activated on an interface, the received packets won't be part of the interface group for the AE interface and won't match filter terms on lo0.0 with the interface group. To ensure terms match, you can set up a separate filter on lo0.0 using port 6784.

- T Series routers with Type 4 FPC or Type 5 FPC

  BFD for LAG is supported on the following PIC types on T-Series:

  - PC-1XGE-XENPAK (Type 3 FPC),
  - PD-4XGE-XFP (Type 4 FPC),
  - PD-5-10XGE-SFPP (Type 4 FPC),

- 24x10GE (LAN/WAN) SFPP, 12x10GE (LAN/WAN) SFPP, 1X100GE Type 5 PICs

- PTX Series routers with 24X10GE (LAN/WAN) SFPP
- Junos OS Release 13.3 or later running on all devices

Overview

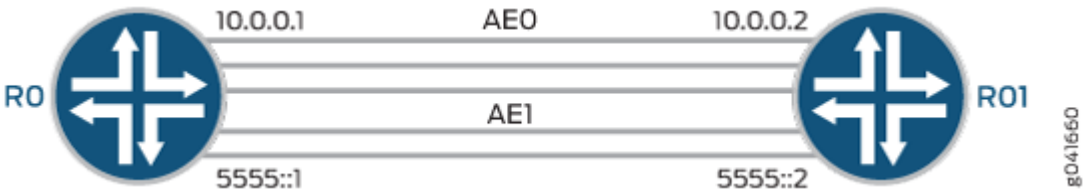
IN THIS SECTION

The example includes two routers that are directly connected. Configure two aggregated Ethernet interfaces, AE0 for IPv4 connectivity and AE1 for IPv6 connectivity. Configure micro BFD session on the AE0 bundle using IPv4 addresses as local and neighbor endpoints on both routers. Configure micro BFD session on the AE1 bundle using IPv6 addresses as local and neighbor endpoints on both routers. This example verifies that independent micro BFD sessions are active in the output.

Topology

Figure 5 on page 109 shows the sample topology.

Figure 5: Configuring an Independent Micro BFD Session for LAG



Configuration

IN THIS SECTION

CLI Quick Configuration

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

Router R0

```
set interfaces ge-1/0/1 unit 0 family inet address 20.20.20.1/30
set interfaces ge-1/0/1 unit 0 family inet6 address 3ffe::1:1/126
set interfaces xe-4/0/0 gigether-options 802.3ad ae0
set interfaces xe-4/0/1 gigether-options 802.3ad ae0
set interfaces xe-4/1/0 gigether-options 802.3ad ae1
set interfaces xe-4/1/1 gigether-options 802.3ad ae1

set interfaces lo0 unit 0 family inet address 10.255.106.107/32
set interfaces lo0 unit 0 family inet6 address 201:DB8:251::aa:aa:1/126
set interfaces ae0 aggregated-ether-options bfd-liveness-detection minimum-interval 100
set interfaces ae0 aggregated-ether-options bfd-liveness-detection neighbor 10.255.106.102
set interfaces ae0 aggregated-ether-options bfd-liveness-detection local-address 10.255.106.107
set interfaces ae0 aggregated-ether-options minimum-links 1
set interfaces ae0 aggregated-ether-options link-speed 10g
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 unit 0 family inet address 10.0.0.1/30
set interfaces ae1 aggregated-ether-options bfd-liveness-detection minimum-interval 100
set interfaces ae1 aggregated-ether-options bfd-liveness-detection multiplier 3
set interfaces ae1 aggregated-ether-options bfd-liveness-detection neighbor 201:DB8:251::bb:bb:1
set interfaces ae1 aggregated-ether-options bfd-liveness-detection local-address
201:DB8:251::aa:aa:1
set interfaces ae1 aggregated-ether-options minimum-links 1
set interfaces ae1 aggregated-ether-options link-speed 10g
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 unit 0 family inet6 address 5555::1/126
set interface ae1 unit 0 family inet6 dad-disable
set routing-options nonstop-routing
set routing-options static route 30.30.30.0/30 next-hop 10.0.0.2
```

```
set routing-options rib inet6.0 static route 3ffe::1:2/126 next-hop 5555::2
set protocols bfd traceoptions file bfd
set protocols bfd traceoptions file size 100m
set protocols bfd traceoptions file files 10
set protocols bfd traceoptions flag all
```

Router R1

```
set interfaces ge-1/1/8 unit 0 family inet address 30.30.30.1/30set interfaces ge-1/1/8 unit 0
family inet6 address 3ffe::1:2/126
set interfaces xe-0/0/0 gigether-options 802.3ad ae0set interfaces xe-0/0/1 gigether-options
802.3ad ae0set interfaces xe-0/0/2 gigether-options 802.3ad ae1set interfaces xe-0/0/3 gigether-
options 802.3ad ae1
set interfaces lo0 unit 0 family inet address 10.255.106.102/32
set interfaces lo0 unit 0 family inet6 address 201:DB8:251::bb:bb:1/126set interfaces ae0
aggregated-ether-options bfd-liveness-detection minimum-interval 150set interfaces ae0
aggregated-ether-options bfd-liveness-detection multiplier 3set interfaces ae0 aggregated-ether-
options bfd-liveness-detection neighbor 10.255.106.107set interfaces ae0 aggregated-ether-
options bfd-liveness-detection local-address 10.255.106.102set interfaces ae0 aggregated-ether-
options minimum-links 1set interfaces ae0 aggregated-ether-options link-speed 10gset interfaces
ae0 aggregated-ether-options lacp passiveset interfaces ae0 unit 0 family inet address
10.0.0.2/30set interfaces ae1 aggregated-ether-options bfd-liveness-detection minimum-interval
200set interfaces ae1 aggregated-ether-options bfd-liveness-detection multiplier 3set interfaces
ae1 aggregated-ether-options bfd-liveness-detection neighbor 201:DB8:251::aa:aa:1set interfaces
ae1 aggregated-ether-options bfd-liveness-detection local-address 201:DB8:251::bb:bb:1set
interfaces ae1 aggregated-ether-options minimum-links 1set interfaces ae1 aggregated-ether-
options link-speed 10gset interfaces ae1 aggregated-ether-options lacp passiveset interfaces ae1
unit 0 family inet6 address 5555::2/126set routing-options static route 20.20.20.0/30 next-hop
10.0.0.1set routing-options rib inet6.0 static route 3ffe::1:1/126 next-hop 5555::1
```

Configuring a Micro BFD Session for Aggregated Ethernet Interfaces

Procedure

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see "*Using the CLI Editor in Configuration Mode*" in the CLI User Guide.

NOTE: Repeat this procedure for Router R1, modifying the appropriate interface names, addresses, and any other parameters for each router.

To configure a micro BFD session for aggregated Ethernet interfaces on Router R0:

1. Configure the physical interfaces.

```
[edit interfaces]
user@R0# set ge-1/0/1 unit 0 family inet address 20.20.20.1/30
user@R0# set ge-1/0/1 unit 0 family inet6 address 3ffe::1:1/126
user@R0# set xe-4/0/0 gigether-options 802.3ad ae0
user@R0# set xe-4/0/1 gigether-options 802.3ad ae0
user@R0# set xe-4/1/0 gigether-options 802.3ad ae1
user@R0# set xe-4/1/1 gigether-options 802.3ad ae1
```

2. Configure the loopback interface.

```
[edit interfaces]
user@R0# set lo0 unit 0 family inet address 10.255.106.107/32
user@R0# set lo0 unit 0 family inet6 address 201:DB8:251::aa:aa:1/128
```

3. Configure an IP address on the aggregated Ethernet interface ae0 with either IPv4 or IPv6 addresses, as per your network requirements.

```
[edit interfaces]
user@R0# set ae0 unit 0 family inet address 10.0.0.1/30
```

4. Set the routing option, create a static route, and set the next-hop address.

NOTE: You can configure either an IPv4 or IPv6 static route, depending on your network requirements.

```
[edit routing-options]
user@R0# set nonstop-routing
user@R0# set static route 30.30.30.0/30 next-hop 10.0.0.2
user@R0# set rib inet6.0 static route 3ffe::1:2/126 next-hop 5555::2
```

5. Configure the Link Aggregation Control Protocol (LACP).

```
[edit interfaces]
user@R0# set ae0 aggregated-ether-options lacp active
```

6. Configure BFD for the aggregated Ethernet interface ae0, and specify the minimum interval, local IP address, and the neighbor IP address.

```
[edit interfaces]
user@R0# set ae0 aggregated-ether-options bfd-liveness-detection minimum-interval 100
user@R0# set ae0 aggregated-ether-options bfd-liveness-detection multiplier 3
user@R0# set ae0 aggregated-ether-options bfd-liveness-detection neighbor 10.255.106.102
user@R0# set ae0 aggregated-ether-options bfd-liveness-detection local-address 10.255.106.107
user@R0# set ae0 aggregated-ether-options minimum-links 1
user@R0# set ae0 aggregated-ether-options link-speed 10g
```

7. Configure an IP addresse on the aggregated Ethernet interface ae1.

   You can assign either IPv4 or IPv6 addresses as per your network requirements.

```
[edit interfaces]
user@R0# set ae1 unit 0 family inet6 address 5555::1/126
```

8. Configure BFD for the aggregated Ethernet interface ae1.

```
[edit interfaces]
user@R0# set ae1 aggregated-ether-options bfd-liveness-detection minimum-interval 100
user@R0# set ae1 aggregated-ether-options bfd-liveness-detection multiplier 3
user@R0# set ae1 aggregated-ether-options bfd-liveness-detection neighbor 201:DB8:251::bb:bb:1
user@R0# set ae1 aggregated-ether-options bfd-liveness-detection local-address
201:DB8:251::aa:aa:1
user@R0# set ae1 aggregated-ether-options minimum-links 1
user@R0# set ae1 aggregated-ether-options link-speed 10g
```

NOTE: Beginning with Junos OS Release 16.1, you can also configure this feature with the AE interface address as the local address in a micro BFD session.

Beginning with Release 16.1R2, Junos OS checks and validates the configured micro BFD `local-address` against the interface or loopback IP address before the configuration commit. Junos OS performs this check on both IPv4 and IPv6 micro BFD address configurations, and if they do not match, the commit fails.

9. Configure tracing options for BFD for troubleshooting.

```
[edit protocols]
user@R0# set bfd traceoptions file bfd
user@R0# set bfd traceoptions file size 100m
user@R0# set bfd traceoptions file files 10
user@R0# set bfd traceoptions flag all
```

Results

From configuration mode, enter the **show interfaces**, **show protocols**, and **show routing-options** commands and confirm your configuration. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0> show interfaces
traceoptions {
    flag bfd-events;
}
ge-1/0/1 {
    unit 0 {
        family inet {
            address 20.20.20.1/30;
        }
        family inet6 {
            address 3ffe::1:1/126;
        }
    }
}
xe-4/0/0 {
    enable;
    gigether-options {
        802.3ad ae0;
    }
}
xe-4/0/1 {
```

```
    gigether-options {
        802.3ad ae0;
    }
}
xe-4/1/0 {
    enable;
    gigether-options {
        802.3ad ae1;
    }
}
xe-4/1/1 {
    gigether-options {
        802.3ad ae1;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.106.107/32;
        }
        family inet6 {
            address  201:DB8:251::aa:aa:1/128;
        }
    }
}
ae0 {
    aggregated-ether-options {
        bfd-liveness-detection {
            minimum-interval 100;
            neighbor 10.255.106.102;
            local-address 10.255.106.107;
        }
        minimum-links 1;
        link-speed 10g;
        lacp {
            active;
        }
    }
    unit 0 {
        family inet {
            address 10.0.0.1/30;
        }
    }
```

```
}
ae1 {
    aggregated-ether-options {
        bfd-liveness-detection {
            minimum-interval 100;
            multiplier 3;
            neighbor 201:DB8:251::bb:bb:1;
            local-address 201:DB8:251::aa:aa:1;
        }
        minimum-links 1
        link-speed 10g;
    }
    unit 0 {
        family inet6 {
            address 5555::1/126;
        }
    }
}
```

```
user@R0> show protocols
bfd {
    traceoptions {
        file bfd size 100m files 10;
        flag all;
    }
}
```

```
user@R0> show routing-options
nonstop-routing ;
    rib inet6.0 {
    static {
        route 3ffe:1:2/126 {
            next-hop 5555::2;
        }
    }
}
static {
    route 30.30.30.0/30 {
        next-hop 10.0.0.2;
```

```
    }
}
```

If you are done configuring the device, commit the configuration.

```
user@R0# commit
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying That the Independent BFD Sessions Are Up

Purpose

Verify that the micro BFD sessions are up, and view details about the BFD sessions.

Action

From operational mode, enter the `show bfd session extensive` command.

```
user@R0> show bfd session extensive
                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
10.255.106.102          Up        xe-4/0/0      9.000     3.000        3
 Client LACPD, TX interval 0.100, RX interval 0.100
 Session up time 4d 23:13, previous down time 00:00:06
 Local diagnostic None, remote diagnostic None
 Remote heard, hears us, version 1
 Replicated
 Session type: Micro BFD
 Min async interval 0.100, min slow interval 1.000
```

```
 Adaptive async TX interval 0.100, RX interval 0.100
 Local min TX interval 0.100, minimum RX interval 0.100, multiplier 3
 Remote min TX interval 3.000, min RX interval 3.000, multiplier 3
 Local discriminator 21, remote discriminator 75
 Echo mode disabled/inactive
 Remote is control-plane independent
  Session ID: 0x0

                                              Detect   Transmit
Address                 State     Interface   Time     Interval  Multiplier
10.255.106.102          Up        xe-4/0/1    9.000     3.000        3
 Client LACPD, TX interval 0.100, RX interval 0.100
 Session up time 4d 23:13, previous down time 00:00:07
 Local diagnostic None, remote diagnostic None
 Remote heard, hears us, version 1
 Replicated
 Session type: Micro BFD
 Min async interval 0.100, min slow interval 1.000
 Adaptive async TX interval 0.100, RX interval 0.100
 Local min TX interval 0.100, minimum RX interval 0.100, multiplier 3
 Remote min TX interval 3.000, min RX interval 3.000, multiplier 3
 Local discriminator 19, remote discriminator 74
 Echo mode disabled/inactive
 Remote is control-plane independent
  Session ID: 0x0

                                              Detect   Transmit
Address                 State     Interface   Time     Interval  Multiplier
201:DB8:251::bb:bb:1    Up        xe-4/1/1    9.000     3.000        3
 Client LACPD, TX interval 0.100, RX interval 0.100
 Session up time 4d 23:13
 Local diagnostic None, remote diagnostic None
 Remote not heard, hears us, version 1
 Replicated
 Session type: Micro BFD
 Min async interval 0.100, min slow interval 1.000
 Adaptive async TX interval 0.100, RX interval 0.100
 Local min TX interval 1.000, minimum RX interval 0.100, multiplier 3
 Remote min TX interval 3.000, min RX interval 3.000, multiplier 3
 Local discriminator 17, remote discriminator 67
 Echo mode disabled/inactive, no-absorb, no-refresh
 Remote is control-plane independent
  Session ID: 0x0
```

```
                                                Detect   Transmit
Address                  State     Interface    Time     Interval  Multiplier
201:DB8:251::bb:bb:1     UP        xe-4/1/0     9.000     3.000        3
 Client LACPD, TX interval 0.100, RX interval 0.100
 Session up time 4d 23:13
 Local diagnostic None, remote diagnostic None
 Remote not heard, hears us, version 1
 Replicated
 Session type: Micro BFD
 Min async interval 0.100, min slow interval 1.000
 Adaptive async TX interval 0.100, RX interval 0.100
 Local min TX interval 1.000, minimum RX interval 0.100, multiplier 3
 Remote min TX interval 3.000, min RX interval 3.000, multiplier 3
 Local discriminator 16, remote discriminator 66
 Echo mode disabled/inactive, no-absorb, no-refresh
 Remote is control-plane independent
  Session ID: 0x0

4 sessions, 4 clients
Cumulative transmit rate 2.0 pps, cumulative receive rate 1.7 pps
```

Meaning

The Micro BFD field represents the independent micro BFD sessions running on the links in a LAG. The TX interval *item*, RX interval *item* output represents the setting configured with the `minimum-interval` statement. All of the other output represents the default settings for BFD. To modify the default settings, include the optional statements under `bfd-liveness-detection` statement.

Viewing Detailed BFD Events

Purpose

View the contents of the BFD trace file to assist in troubleshooting, if required.

Action

From operational mode, enter the **file show /var/log/bfd** command.

```
user@R0> file show /var/log/bfd
Jun  5 00:48:59    Protocol (1) len 1: BFD
```

```
Jun  5 00:48:59    Data (9) len 41: (hex) 42 46 44 20 6e 65 69 67 68 62 6f 72 20 31 30 2e 30 2e
30
Jun  5 00:48:59 PPM Trace: BFD neighbor 10.255.106.102 (IFL 349) set, 9 0
Jun  5 00:48:59 Received Downstream RcvPkt (19) len 108:
Jun  5 00:48:59    IfIndex (3) len 4: 329
Jun  5 00:48:59    Protocol (1) len 1: BFD
Jun  5 00:48:59    SrcAddr (5) len 8: 10.255.106.102
Jun  5 00:48:59    Data (9) len 24: (hex) 00 88 03 18 00 00 00 4b 00 00 00 15 00 2d c6 c0 00 2d
c6
Jun  5 00:48:59    PktError (26) len 4: 0
Jun  5 00:48:59    RtblIdx (24) len 4: 0
Jun  5 00:48:59    MultiHop (64) len 1: (hex) 00
Jun  5 00:48:59    Unknown (168) len 1: (hex) 01
Jun  5 00:48:59    Unknown (171) len 2: (hex) 02 3d
Jun  5 00:48:59    Unknown (172) len 6: (hex) 80 71 1f c7 81 c0
Jun  5 00:48:59    Authenticated (121) len 1: (hex) 01
Jun  5 00:48:59 BFD packet from 10.0.0.2 (IFL 329), len 24
Jun  5 00:48:59    Ver 0, diag 0, mult 3, len 24
Jun  5 00:48:59    Flags: IHU Fate
Jun  5 00:48:59    My discr 0x0000004b, your discr 0x00000015
Jun  5 00:48:59    Tx ivl 3000000, rx ivl 3000000, echo rx ivl 0
Jun  5 00:48:59 [THROTTLE]bfdd_rate_limit_can_accept_pkt: session 10.255.106.102 is up or
already in program thread
Jun  5 00:48:59 Replicate: marked session (discr 21) for update
```

Meaning

BFD messages are being written to the specified trace file.

Configuring BFD for PIM

The Bidirectional Forwarding Detection (BFD) Protocol is a simple hello mechanism that detects failures in a network. BFD works with a wide variety of network environments and topologies. A pair of routing devices exchanges BFD packets. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. The BFD failure detection timers have shorter time limits than the Protocol Independent Multicast (PIM) hello hold time, so they provide faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the `clear bfd adaptation` command to return BFD interval timers to their configured values. The `clear bfd adaptation` command is hitless, meaning that the command does not affect traffic flow on the routing device.

You must specify the minimum transmit and minimum receive intervals to enable BFD on PIM.

To enable failure detection:

1. Configure the interface globally or in a routing instance.

   This example shows the global configuration.

   ```
   [edit protocols pim]
   user@host# edit interface fe-1/0/0.0 family inet bfd-liveness-detection
   ```

2. Configure the minimum transmit interval.

   This is the minimum interval after which the routing device transmits hello packets to a neighbor with which it has established a BFD session. Specifying an interval smaller than 300 ms can cause undesired BFD flapping.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set transmit-interval 350
   ```

3. Configure the minimum interval after which the routing device expects to receive a reply from a neighbor with which it has established a BFD session.

   Specifying an interval smaller than 300 ms can cause undesired BFD flapping.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set minimum-receive-interval 350
   ```

4. (Optional) Configure other BFD settings.

As an alternative to setting the receive and transmit intervals separately, configure one interval for both.

```
[edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
user@host# set minimum-interval 350
```

5. Configure 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.

```
[edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
user@host# set detection-time threshold 800
```

6. Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

```
[edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
user@host# set multiplier 50
```

7. Configure the BFD version.

```
[edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
user@host# set version 1
```

8. Specify that BFD sessions should not adapt to changing network conditions.

   We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

```
[edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
user@host# set no-adaptation
```

9. Verify the configuration by checking the output of the `show bfd session` command.

Enabling Dedicated and Real-Time BFD on SRX Series Firewalls

IN THIS SECTION

By default, SRX Series Firewalls operate in centralized BFD mode. They also support distributed BFD, dedicated BFD, and real-time BFD.

Dedicated BFD

Enabling dedicated BFD impacts traffic throughput as one CPU core is removed from data plane processing.

To enable dedicated BFD on the SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, vSRX, and vSRX3.0 devices:

1. Include the `dedicated-ukern-cpu` statement at the `[edit chassis]` hierarchy level and then commit the configuration.

   a. `[edit]`

   b. user@host# **set chassis dedicated-ukern-cpu**

      user@host# **commit**

      The following warning message to reboot the system displays when you commit the configuration:

      ```
      warning: Packet processing throughput may be impacted in dedicated-ukernel-cpu mode. warning: A reboot is required for dedicated-ukernel-cpu mode to be enabled. Please use "request system reboot" to reboot the system. commit complete
      ```

2. Reboot the device to enable the configuration:

   a. user@host> **request system reboot**

3. Verify that dedicated BFD is enabled.

user@host> **show chassis dedicated-ukern-cpu**

```
Dedicated Ukern CPU Status: Enabled
```

Real-Time BFD

Enabling real-time BFD does not impact data plane performance. Higher priority is given to the Packet Forwarding Engine process handling BFD in distributed mode. This is suitable for scenarios where less than half of the maximum number of BFD sessions are being used. See "this list" on page 124 for the maximum number of BFD sessions supported per SRX device.

NOTE: For more information about BFD in distributed mode, see "Understanding How BFD Detects Network Failures" on page 25.

To enable real-time BFD on SRX300, SRX320, SRX340, and SRX345 devices:

1. Include the `realtime-ukern-thread` statement at the `[edit chassis]` hierarchy level and then commit the configuration.

   a. [edit]

   b. user@host# **set chassis realtime-ukern-thread**

      user@host# **commit**

      The following warning message to reboot the system displays when you commit the configuration:

      ```
      WARNING: realtime-ukern-thread is enable. Please use the command request system reboot.
      ```

2. Reboot the device to enable the configuration:

   a. user@host> **request system reboot**

3. Verify that real-time BFD is enabled.

   user@host> **show chassis realtime-ukern-thread**

   ```
   realtime Ukern thread Status: Enabled
   ```

BFD Support By SRX Platform

SRX Series Firewalls support the following maximum number of BFD sessions:

- Up to four sessions on SRX300 and SRX320 devices.
- Up to 50 sessions on SRX340, SRX345, and SRX380 devices.

- Up to 120 sessions on SRX1500 devices.

On all SRX Series Firewalls, high CPU utilization triggered for reasons such as CPU intensive commands and SNMP walks causes the BFD protocol to flap while processing large BGP updates. (Platform support depends on the Junos OS release in your installation.)

SRX Series Firewalls operating in chassis cluster mode support only BFD centralized mode.

The table below shows the BFD modes supported on each SRX Series Firewall.

Table 3: BFD Modes Supported on SRX Series Firewalls

SRX Series Firewall	Centralized BFD Mode	Distributed BFD	Real-Time BFD	Dedicated Core
SRX300	Default	Configuration	Configuration (Optional)	Not supported
SRX320	Default	Configuration	Configuration (Optional)	Not supported
SRX340	Default	Configuration	Configuration	Configuration (Optional)
SRX345	Default	Configuration	Configuration	Configuration (Optional)
SRX380	Default	Configuration	Configuration	Configuration (Optional)
SRX1500	BFD failure detection time >= 500 ms and dedicated mode is not enabled	BFD failure detection time < 500 ms and dedicated mode is not enabled	Not supported	Configuration
SRX4100	BFD failure detection time >= 500 ms	BFD failure detection time < 500 ms	Not supported	Not supported
SRX4200	BFD failure detection time >= 500 ms	BFD failure detection time < 500 ms	Not supported	Not supported

Table 3: BFD Modes Supported on SRX Series Firewalls *(Continued)*

SRX Series Firewall	Centralized BFD Mode	Distributed BFD	Real-Time BFD	Dedicated Core
SRX4600	BFD failure detection time >= 500 ms	BFD failure detection time < 500 ms	Not supported	Not supported
SRX5000 line of devices with SPC2 card	Default	Not supported	Not supported	Not supported
SRX5000 line of devices with SPC3 card	BFD failure detection time >= 500 ms	BFD failure detection time < 500 ms	Not supported	Not supported
vSRX 3.0	BFD failure detection time > 500ms	BFD failure detection time <= 500ms	Not supported	Configuration

RELATED DOCUMENTATION

Understanding How BFD Detects Network Failures | 25

PART 4

Configuring Routing Engine Redundancy

Understanding Routing Engine Redundancy

SUMMARY

Routing engine redundancy ensures the continued functionality of your network. If the primary Routing Engine is taken offline (either by failover or switchover), the standby Routing Engine takes over all routing functions.

IN THIS SECTION

Routing Engine Redundancy Overview

Redundant Routing Engines are two Routing Engines that are installed in the same routing platform. One functions as the primary, while the other stands by as a backup should the primary Routing Engine fail. On routing platforms with dual Routing Engines, network reconvergence takes place more quickly than on routing platforms with a single Routing Engine.

When a Routing Engine is configured as primary, it has full functionality. It receives and transmits routing information, builds and maintains routing tables, communicates with interfaces and Packet Forwarding Engine components, and has full control over the chassis. When a Routing Engine is configured to be the backup, it does not communicate with the Packet Forwarding Engine or chassis components.

NOTE: You cannot configure multiple Routing Engines as primary. This configuration causes the commit check to fail.

A failover from the primary Routing Engine to the backup Routing Engine occurs automatically when the primary Routing Engine experiences a hardware failure or when you have configured the software to support a change in primary role based on specific conditions. You can also manually switch Routing Engine primary role by issuing one of the `request chassis routing-engine` commands. In this topic, the term *failover* refers to an automatic event, whereas *switchover* refers to either an automatic or a manual event.

When a failover or a switchover occurs, the backup Routing Engine takes control of the system as the new primary Routing Engine.

- If *graceful Routing Engine switchover* is not configured, when the backup Routing Engine becomes primary, it resets the switch plane and downloads its own version of the microkernel to the Packet Forwarding Engine components. Traffic is interrupted while the Packet Forwarding Engine is reinitialized. All kernel and forwarding processes are restarted.
- If graceful Routing Engine switchover is configured, interface and kernel information is preserved. The switchover is faster because the Packet Forwarding Engines are not restarted. The new primary Routing Engine restarts the routing protocol process (rpd). All hardware and interfaces are acquired by a process that is similar to a warm restart.
- If graceful Routing Engine switchover and *nonstop active routing* (NSR) are configured, traffic is not interrupted during the switchover. Interface, kernel, and routing protocol information is preserved.
- If graceful Routing Engine switchover and graceful restart are configured, traffic is not interrupted during the switchover. Interface and kernel information is preserved. Graceful restart protocol extensions quickly collect and restore routing information from the neighboring routers.

Conditions That Trigger a Routing Engine Failover

The following events can result in an automatic change in Routing Engine primary role, depending on your configuration:

- The routing platform experiences a hardware failure. A change in Routing Engine primary role occurs if either the Routing Engine or the associated host module or subsystem is abruptly powered off. You can also configure the backup Routing Engine to take primary role if it detects a hard disk error on the primary Routing Engine. To enable this feature, include the `failover on-disk-failure` statement at the `[edit chassis redundancy]` hierarchy level.
- The routing platform experiences a software failure, such as a kernel crash or a CPU lock. You must configure the backup Routing Engine to take primary role when it detects a loss of keepalive signal. To enable this failover method, include the `failover on-loss-of-keepalives` statement at the `[edit chassis redundancy]` hierarchy level.
- The routing platform experiences an em0 interface failure on the primary Routing Engine. You must configure the backup Routing Engine to take primary role when it detects the em0 interface failure. To enable this failover method, include the `on-re-to-fpc-stale` statement at the `[edit chassis redundancy failover]` hierarchy level.
- A specific software process fails. You can configure the backup Routing Engine to take primary role when one or more specified processes fail at least four times within 30 seconds. Include the `failover other-routing-engine` statement at the `[edit system processes` *`process-name`*`]` hierarchy level.

If any of these conditions is met, a message is logged and the backup Routing Engine attempts to take primary role. By default, an alarm is generated when the backup Routing Engine becomes active. After the backup Routing Engine takes primary role, it continues to function as primary even after the originally configured primary Routing Engine has successfully resumed operation. You must manually restore it to its previous backup status. (However, if at any time one of the Routing Engines is not present, the other Routing Engine becomes primary automatically, regardless of how redundancy is configured.)

Default Routing Engine Redundancy Behavior

By default, Junos OS uses **re0** as the primary Routing Engine and **re1** as the backup Routing Engine. Unless otherwise specified in the configuration, **re0** always becomes primary when the acting primary Routing Engine is rebooted.

NOTE: A single Routing Engine in the chassis always becomes the primary Routing Engine even if it was previously the backup Routing Engine.

Perform the following steps to see how the default Routing Engine redundancy setting works:

1. Ensure that **re0** is the primary Routing Engine.
2. Manually switch the state of Routing Engine primary role by issuing the `request chassis routing-engine master switch` command from the primary Routing Engine. **re0** is now the backup Routing Engine and **re1** is the primary Routing Engine.

   NOTE: On the next reboot of the primary Routing Engine, Junos OS returns the router to the default state because you have not configured the Routing Engines to maintain this state after a reboot.

3. Reboot the primary Routing Engine **re1**.

   The Routing Engine boots up and reads the configuration. Because you have not specified in the configuration which Routing Engine is the primary, **re1** uses the default configuration as the backup. Now both **re0** and **re1** are in a backup state. Junos OS detects this conflict and, to prevent a no-primary state, reverts to the default configuration to direct **re0** to become primary.

Situations That Require You to Halt Routing Engines

Before you shut the power off to a routing platform that has two Routing Engines or before you remove the primary Routing Engine, you must first halt the backup Routing Engine and then halt the primary Routing Engine. Otherwise, you might need to reinstall Junos OS. You can use the `request system halt both-routing-engines` command on the primary Routing Engine, which first shuts down the primary Routing Engine and then shuts down the backup Routing Engine. To shut down only the backup Routing Engine, issue the `request system halt` command on the backup Routing Engine.

If you halt the primary Routing Engine and do not power it off or remove it, the backup Routing Engine remains inactive unless you have configured it to become the primary when it detects a loss of keepalive signal from the primary Routing Engine.

NOTE: To restart the router, you must log in to the console port (rather than the Ethernet management port) of the Routing Engine. When you log in to the console port of the primary Routing Engine, the system automatically reboots. After you log in to the console port of the backup Routing Engine, press Enter to reboot it.

NOTE: If you have upgraded the backup Routing Engine, first reboot it and then reboot the primary Routing Engine.

RELATED DOCUMENTATION

Understanding High Availability Features on Juniper Networks Routers | 2

Understanding Switching Control Board Redundancy | 15

Configuring Routing Engine Redundancy | 132

Configuring Routing Engine Redundancy

SUMMARY

Follow the steps and examples below to configure routing engine redundancy.

IN THIS SECTION

NOTE: To complete the tasks in the following sections, **re0** and **re1** configuration groups must be defined. For more information about configuration groups, see the Junos OS CLI User Guide.

Modifying the Default Routing Engine Primary Role

For routers with two Routing Engines, you can configure which Routing Engine is the primary and which is the backup. By default, the Routing Engine in slot 0 is the primary (**re0**) and the one in slot 1 is the backup (**re1**).

NOTE: In systems with two Routing Engines, both Routing Engines cannot be configured to be primary at the same time. This configuration causes the commit check to fail.

To modify the default configuration, include the `routing-engine` statement at the `[edit chassis redundancy]` hierarchy level:

```
[edit chassis redundancy]
routing-engine slot-number (master | backup | disabled);
```

slot-number can be 0 or 1. To configure the Routing Engine to be the primary, specify the **master** option. To configure it to be the backup, specify the **backup** option. To disable a Routing Engine, specify the **disabled** option.

NOTE: To switch between the primary and the backup Routing Engines, see "Manually Switching Routing Engine Primary Role" on page 136.

Configuring Automatic Failover to the Backup Routing Engine

IN THIS SECTION

The following sections describe how to configure automatic failover to the backup Routing Engine when certain failures occur on the primary Routing Engine.

Without Interruption to Packet Forwarding

For routers with two Routing Engines, you can configure graceful Routing Engine switchover (GRES). When graceful switchover is configured, socket reconnection occurs seamlessly without interruption to packet forwarding. For information about how to configure graceful Routing Engine switchover, see "Configuring Graceful Routing Engine Switchover" on page 208.

On Detection of a Hard Disk Error on the Primary Routing Engine

After you configure a backup Routing Engine, you can direct it to take primary role automatically if it detects a hard disk error from the primary Routing Engine. To enable this feature, include the `on-disk-failure` statement at the `[edit chassis redundancy failover]` hierarchy level.

```
[edit chassis redundancy failover]
on-disk-failure;
```

NOTE: The `on-disk-failure` statement at the `[edit chassis redundancy]` hierarchy level is not supported on PTX platforms running Junos Evolved. These platforms default to a switchover when disk failure is detected.

On Detection of a Broken LCMD Connectivity Between the VM and RE

Set the following configuration that will result in an automatic RE switchover when the LCMD connectivity between VM and RE is broken. To enable this feature, include the `on-loss-of-vm-host-connection` statement at the `[edit chassis redundancy failover]` hierarchy level.

```
[edit chassis redundancy failover]
on-loss-of-vm-host-connection;
```

If the LCMD process is crashing on the primary, the system will switchover after one minute provided the backup RE LCMD connection is stable. The system will not switchover under the following conditions: if the backup RE LCMD connection is unstable or if the current primary just gained primary role. When the primary has just gained primary role, the switchover happens only after four minutes.

On Detection of a Loss of Keepalive Signal from the Primary Routing Engine

After you configure a backup Routing Engine, you can direct it to take primary role automatically if it detects a loss of keepalive signal from the primary Routing Engine.

To enable failover on receiving a loss of keepalive signal, include the `on-loss-of-keepalives` statement at the `[edit chassis redundancy failover]` hierarchy level:

```
[edit chassis redundancy failover]
on-loss-of-keepalives;
```

NOTE: The `on-loss-of-keepalives` statement at the `[edit chassis redundancy]` hierarchy is not supported on PTX platforms running Junos Evolved. These platforms default to a switchover when keepalive messages are not detected.

When graceful Routing Engine switchover is not configured, by default, failover occurs after 300 seconds (5 minutes). You can configure a shorter or longer time interval.

NOTE: The keepalive time period is reset to 360 seconds when the primary Routing Engine has been manually rebooted or halted.

To change the keepalive time period, include the `keepalive-time` statement at the `[edit chassis redundancy]` hierarchy level:

```
[edit chassis redundancy]
keepalive-time seconds;
```

The range for **keepalive-time** is 2 through 10,000 seconds.

The following example describes the sequence of events if you configure the backup Routing Engine to detect a loss of keepalive signal in the primary Routing Engine:

1. Manually configure a **keepalive-time** of 25 seconds.
2. After the Packet Forwarding Engine connection to the primary Routing Engine is lost and the keepalive timer expires, packet forwarding is interrupted.
3. After 25 seconds of keepalive loss, a message is logged, and the backup Routing Engine attempts to take primary role. An alarm is generated when the backup Routing Engine becomes active, and the display is updated with the current status of the Routing Engine.
4. After the backup Routing Engine takes primary role, it continues to function as primary.

NOTE: When graceful Routing Engine switchover is configured, the keepalive signal is automatically enabled and the failover time is set to 2 seconds. You cannot manually reset the keepalive time.

NOTE: When you halt or reboot the primary Routing Engine, Junos OS resets the keepalive time to 360 seconds, and the backup Routing Engine does not take over primary role until the 360-second keepalive time period expires.

A former primary Routing Engine becomes a backup Routing Engine if it returns to service after a failover to the backup Routing Engine. To restore primary status to the former primary Routing Engine, you can use the **request chassis routing-engine master switch** operational mode command.

If at any time one of the Routing Engines is not present, the remaining Routing Engine becomes primary automatically, regardless of how redundancy is configured.

On Detection of the em0 Interface Failure on the Primary Routing Engine

After you configure a backup Routing Engine, you instruct it to take primary role automatically if the em0 interface fails on the primary Routing Engine. To enable this feature, include the `on-re-to-fpc-stale` statement at the `[edit chassis redundancy failover]` hierarchy level.

```
[edit chassis redundancy failover]
on-re-to-fpc-stale;
```

When a Software Process Fails

To configure automatic switchover to the backup Routing Engine if a software process fails, include the `failover other-routing-engine` statement at the `[edit system processes` *`process-name`*`]` hierarchy level:

```
[edit system processes process-name]
failover other-routing-engine;
```

process-name is one of the valid process names. If this statement is configured for a process, and that process fails four times within 30 seconds, the router reboots from the other Routing Engine. Another statement available at the `[edit system processes]` hierarchy level is **failover alternate-media**. For information about the alternate media option, see the Junos OS Administration Library for Routing Devices.

Manually Switching Routing Engine Primary Role

To manually switch Routing Engine primary role, use one of the following commands:

- On the backup Routing Engine, request that the backup Routing Engine take primary role by issuing the `request chassis routing-engine master acquire` command.
- On the primary Routing Engine, request that the backup Routing Engine take primary role by using the `request chassis routing-engine master release` command.

- On either Routing Engine, switch primary role by issuing the `request chassis routing-engine master switch` command.

Verifying Routing Engine Redundancy Status

A separate log file is provided for redundancy logging at **/var/log/mastership**. To view the log, use the `file show /var/log/mastership` command. Table 4 on page 137 lists the primary role log event codes and descriptions.

Table 4: Routing Engine Primary Role Log

Event Code	Description
E_NULL = 0	The event is a null event.
E_CFG_M	The Routing Engine is configured as primary.
E_CFG_B	The Routing Engine is configured as backup.
E_CFG_D	The Routing Engine is configured as disabled.
E_MAXTRY	The maximum number of tries to acquire or release primary role was exceeded.
E_REQ_C	A claim primary role request was sent.
E_ACK_C	A claim primary role acknowledgement was received.
E_NAK_C	A claim primary role request was not acknowledged.
E_REQ_Y	Confirmation of primary role is requested.
E_ACK_Y	Primary Role is acknowledged.
E_NAK_Y	Primary Role is not acknowledged.

Table 4: Routing Engine Primary Role Log *(Continued)*

Event Code	Description
E_REQ_G	A release primary role request was sent by a Routing Engine.
E_ACK_G	The Routing Engine acknowledged release of primary role.
E_CMD_A	The command **request chassis routing-engine master acquire** was issued from the backup Routing Engine.
E_CMD_F	The command **request chassis routing-engine master acquire force** was issued from the backup Routing Engine.
E_CMD_R	The command **request chassis routing-engine master release** was issued from the primary Routing Engine.
E_CMD_S	The command request **chassis routing-engine master switch** was issued from a Routing Engine.
E_NO_ORE	No other Routing Engine is detected.
E_TMOUT	A request timed out.
E_NO_IPC	Routing Engine connection was lost.
E_ORE_M	Other Routing Engine state was changed to primary.
E_ORE_B	Other Routing Engine state was changed to backup.
E_ORE_D	Other Routing Engine state was changed to disabled.

Check Overall CPU and Memory Usage

IN THIS SECTION

Purpose

You can display exhaustive system process information about software processes that are running on the router and have controlling terminals. This command is equivalent to the UNIX top command. However, the UNIX top command shows real-time memory usage, with the memory values constantly changing, while the show system processes extensive command provides a snapshot of memory usage in a given moment.

Action

To check overall CPU and memory usage, enter the following Junos OS command-line interface (CLI) command:

```
user@host> show system processes extensive
```

Sample Output

user@R1> **show system processes extensive**

```
last pid:  5251;  load averages:  0.00,  0.00,  0.00  up 4+20:22:16    10:44:41
58 processes:  1 running, 57 sleeping
Mem: 57M Active, 54M Inact, 17M Wired, 184K Cache, 35M Buf, 118M Free
Swap:  512M Total, 512M Free
  PID USERNAME PRI NICE  SIZE    RES STATE    TIME   WCPU    CPU COMMAND
 4480 root       2   0  3728K  1908K select 231:17  2.34%  2.34% chassisd
 4500 root       2   0  1896K   952K select   0:36  0.00%  0.00% fud
```

```
 4505 root       2   0  1380K   736K select   0:35  0.00%  0.00% irsd
 4481 root       2   0  1864K   872K select   0:32  0.00%  0.00% alarmd
 4488 root       2   0  8464K  4600K kqread   0:28  0.00%  0.00% rpd
 4501 root       2 -15  1560K   968K select   0:21  0.00%  0.00% ppmd
 4510 root       2   0  1372K   812K select   0:13  0.00%  0.00% bfdd
    5 root      18   0     0K     0K syncer   0:09  0.00%  0.00% syncer
 4485 root       2   0  3056K  1776K select   0:07  0.00%  0.00% snmpd
 4499 root       2   0  3688K  1676K select   0:05  0.00%  0.00% kmd
 4486 root       2   0  3760K  1748K select   0:05  0.00%  0.00% mib2d
 4493 root       2   0  1872K   928K select   0:03  0.00%  0.00% pfed
 4507 root       2   0  1984K  1052K select   0:02  0.00%  0.00% fsad
 4518 root       2   0  3780K  2400K select   0:02  0.00%  0.00% dcd
    8 root     -18   0     0K     0K psleep   0:02  0.00%  0.00% vmuncachedaemo
    4 root     -18   0     0K     0K psleep   0:02  0.00%  0.00% bufdaemon
 4690 root       2   0     0K     0K peer_s   0:01  0.00%  0.00% peer proxy
 4504 root       2   0  1836K   968K select   0:01  0.00%  0.00% dfwd
 4477 root       2   0   992K   320K select   0:01  0.00%  0.00% watchdog
 4354 root       2   0  1116K   604K select   0:01  0.00%  0.00% syslogd
 4492 root      10   0  1004K   400K nanslp   0:01  0.00%  0.00% tnp.sntpd
 4446 root      10   0  1108K   616K nanslp   0:01  0.00%  0.00% cron
 4484 root       2   0 15716K  7468K select   0:01  0.00%  0.00% mgd
 4494 root       2  15  2936K  2036K select   0:01  0.00%  0.00% sampled
 5245 remote     2   0  8340K  3472K select   0:01  0.00%  0.00% cli
    2 root     -18   0     0K     0K psleep   0:00  0.00%  0.00% pagedaemon
 4512 root       2   0  2840K  1400K select   0:00  0.00%  0.00% l2tpd
    1 root      10   0   852K   580K wait     0:00  0.00%  0.00% init
 5244 root       2   0  1376K   784K select   0:00  0.00%  0.00% telnetd
 4509 root      10   0  1060K   528K nanslp   0:00  0.00%  0.00% eccd
 4508 root       2   0  2264K  1108K select   0:00  0.00%  0.00% spd
 2339 root      10   0   514M 17260K mfsidl   0:00  0.00%  0.00% newfs
 4497 root       2   0  2432K  1152K select   0:00  0.00%  0.00% cosd
 4490 root       2 -15  2356K  1020K select   0:00  0.00%  0.00% apsd
 4496 root       2   0  2428K  1108K select   0:00  0.00%  0.00% rmopd
 4491 root       2   0  2436K  1104K select   0:00  0.00%  0.00% vrrpd
 4487 root       2   0 15756K  7648K sbwait   0:00  0.00%  0.00% mgd
 5246 root       2   0 15776K  8336K select   0:00  0.00%  0.00% mgd
    0 root     -18   0     0K     0K sched    0:00  0.00%  0.00% swapper
 5251 root      30   0 21732K   840K RUN      0:00  0.00%  0.00% top
 4511 root       2   0  1964K   908K select   0:00  0.00%  0.00% pgmd
 4502 root       2   0  1960K   956K select   0:00  0.00%  0.00% lmpd
 4495 root       2   0  1884K   876K select   0:00  0.00%  0.00% ilmid
 4482 root       2   0  1772K   776K select   0:00  0.00%  0.00% craftd
 4503 root      10   0  1040K   492K nanslp   0:00  0.00%  0.00% smartd
```

```
    6 root      28   0     0K     0K sleep    0:00  0.00%  0.00% netdaemon
 4498 root       2   0  1736K   932K select   0:00  0.00%  0.00% nasd
 4506 root       2   0  1348K   672K select   0:00  0.00%  0.00% rtspd
 4489 root       2   0  1160K   668K select   0:00  0.00%  0.00% inetd
 4478 root       2   0  1108K   608K select   0:00  0.00%  0.00% tnetd
 4483 root       2   0  1296K   540K select   0:00  0.00%  0.00% ntpd
 4514 root       3   0  1080K   540K ttyin    0:00  0.00%  0.00% getty
 4331 root       2   0   416K   232K select   0:00  0.00%  0.00% pccardd
    7 root       2   0     0K     0K pfeacc   0:00  0.00%  0.00% if_pfe_listen
   11 root       2   0     0K     0K picacc   0:00  0.00%  0.00% if_pic_listen
    3 root      18   0     0K     0K psleep   0:00  0.00%  0.00% vmdaemon
    9 root       2   0     0K     0K scs_ho   0:00  0.00%  0.00% scs_housekeepi
   10 root       2   0     0K     0K cb-pol   0:00  0.00%  0.00% cb_poll
```

Meaning

The sample output shows the amount of virtual memory used by the Routing Engine and software processes. For example, 118 MB of physical memory is free and 512 MB of the swap file is free, indicating that the router is not short of memory. The processes field shows that most of the 58 processes are in the sleeping state, with 1 in the running state. The process or command that is running is the top command.

The commands column lists the processes that are currently running. For example, the chassis process (chassisd) has a process identifier (PID) of 4480, with a current priority (PRI) of 2. A lower priority number indicates a higher priority.

The processes are listed according to level of activity, with the most active process at the top of the output. For example, the chassis (chassisd) process is consuming the largest amount of CPU resource at 2.34 percent.

The memory field (Mem) shows the virtual memory managed by the Routing Engine and used by processes. The value in the memory field is in KB and MB, and is broken down as follows:

- Active—Memory that is allocated and actually in use by programs.
- Inact—Memory that is either allocated but not recently used or memory that was freed by programs. Inactive memory is still mapped in the address space of one or more processes and, therefore, counts toward the resident set size of those processes.
- Wired—Memory that is not eligible to be swapped, and is usually used for Routing Engine memory structures or memory physically locked by a process.
- Cache—Memory that is not associated with any program and does not need to be swapped before being reused.

- Buf—The size of the memory buffer used to hold data recently called from disk.
- Free—Memory that is not associated with any programs. Memory freed by a process can become Inactive, Cache, or Free, depending on the method used by the process to free the memory.

When the system is under memory pressure, the pageout process reuses memory from the free, cache, inactive and, if necessary, active pages.

The Swap field shows the total swap space available and how much is unused. In the example, the output shows 512 MB of total swap space and 512 MB of free swap space.

Finally, the memory usage of each process is listed. The SIZE field indicates the size of the virtual address space, and the RES field indicates the amount of the program in physical memory, which is also known as RSS or Resident Set Size. In the sample output, the chassis (chassisd) process has 3728 KB of virtual address space and 1908 KB of physical memory.

Initial Routing Engine Configuration Example

You can use configuration groups to ensure that the correct IP addresses are used for each Routing Engine and to maintain a single configuration file for both Routing Engines.

The following example defines configuration groups **re0** and **re1** with separate IP addresses. These well-known configuration group names take effect only on the appropriate Routing Engine.

```
groups {
    re0 {
        system {
            host-name my-re0;
        }
        interfaces {
            fxp0 {
                description "10/100 Management interface";
                unit 0 {
                    family inet {
                        address 10.255.2.40/24;
                    }
                }
            }
        }
    }
    re1 {
        system {
```

```
            host-name my-re1;
        }
        interfaces {
            fxp0 {
                description "10/100 Management interface";
                unit 0 {
                    family inet {
                        address 10.255.2.41/24;
                    }
                }
            }
        }
    }
}
```

You can assign an additional IP address to the management Ethernet interface (**fxp0** in this example) on both Routing Engines. The assigned address uses the **master-only** keyword and is identical for both Routing Engines, ensuring that the IP address for the primary Routing Engine can be accessed at any time. The address is active only on the primary Routing Engine's management Ethernet interface. During a Routing Engine switchover, the address moves over to the new primary Routing Engine.

For example, on **re0**, the configuration is:

```
[edit groups re0 interfaces fxp0]

unit 0 {
    family inet {
        address 10.17.40.131/25 {
            master-only;
        }
        address 10.17.40.132/25;
    }
}
```

On **re1**, the configuration is:

```
[edit groups re1 interfaces fxp0]
unit 0 {
    family inet {
        address 10.17.40.131/25 {
            master-only;
```

```
        }
        address 10.17.40.133/25;
    }
}
```

For more information about the initial configuration of dual Routing Engines, see the Junos OS Software Installation and Upgrade Guide. For more information about assigning an additional IP address to the management Ethernet interface with the **master-only** keyword on both Routing Engines, see the Junos OS CLI User Guide.

Copying a Configuration File from One Routing Engine to the Other

You can use either the console port or the management Ethernet port to establish connectivity between the two Routing Engines. You can then copy or use FTP to transfer the configuration from the primary to the backup, and load the file and commit it in the normal way.

To connect to the other Routing Engine using the management Ethernet port, issue the following command:

```
user@host> request routing-engine login (other-routing-engine | re0 | re1)
```

On a TX Matrix router, to make connections to the other Routing Engine using the management Ethernet port, issue the following command:

```
user@host> request routing-engine login (backup | lcc number | master | other-routing-engine |
re0 | re1)
```

For more information about the `request routing-engine login` command, see the CLI Explorer.

To copy a configuration file from one Routing Engine to the other, issue the `file copy` command:

```
user@host> file copy source destination
```

In this case, ***source*** is the name of the configuration file. These files are stored in the directory **/config**. The active configuration is **/config/juniper.conf**, and older configurations are in **/config/juniper.conf {1...9}**. The ***destination*** is a file on the other Routing Engine.

The following example copies a configuration file from Routing Engine 0 to Routing Engine 1:

```
user@host> file copy /config/juniper.conf re1:/var/tmp/copied-juniper.conf
```

The following example copies a configuration file from Routing Engine 0 to Routing Engine 1 on a TX Matrix router:

```
user@host> file copy /config/juniper.conf scc-re1:/var/tmp/copied-juniper.conf
```

To load the configuration file, enter the `load replace` command at the `[edit]` hierarchy level:

```
user@host> load replace /var/tmp/copied-juniper.conf
```

CAUTION: Make sure you change any IP addresses specified in the management Ethernet interface configuration on Routing Engine 0 to addresses appropriate for Routing Engine 1.

Loading a Software Package from the Other Routing Engine

You can load a package from the other Routing Engine onto the local Routing Engine using the existing `request system software add` *package-name* command:

```
user@host> request system software add re(0|1):/filename
```

In the **re** portion of the URL, specify the number of the other Routing Engine. In the ***filename*** portion of the URL, specify the path to the package. Packages are typically in the directory **/var/sw/pkg**.

RELATED DOCUMENTATION

Understanding Routing Engine Redundancy | 128

Understanding Switching Control Board Redundancy | 15

Platform Redundancy FEB Redundancy Support for High Availability of ACX7509 Devices

IN THIS SECTION

FEB redundancy is supported on the ACX7509 device. The Routing Control Board (RCB) and Forwarding Engine Board (FEB) mastership is tied and switchover together. The master Routing Engine (RE) software manages both FEBs. Graceful RE switchover (GRES) for RCB and FEB simultaneous switchover is supported with support for replication and restoration of statistics.

The graceful Routing Engine switchover (GRES) feature in Junos OS and Junos OS Evolved enables a router with redundant Routing Engines to continue forwarding packets, even if one Routing Engine fails. GRES preserves interface and kernel information. Traffic is not interrupted. However, GRES does not preserve the control plane.

When GRES mode is not enabled on the Junos OS, or Junos OS Evolved, it is considered as Non-GRES mode.

Routing Engine Switchover Conditions and Prerequisites

The prerequisite for a GRES is that, the backup RCB and FEB are online for 360 seconds. The back to back switchover time is more than 360 seconds.

The switchover conditions are as follows:

- Master RCB power-failure.
- Master RE rebooted,

- Master RE plugged out,
- Master RE offlined.
- Linux kernel crash on master RE.
- Critical application failure on master RE (including PFE management, PP management & packet input output applications).
- Power fault on master FEB.

When the FEB redundancy configuration is active, the supported RCB-FEB configurations for ACX7509 are as follows:

Table 5: Supported Redundancy Modes in ACX7509

Supported Redundancy Modes	Condition
Both RCBs and FEBs present	Fully redundant system.
RCB0/FEB0 present and RCB1/FEB1 not present	Non-redundant system.
RCB1/FEB1 present and RCB0/FEB0 not present	Non-redundant system.

The unsupported redundancy modes for ACX7509 with related alarms are as follows:

Table 6: Unsupported Redundancy Modes in ACX7509

Unsupported Redundancy Modes	Condition
RCB0/FEB1 present	The unmatched FEB1 does not come online.
RCB1/FEB0 present	The unmatched FEB0 does not come online.
Two RCBs with FEB0	Both RCBs and the FEB come online. RCB0/FEB0 becomes master. CLI based switchover is allowed. Fault-triggered automatic switchover does not happen.

Support for Replication and Restoration of Statistics on RE Switchover (ACX7509)

The RCB and FEB mastership is tied and switchover together. The master RE software manages both FEBs.

Traffic Flow and Switchover

The WAN traffic entering the system is forwarded to and flows through both FEBs. The FEBs route the traffic internally to the egress FPC. The FPC transmits traffic from master FEB to WAN and drops the traffic from the backup FEB. The backup FEB is in hot standby and ready to forward traffic.

Junos software supports the replication and restoration of the following statistics for traffic flow and switchover:

- Interface Statistics.
- Sub-interface Statistics (Without Queue Statistics).
- The replication for other PFE statistics entities, such as:
  - Interface per Queue Statistics
  - Firewall Statistics
  - uRPF Statistics
  - DDoS PFE Statistics (excluding the RE level DDoS counters)

Limitations for GRES mode

- RCB and FEB are a host-subsystem and switchover due to one will cause complete host sub-system switch. Hence, the independent switchover of RCB and FEB is not supported.
- For GRES, if there is sudden master PFE loss, the interface could go down briefly at the end router. The link comes back up immediately when the switchover takes effect.
- The `show interfaces` command displays the restoration of the RE based statistics.
- Ungraceful removal of FEB is not supported.

Traffic Management

When the WAN traffic is affected due to various fault scenarios, the traffic hit duration depends on:

- The time taken to detect the fault condition.
- The time taken for RCB-FEB switchover.

The RE manages the traffic switchover.

RELATED INFORMATION

5

PART

Configuring Load Balancing

Load Balancing on Aggregated Ethernet Interfaces

SUMMARY

Load balancing on aggregated ethernet interfaces reduces network congestion by dividing traffic among multiple interfaces.

IN THIS SECTION

When you bundle several physical aggregated Ethernet Interfaces to form a single logical interface, it is called link aggregation. Link aggregation increases bandwidth, provides graceful degradation as failure occurs, increases availability and provides load-balancing capabilities. Load balancing enables the device to divide incoming and outgoing traffic along multiple interfaces to reduce congestion in the network. This topic describes load balancing and how to configure load balancing on your device.

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

Review the "Platform-Specific Aggregated Ethernet Load Balancing Behavior" on page 192 section for notes related to your platform.

Load Balancing and Ethernet Link Aggregation Overview

You can create a LAG for a group of Ethernet ports. L2 bridging traffic is load balanced across the member links of this group, making the configuration attractive for congestion concerns as well as for redundancy. Each LAG bundle contains up to 16 links. Platform support depends on the Junos OS release in your installation.

For LAG bundles, the hashing algorithm determines how traffic entering a LAG bundle is placed onto the bundle's member links. The hashing algorithm tries to manage bandwidth by evenly load-balancing all incoming traffic across the member links in the bundle. The hash-mode of the hashing algorithm is set to L2 payload by default. When the hash-mode is set to L2 payload, the hashing algorithm uses the IPv4 and IPv6 payload fields for hashing. You can also configure the load balancing hash key for L2 traffic to use fields in the L3 and Layer 4 headers using the `payload` statement. However, note that the load-balancing behavior is platform-specific and based on appropriate hash-key configurations.

For more information, see *Configuring Load Balancing on a LAG Link*. In an L2 switch, one link is overutilized and other links are underutilized.

Understanding Aggregated Ethernet Load Balancing

The link aggregation feature is used to bundle several physical aggregated Ethernet interfaces to form one logical interface. One or more links are aggregated to form a virtual link or link aggregation group (LAG). The MAC client treats this virtual link as if it were a single link. Link aggregation increases bandwidth, provides graceful degradation as failure occurs, and increases availability.

In addition to these benefits, an aggregated Ethernet bundle is enhanced to provide load-balancing capabilities that ensure that the link utilization among the member links of the aggregated Ethernet bundle are fully and efficiently utilized.

The load-balancing feature allows a device to divide incoming and outgoing traffic along multiple paths or interfaces in order to reduce congestion in the network. Load balancing improves the utilization of various network paths and provides more effective network bandwidth.

Typically, the applications that use load balancing include:

- Aggregated Interfaces (Layer 2)

Aggregated Interfaces (also called AE for aggregated Ethernet, and AS for aggregated SONET) are a Layer 2 mechanism for load-balancing across multiple interfaces between two devices. Because this is a Layer 2 load-balancing mechanism, all of the individual component links must be between the same two devices on each end. Junos OS supports a non-signaled (static) configuration for Ethernet and SONET, as well as the 802.3ad standardized LACP protocol for negotiation over Ethernet links.

- Equal-Cost Multipath (ECMP) (Layer 3)

  By default, when there are multiple equal-cost paths to the same destination for the active route, Junos OS uses a hash algorithm to choose one of the next-hop addresses to install in the forwarding table. Whenever the set of next hops for a destination changes in any way, the next-hop address is rechosen using the hash algorithm. There is also an option that allows multiple next-hop addresses to be installed in the forwarding table, known as per-packet load balancing.

  ECMP load balancing can be:

  - Across BGP paths (BGP multipath)
  - Within a BGP path, across multiple LSPs

In complex Ethernet topologies, traffic imbalances occur due to increased traffic flow, and load balancing becomes challenging for some of the following reasons:

- Incorrect load balancing by aggregate next hops
- Incorrect packet hash computation
- Insufficient variance in the packet flow
- Incorrect pattern selection

As a result of traffic imbalance, the load is not well distributed causing congestion in certain links, whereas some other links are not efficiently utilized.

To overcome these challenges, Junos OS provides the following solutions for resolving the genuine traffic imbalance on aggregated Ethernet bundles (IEEE 802.3ad).

- Adaptive Load Balancing

  Adaptive load balancing uses a feedback mechanism to correct a genuine traffic imbalance. To correct the imbalance weights, the bandwidth and packet stream of links are adapted to achieve efficient traffic distribution across the links in an AE bundle.

  To configure adaptive load balancing, include the `adaptive` statement at the `[edit interfaces ae`*x*` aggregated-ether-options load-balance]` hierarchy level.

  To configure the tolerance value as a percentage, include the `tolerance` optional keyword at the `[edit interfaces ae`*x*` aggregated-ether-options load-balance adaptive]` hierarchy level.

To configure adaptive load balancing based on packets per second (instead of the default bits per second setting), include the `pps` optional keyword at the `[edit interfaces aex aggregated-ether-options load-balance adaptive]` hierarchy level.

To configure the scan interval for the hash value based on the sample rate for the last two seconds, include the `scan-interval` optional keyword at the `[edit interfaces aex aggregated-ether-options load-balance adaptive]` hierarchy level.

- Per-Packet Random Spray Load Balancing

  When the adaptive load-balancing option fails, per-packet random spray load balancing serves as a last resort. It ensures that the members of an AE bundle are equally loaded without taking bandwidth into consideration. Per packet causes packet reordering and hence is recommended only if the applications absorb reordering. Per-packet random spray eliminates traffic imbalance that occurs as a result of software errors, except for packet hash.

  To configure per-packet random spray load balancing, include the `per-packet` statement at the `[edit interfaces aex aggregated-ether-options load-balance]` hierarchy level.

The aggregated Ethernet load-balancing solutions are mutually exclusive. When more than one of the load-balancing solutions is configured, the solution that is configured last overrides the previously configured one. You can verify the load-balancing solution being used by issuing the `show interfaces aex aggregated-ether-options load-balance` command.

SEE ALSO

show interfaces (Aggregated Ethernet)

Stateful Load Balancing for Aggregated Ethernet Interfaces Using 5-Tuple Data

IN THIS SECTION

When multiple flows are transmitted out of an aggregated Ethernet (ae) interface, the flows must be distributed across the different member links evenly to enable an effective and optimal load-balancing

behavior. To obtain a streamlined and robust method of load-balancing, the member link of the aggregated Ethernet interface bundle that is selected each time for load balancing plays a significant part. The balanced mode of link selection uses 'n' bits in a precomputed hash value if it needs to select one of 2^n (2 raised to the power of n) next-hop in the unilist. The unbalanced mode of member-link or next-hop selection uses 8 bits in a precomputed hash to select an entry in a selector table, which is randomly done with the member link IDs of the link aggregation group (LAG) or aebundle.

The term balanced versus unbalanced indicates whether a selector table is used for load balancing mechanism or not. The LAG bundle uses the unbalanced mode (selector table balancing) to balance the traffic across member links. When the traffic flows are minimal, the following problems might occur with the unbalanced mode: The link selection logic utilizes only subset bits of the precomputed hash. Regardless of the efficiency of the hashing algorithm, it is only the compressed representation of a flow. Because the inter-flow variance is very low, the resultant hashes and the subset that are computed do not provide the necessary variability to effectively utilize all the LAG member links. An excessive amount of random nature exists in the hash computation and also in the selector table. As a result, the deviation from being an optimal load-balancing technique for each child link that is selected is higher when the number of flows is lower.

The deviation per child link is defined as

Vi = ((Ci - (M/N)))/N

where

- Vi denotes the deviation for that child link ’i’.
- i denotes the child link member/index.
- Ci represents the packets transmitted for that child link 'i'.
- M signifies the total packets transmitted on that LAG bundle.
- N denotes the number of child links in that LAG.

Because of these drawbacks, for smaller number of flows, or flows with less inter-flow variance, the link utilization is skewed, and a high probability of a few child links not being utilized entirely exists.

The mechanism to record and retain states for the flows and distribute the traffic load accordingly is added. As a result, for m number of flows, they are distributed among n member links of a LAG bundle or among the unilist of next-hops in an ECMP link. This method of splitting the load among member links is called *stateful load balancing* and it uses 5-tuple information (source and destination addresses, protocol, source and destination ports). Such a method can be mapped directly to the flows, or to a precompute hash based on certain fields in the flow. As a result, the deviation observed on each child link is reduced.

This mechanism works efficiently only for minimal number of flows (less than thousands of flows, approximately). For a larger number of flows (between 1000 and 10,000 flows), we recommend that distributed Trio-based load-balancing mechanism is used.

Consider a sample scenario in which 'n' links in the LAG are identified with link IDs of 0 through n-1. A hash table or a flow table is used to record the flows as and when they show up. The hashing key is constructed using the fields that uniquely identify a flow. The result of the lookup identifies the link_id that the flow is currently using. For each packet, the flow table based on the flow identifier is examined. If a match is found, it denotes a packet that belongs to a flow that is previously processed or detected. The link ID is associated with the flow. If a match is not found, it is the first packet that belongs to the flow. The link ID is used to select the link and the flow is inserted into the flow table.

To enable per-flow load balancing based on hash values, include the `per-flow` statement at the at the `[edit interfaces ae`*`X`*` unit `*`logical-unit-number`*` forwarding-options load-balance-stateful]` hierarchy level. By default, Junos OS uses a hashing method based only on the destination address to elect a forwarding next hop when multiple equal-cost paths are available. All Packet Forwarding Engine slots are assigned the same hash value by default. To configure the load-balancing algorithm to dynamically rebalance the LAG using existing parameters, include the `rebalance` *`interval`* statement at the `[edit interfaces ae`*`X`*` unit `*`logical-unit-number`*` forwarding-options load-balance-stateful]` hierarchy level. This parameter periodically load balances traffic by providing a synchronized rebalance switchover across all the ingress Packet Forwarding Engines (PFEs) over a rebalance interval. You can specify the interval as a value in the range of 1 through 1000 flows per minute. To configure the load type, include the `load-type (low | medium | high)` statement at the `[edit interfaces ae`*`X`*` unit `*`logical-unit-number`*` forwarding-options load-balance-stateful]` hierarchy level.

The `stateful per-flow` option enables the load-balancing capability on AE bundles. The `rebalance` option clears the load balance state at specified intervals. The `load` option informs the Packet Forwarding Engine regarding the appropriate memory pattern to be used. If the number of flows that flow on this aggregated Ethernet interface is less (between 1 and 100 flows), then the `low` keyword can be used. Similarly for relatively higher flows (between 100 and 1000 flows), the `medium` keyword can be used and the `large` keyword can be used for the maximum flows (between 1000 and 10,000 flows). The approximate number of flows for effective load-balancing for each keyword is a derivative.

The `clear interfaces ae`*`X`*` unit `*`logical-unit-number`*` forwarding-options load-balance state` command clears the load balance state at the hardware level and enables rebalancing from the cleaned up, empty state. This clear state is triggered only when you use this command. The `clear interfaces aggregate forwarding-options load-balance state` command clears all the aggregate Ethernet interface load balancing states and re-creates them newly.

Guidelines for Configuring Stateful Load Balancing for Aggegated Ethernet Interfaces or LAG Bundles

Keep the following points in mind while configuring stateful load-balancing for aggregated Ethernet interfaces:

- When a child link is removed or added, a new aggregate selector is selected and traffic flows onto the new selector. Because the selector is empty, flows are filled in the selector. This behavior causes redistribution of flows because the old state is lost. This is the existing behavior without enabling stateful per-flow load-balancing.

- Stateful per-flow load-balancing functions on AE interfaces if the incoming traffic reaches the MPC1E, MPC2E, MPC3E-3D, MPC5E, and MPC6E line cards. Any other type of line card does not rigger this functionality. Appropriate CLI errors are displayed if the MPCs do not support this capability.

  With the ingress line card as MPC and the egress line card as MPC or DPC, this feature works properly. Stateful load-balancing is not supported if the ingress line card is a DPC and the egress line card is a DPC or an MPC.

- This capability is not supported for multicast traffic (native/flood).

- Enabling the rebalance option or clearing the load balance state can cause packet reordering for active flows because different sets of links can be selected for traffic flows.

- Although the feature performance is high, it consumes significant amount of line card memory. Approximately, 4000 logical interfaces or 16 aggregated Ethernet logical interfaces can have this feature enabled on supported MPCs. However, when the Packet Forwarding Engine hardware memory is low, depending upon the available memory, it falls back to the default load balancing mechanism. A system logging message is generated in such a situation and sent to the Routing Engine. A restriction on the number of AE interfaces that support stateful load-balancing does not exist; the limit is determined by the line cards.

- If the traffic flows become aged frequently, then the device needs to remove or refresh the load balancing states. As a result, you must configure rebalancing or run the clear command at periodic intervals for proper load-balancing. Otherwise, traffic skewing can occur. When a child link goes down or comes up, the load balancing behavior does not undergo changes on existing flows. This condition is to avoid packet reordering. New flows pick up the child link that come up. If you observe load distribution to be not very effective, you can clear the load-balancing states or use rebalancing functionality to cause an automatic clearance of the hardware states. When you configure the rebalancing facility, traffic flows can get redirected to different links, which can cause packet reordering.

Configuring Stateful Load Balancing on Aggregated Ethernet Interfaces

The mechanism to record and retain states for the flows and distribute the traffic load accordingly is added. As a result, for m number of flows, they are distributed among n member links of a LAG bundle or among the unilist of next-hops in an ECMP link. This method of splitting the load among member links is called *stateful load balancing* and it uses 5-tuple information (source and destination addresses,

protocol, source and destination ports). Such a method can be mapped directly to the flows, or to a precompute hash based on certain fields in the flow. As a result, the deviation observed on each child link is reduced.

To configure stateful load balancing on ae interface bundles:

1. Specify that you want to configure an aggregated Ethernet interface.

```
[edit]
user@R2# set interfaces aeX unit logical-unit-number
```

2. Specify that you want to configure stateful load-balancing.

```
[edit interfaces aeX unit logical-unit-number]
user@R2# edit forwarding-options load-balance-stateful
```

3. Enable the mechanism to perform an even, effective distribution of traffic flows across member links of an aggregated Ethernet interface (ae) bundle on MX Series routers with MPCs, except MPC3Es and MPC4Es.

```
[edit interfaces aeX unit logical-unit-number load-balance-stateful]
user@R2# set per-flow
```

4. Configure periodic rebalancing of traffic flows of an aggregated Ethernet bundle by clearing the load balance state at a specified interval.

```
[edit interfaces aeX unit logical-unit-number load-balance-stateful]
user@R2# set rebalance interval
```

5. Define the load-balancing type to inform the Packet Forwarding Engine regarding the appropriate memory pattern to be used for traffic flows. The approximate number of flows for effective load-balancing for each keyword is a derivative.

```
[edit interfaces aeX unit logical-unit-number load-balance-stateful]
user@R2# set load-type (low | medium | large)
```

6. Configure the address family and IP address for the ae interface.

```
[edit interfaces aeX unit logical-unit-number]]
user@R2# set family family-name address address
```

Configuring Adaptive Load Balancing

This topic describes how to configure adaptive load balancing. Adaptive load balancing maintains efficient utilization of member link bandwidth for an aggregated Ethernet (AE) bundle. Adaptive load balancing uses a feedback mechanism to correct traffic load imbalance by adjusting the bandwidth and packet streams on links within an AE bundle.

Before you begin:

- Configure a set of interfaces with a protocol family and IP address. These interfaces can make up the membership for the AE bundle.
- Create an AE bundle by configuring a set of router interfaces as aggregated Ethernet and with a specific AE group identifier.

To configure adaptive load balancing for an AE bundles:

1. Enable adaptive load balancing on the AE bundle:

```
[edit interfaces ae-x aggregated-ether-options load-balance]
user@router# set adaptive
```

2. Configure the scan interval value for adaptive load balancing on the AE bundle. The scan interval value determines the length of the traffic scan by multiplying the integer value with a 30-second time period:

```
[edit interfaces ae-x aggregated-ether-options load-balance adaptive]
user@router# set scan-interval multiplier
```

3. Configure the tolerance percentage value. The tolerance value determines the allowed deviation in the traffic rates among the members of the AE bundle before the router triggers an adaptive load balancing update:

```
[edit interfaces ae-x aggregated-ether-options load-balance adaptive]
user@router# set tolerance percentage
```

4. (Optional) Enable packet-per-second-based adaptive load balancing on the AE bundle:

```
[edit interfaces ae-x aggregated-ether-options load-balance adaptive]
user@router# set pps
```

SEE ALSO

adaptive

Understanding Symmetric Hashing for Load Balancing

IN THIS SECTION

On devices that support this feature, symmetric hashing keeps the forward and reverse directions of a flow on the same path for features that depend on traffic symmetry. Symmetric hashing helps services maintain session context while load balancing across ECMP next hops and LAG members.

Benefits of Symmetric Hashing

- Maintains bidirectional path symmetry so stateful services see both directions of a flow on the same path.
- Improves load-balancing predictability by using the same normalized inputs for both directions of a flow.
- Reduces asymmetric routing issues that can disrupt service processing or session state.

To configure symmetric hashing, use the `symmetric-hash` configuration statement at the `[edit forwarding-options enhanced-hash-key]` hierarchy. Use the `enhanced-hash-key` statement to configure the packet fields included in hash calculations. By selecting stable, bidirectional fields and excluding unidirectional inputs, you normalize the inputs so the device derives the same hash for both directions of a flow.

You can exclude unidirectional attributes, such as the ingress interface, from the hash with `no-incoming-port` configuration statement at the `[edit forwarding-options enhanced-hash-key]` hierarchy.

Verify the current hash inputs and symmetric hashing status with the `show forwarding-options enhanced-hash-key` command.

Additional considerations:

- Dynamic load balancing (DLB) takes precedence. Symmetric hashing applies to static load balancing; if DLB is enabled, DLB overrides symmetric behavior.
- Use Layer 3 and Layer 4 fields for symmetric hashing. Layer 2 header fields are not included in symmetric hashing and you should use the `hash-mode` configuration statement to use the Layer 2 data field when required.
- Hash polarization can occur in cascaded topologies if all devices use identical hash seeds and inputs. Consider resilient hashing and seed diversity to reduce polarization when links flap or members change.

Configuring Symmetrical Load Balancing on an 802.3ad Link Aggregation Group on MX Series Routers

IN THIS SECTION

Symmetrical Load Balancing on an 802.3ad LAG on MX Series Routers Overview

MX Series routers with Aggregated Ethernet PICs support symmetrical load balancing on an 802.3ad LAG. This feature is significant when two MX Series routers are connected transparently through deep

packet inspection (DPI) devices over an LAG bundle. DPI devices keep track of flows and require information of a given flow in both forward and reverse directions. Without symmetrical load balancing on an 802.3ad LAG, the DPIs could misunderstand the flow, leading to traffic disruptions. By using this feature, a given flow of traffic (duplex) is ensured for the same devices in both directions.

Symmetrical load balancing on an 802.3ad LAG utilizes a mechanism of interchanging the source and destination addresses for a hash computation of fields, such as source address and destination address. The result of a hash computed on these fields is used to choose the link of the LAG. The hash-computation for the forward and reverse flow must be identical. This is achieved by swapping source fields with destination fields for the reverse flow. The swapped operation is referred to as *complement hash computation* or `symmetric-hash complement` and the regular (or unswapped) operation as *symmetric-hash computation* or `symmetric-hash`. The swappable fields are MAC address, IP address, and port.

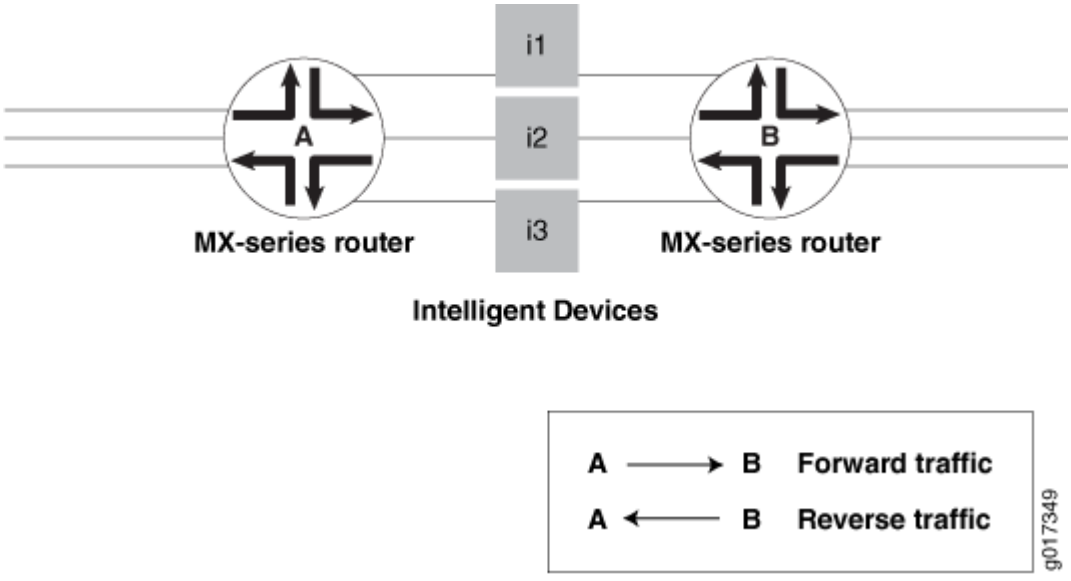
Configuring Symmetric Load Balancing on an 802.3ad LAG on MX Series Routers

You can specify whether symmetric hash or complement hash is done for load-balancing traffic. To configure symmetric hash, use the `symmetric-hash` statement at the `[edit forwarding-options hash-key family inet]` hierarchy level. To configure symmetric hash complement, use the `symmetric-hash complement` statement and option at the `[edit forwarding-options hash-key family inet]` hierarchy level.

These operations can also be performed at the PIC level by specifying a *hash key*. To configure a hash key at the PIC level, use the `symmetric-hash` or `symmetric-hash complement` statement at the `[edit chassis hash-key family inet]` and `[edit chassis hash-key family multiservice]` hierarchy levels.

Consider the example in Figure 6 on page 162.

Figure 6: Symmetric Load Balancing on an 802.3ad LAG on MX Series Routers



Router A is configured with symmetric hash and Router B is configured with symmetric hash complement. Thus, for a given flow *fx*, post hash computation is from Router A to Router B through i2. The reverse traffic for the same flow *fx* is from Router B to Router A through the same i2 device as its hashing (done after swapping source and destination fields) and returns the same link index; since it is performed on the interchanged source and destination addresses.

However, the link chosen may or may not correspond to what was attached to the DPI. In other words, the hashing result should point to the same links that are connected, so that the traffic flows through the same DPI devices in both directions. To make sure this happens, you need to also configure the counterpart ports (ports that are connected to same DPI-iN) with the identical link index. This is done when configuring a child-link into the LAG bundle. This ensures that the link chosen for a given hash result is always the same on either router.

Note that any two links connected to each other should have the same link index and these link indices must be unique in a given bundle.

NOTE: The following restrictions apply when configuring symmetric load balancing on an 802.3ad LAG on MX Series routers:

- The Packet Forwarding Engine (PFE) can be configured to hash the traffic in either symmetric or complement mode. A single PFE complex cannot work simultaneously in both operational modes and such a configuration can yield undesirable results.
- The per-PFE setting overrides the chassis-wide setting only for the family configured. For the other families, the PFE complex still inherits the chassis-wide setting (when configured) or the default setting.
- This feature supports VPLS, INET, and bridged traffic only.
- This feature cannot work in tandem with the `per-flow-hash-seed load-balancing` option. It requires that all the PFE complexes configured in complementary fashion share the same seed. A change in the seed between two counterpart PFE complexes may yield undesired results.

For additional information, see the Junos OS VPNs Library for Routing Devices and the Junos OS Administration Library for Routing Devices.

Example Configuration Statements

To configure 802.3ad LAG parameters at the bundle level:

```
[edit interfaces]
g(x)e-fpc/pic/port {
    gigether-options {
        802.3ad {
```

```
            bundle;
            link-index number;
        }
    }
}
```

where the `link-index number` ranges from 0 through 15.

You can check the link index configured above using the `show interfaces` command:

```
[edit forwarding-options hash-key]
family inet {
    layer-3;
    layer-4;
    symmetric-hash {
        [complement;]
    }
}
family multiservice {
    source-mac;
    destination-mac;
    payload {
    ip {
        layer-3 {
            source-ip-only | destination-ip-only;
        }
        layer-4;
    }
    }
    symmetric-hash {
        [complement;]
    }
}
```

For load-balancing Layer 2 traffic based on Layer 3 fields, you can configure 802.3ad LAG parameters at a per PIC level. These configuration options are available under the chassis hierarchy as follows:

```
[edit chassis]
fpc X {
    pic Y {
        .
        .
```

```
        .
        hash-key {
            family inet {
                layer-3;
                layer-4;
                symmetric-hash {
                    [complement;]
                }
            }
            family multiservice {
                source-mac;
                destination-mac;
                payload {
                    ip {
                        layer-3 {
                            source-ip-only | destination-ip-only;
                        }
                        layer-4;
                    }
                }
                symmetric-hash {
                    [complement;]
                }
            }
        }
        .
        .
        .
    }
}
```

Configuring Symmetrical Load Balancing on Trio-Based MPCs

With some configuration differences, symmetrical load-balancing over an 802.3ad link aggregation group is supported on MX Series routers with Trio-based MPCs.

To achieve symmetrical load-balancing on Trio-Based MPCs, the following needs to be done:

- Compute a Symmetrical Hash

  Both routers must compute the same hash value from the flow in the forward and reverse directions. On Trio-based platforms, the calculated hash value is independent of the direction of the flow, and hence is always symmetric in nature. For this reason, no specific configuration is needed to compute a symmetric hash value on Trio-based platforms.

However, it should be noted that the fields used to configure the hash should have identical include and exclude settings on both ends of the LAG.

- Configure Link Indexes

  To allow both routers to choose the same link using the same hash value, the links within the LAG must be configured with the same link index on both routers. This can be achieved with the `link-index` statement.

- Enable Symmetric Load Balancing

  To configure symmetric load balancing on Trio-based MPCs, include the `symmetric` statement at the `[edit forwarding-options enhanced-hash-key]` hierarchy level. This statement is applicable to Trio-based platforms only.

  The `symmetric` statement can be used with any protocol family and enables symmetric load-balancing for all aggregated Ethernet bundles on the router. The statement needs to be enabled at both ends of the LAG. This statement is disabled by default.

- Achieve Symmetry for Bridged and Routed Traffic

  In some deployments, the LAG bundle on which symmetry is desired is traversed by Layer 2 bridged traffic in the upstream direction and by IPv4 routed traffic in the downstream direction. In such cases, the computed hash is different in each direction because the Ethernet MAC addresses are taken into account for bridged packets. To overcome this, you can exclude source and destination MAC addresses from the enhanced-hash-key computation.

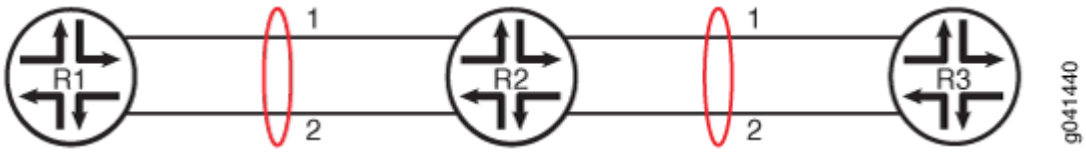
  To exclude source and destination MAC addresses from the enhanced-hash-key computation, include the `no-mac-addresses` statement at the `[edit forwarding-options enhanced-hash-key family multiservice]` hierarchy level. This statement is disabled by default.

When symmetrical load balancing is enabled on Trio-based MPCs, keep in mind the following caveats:

- Traffic polarization is a phenomenon that occurs when using topologies that distribute traffic by using hashing of the same type. When routers are cascaded, traffic polarization can occur, and this can lead to unequal traffic distribution.

  Traffic polarization occurs when LAGs are configured on cascaded routers. For example, in Figure 7 on page 167, if a certain flow uses Link 1 of the aggregated Ethernet bundle between Device R1 and Device R2, the flow also uses Link 1 of the aggregated Ethernet bundle between Device R2 and Device R3.

Figure 7: Traffic Polarization on Cascaded Routers When Symmetrical Load Balancing in Enabled on Trio-based MPCs



This is unlike having a random link selection algorithm, where a flow might use Link 1 of the aggregated Ethernet bundle between Device R1 and Device R2, and Link 2 of the aggregated Ethernet bundle between Device R2 and Device R3.

- Symmetric load balancing is not applicable to per-prefix load-balancing where the hash is computed based on the route prefix.
- Symmetric load balancing is not applicable to MPLS or VPLS traffic, because in these scenarios the labels are not the same in both directions.

Example Configurations

IN THIS SECTION

Example Configurations of Chassis Wide Settings

Router A

```
user@host> show configuration forwarding-options hash-key
family multiservice {
    payload {
        ip {
            layer-3;
        }
    }
    symmetric hash;
}
```

Router B

```
user@host> show configuration forwarding-options hash-key
family multiservice {
    payload {
        ip {
            layer-3;
        }
    }
    symmetric-hash {
        complement;
    }
}
```

Example Configurations of Per–Packet-Forwarding-Engine Settings

Router A

```
user@host> show configuration chassis fpc 2 pic 2 hash-key
family multiservice {
    payload {
        ip {
            layer-3;
        }
    }
    symmetric hash;
}
```

Router B

```
user@host> show configuration chassis fpc 2 pic 3 hash-key
family multiservice {
    payload {
        ip {
            layer-3;
        }
    }
    symmetric-hash {
        complement;
```

```
    }
}
```

RELATED DOCUMENTATION

- Junos OS VPNs Library for Routing Devices
- Junos OS Administration Library for Routing Devices

Configuring PIC-Level Symmetrical Hashing for Load Balancing on 802.3ad LAGs for MX Series Routers

Symmetrical hashing for load balancing on an 802.3ad Link Aggregation Group (LAG) is useful when two MX Series routers (for example, Router A and Router B) are connected transparently through Deep Packet Inspection (DPI) devices over a LAG bundle. The DPI devices keep track of traffic flows in both the forward and reverse directions.

If symmetrical hashing is configured, the reverse flow of traffic is also directed through the same child link on the LAG and is bound to flow through the same DPI device. This enables proper accounting on the DPI of the traffic in both the forward and reverse flows.

If symmetrical hashing is not configured, a different child link on the LAG might be chosen for the reverse flow of traffic through a different DPI device. This results in incomplete information about the forward and reverse flows of traffic on the DPI device leading to incomplete accounting of the traffic by the DPI device.

Symmetrical hashing is computed based on fields like source address and destination address. You can configure symmetrical hashing both at the chassis level and the PIC level for load balancing based on Layer 2, Layer 3, and Layer 4 data unit fields for family inet (IPv4 protocol family) and multiservice (switch or bridge) traffic. Symmetrical hashing configured at the chassis level is applicable to the entire router, and is inherited by all its PICs and Packet Forwarding Engines. Configuring PIC-level symmetrical hashing provides you more granularity at the Packet Forwarding Engine level.

For the two routers connected through the DPI devices over a LAG bundle, you can configure **symmetric-hash** on one router and **symmetric-hash complement** on the remote-end router or vice-versa.

To configure symmetrical hashing at the chassis level, include the **symmetric-hash** or the `symmetric-hash complement` statements at the `[edit forwarding-options hash-key family]` hierarchy level. For information about configuring symmetrical hashing at the chassis level and configuring the link index, see the Junos OS Network Interfaces Library for Routing Devices and the Junos OS VPNs Library for Routing Devices.

NOTE: On MX Series DPCs, configuring symmetrical hashing at the PIC level refers to configuring symmetrical hashing at the Packet Forwarding Engine level.

To configure symmetrical hashing at the PIC level on the inbound traffic interface (where traffic enters the router), include the **symmetric-hash** or `symmetric-hash complement` statement at the [**edit chassis fpc** *slot-number* **pic** *pic-number* **hash-key**] hierarchy level:

```
[edit chassis fpc slot-number pic pic-number hash-key]
family multiservice {
    source-mac;
    destination-mac;
    payload {
        ip {
            layer-3 (source-ip-only | destination-ip-only);
            layer-4;
        }
    }
    symmetric-hash {
        complement;
    }
}
```

```
family inet {
    layer-3;
    layer-4;
    symmetric-hash {
        complement;
    }
}
```

NOTE:

- PIC-level symmetrical hashing overrides the chassis-level symmetrical hashing configured at the [**edit chassis forwarding-options hash-key**] hierarchy level.

- Symmetrical hashing for load balancing on 802.3ad Link Aggregation Groups is currently supported for the VPLS, INET and bridged traffic only.
- Hash key configuration on a PIC or Packet Forwarding Engine can be either in the “symmetric hash” or the “symmetric hash complement” mode, but not both at the same time.

SEE ALSO

hash-key

inet

payload

symmetric-hash

Examples: Configuring PIC-Level Symmetrical Hashing for Load Balancing on 802.3ad LAGs on MX Series Routers

IN THIS SECTION

NOTE: These examples are applicable only to the DPCs Supported on MX240, MX480, and MX960 Routers. For the list of DPCs supported, see *DPCs Supported on MX240, MX480, and MX960 Routers* in the Related Documentation section.

The following examples show how to configure symmetrical hashing at the PIC level for load balancing on MX Series routers:

Configuring Symmetrical Hashing for family multiservice on Both Routers

On the inbound traffic interface where traffic enters Router A, include the `symmetric-hash` statement at the `[edit chassis fpc slot-number pic pic-number hash-key family multiservice]` hierarchy level:

```
[edit chassis fpc 2 pic 2 hash-key]
family multiservice {
    source-mac;
    destination-mac;
    payload {
        ip {
            layer-3;
            layer-4;
        }
    }
    symmetric-hash;
}
```

On the inbound traffic interface where traffic enters Router B, include the `symmetric-hash complement` statement at the `[edit chassis fpc slot-number pic pic-number hash-key family multiservice]` hierarchy level:

```
[edit chassis fpc 0 pic 3 hash-key]
family multiservice {
    source-mac;
    destination-mac;
    payload {
        ip {
            layer-3;
            layer-4;
        }
    }
    symmetric-hash {
        complement;
    }
}
```

Configuring Symmetrical Hashing for family inet on Both Routers

On the inbound traffic interface where traffic enters Router A, include the `symmetric-hash` statement at the `[edit chassis fpc slot-number pic pic-number hash-key family inet]` hierarchy level:

```
[edit chassis fpc 0 pic 1 hash-key]
family inet {
    layer-3;
    layer-4;
    symmetric-hash;
}
```

On the inbound traffic interface where traffic enters Router B, include the `symmetric-hash complement` statement at the `[edit chassis fpc slot-number pic pic-number hash-key family inet]` hierarchy level:

```
[edit chassis fpc 1 pic 2 hash-key]
family inet {
    layer-3;
    layer-4;
    symmetric-hash {
        complement;
    }
}
```

Configuring Symmetrical Hashing for family inet and family multiservice on the Two Routers

On the inbound traffic interface where traffic enters Router A, include the `symmetric-hash` statement at the `[edit chassis fpc slot-number pic pic-number hash-key family multiservice]` hierarchy level:

```
[edit chassis fpc 1 pic 0 hash-key]
family multiservice {
    payload {
        ip {
            layer-3;
            layer-4;
        }
    }
```

```
    symmetric-hash;
}
```

On the inbound traffic interface where traffic enters Router B, include the `symmetric-hash complement` statement at the `[edit chassis fpc slot-number pic pic-number hash-key family inet]` hierarchy level:

```
[edit chassis fpc 0 pic 3 hash-key]
family inet {
    layer-3;
    layer-4;
    symmetric-hash {
        complement;
    }
}
```

SEE ALSO

DPCs Supported on MX240, MX480, and MX960 Routers

Example: Configuring Aggregated Ethernet Load Balancing

IN THIS SECTION

Example: Configuring Aggregated Ethernet Load Balancing

IN THIS SECTION

This example shows how to configure aggregated Ethernet load balancing.

Requirements

This example uses the following hardware and software components:

- Three MX Series routers with MIC and MPC interfaces or three PTX Series Packet Transport Routers with PIC and FPC interfaces
- Junos OS Release 13.3 or later running on all devices

Overview

IN THIS SECTION

Load balancing is required on the forwarding plane when there are multiple paths or interfaces available to the next hop router, and it is best if the incoming traffic is load balanced across all available paths for better link utilization.

Aggregated Ethernet bundle is a typical application that uses load balancing to balance traffic flows across the member links of the bundle (IEEE 802.3ad).

Starting with Junos OS Release 13.3, aggregated Ethernet load balancing is enhanced to provide two solutions for resolving genuine traffic imbalance on aggregated Ethernet bundles on MICs or MPCs of MX Series routers. Starting with Junos OS Release 14.1, aggregated Ethernet load balancing is enhanced to provide two solutions for resolving genuine traffic imbalance on aggregated Ethernet bundles on PICs or FPCs of PTX Series Packet Transport Routers.

The aggregated Ethernet load-balancing solutions are:

- Adaptive—Adaptive load balancing is used in scenarios where flow-based hashing is not sufficient to achieve a uniform load distribution. This load-balancing solution implements a real-time feedback and control mechanism to monitor and manage imbalances in network load.

The adaptive load-balancing solution corrects the traffic flow imbalance by modifying the selector entries, and periodically scanning the link utilization on each member link of the AE bundle to detect any deviations. When a deviation is detected, an adjustment event is triggered and fewer flows are mapped to the affected member link. As a result, the offered bandwidth of that member link goes down. This causes a continuous feedback loop, which over a period of time ensures that the same amount of byte rate is offered to all the member links, thus providing efficient traffic distribution across each member link in the AE bundle.

To configure adaptive load balancing, include the `adaptive` statement at the `[edit interfaces ae`*x*` aggregated-ether-options load-balance]` hierarchy level.

NOTE: Enabling adaptive load balancing may cause packet reordering once every rebalance interval.

The `pps` option enables load balancing based on the packets-per-second rate. The default setting is bits-per-second load balancing.

The `scan-interval` value configures the length of time for scanning as a multiple of 30 seconds.

The `tolerance` value is the limit to the variance in the packet traffic flow to the aggregated Ethernet links in the bundle. You can specify a maximum of 100-percent variance. When the tolerance attribute is not configured, a default value of 20 percent is enabled for adaptive load balancing. A smaller tolerance value balances better bandwidth, but takes a longer convergence time.

- Per-packet random spray—When the adaptive load-balancing solution fails, per-packet random spray acts as a last resort. The per-packet random spray load-balancing solution helps to address traffic imbalance by randomly spraying the packets to the aggregate next hops. This ensures that all the member links of the AE bundle are equally loaded, resulting in packet reordering.

  In addition, per-packet random spray identifies the ingress Packet Forwarding Engine that caused the traffic imbalance and eliminates traffic imbalance that occurs as a result of software errors, except for packet hash.

  To configure per-packet random spray load balancing, include the `per-packet` statement at the `[edit interfaces ae`*x*` aggregated-ether-options load-balance]` hierarchy level.

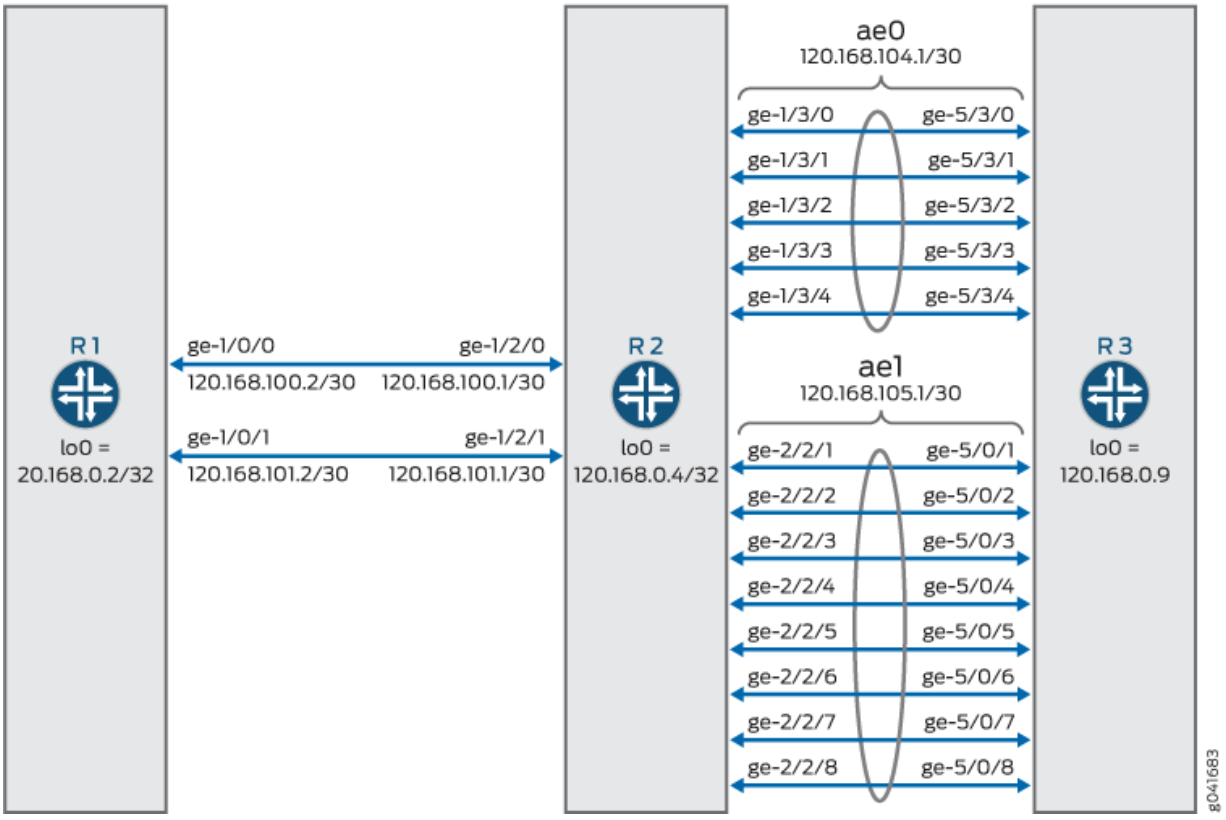
  NOTE: The Per-Packet option for load balancing is not supported on the PTX Series Packet Transport Routers.

The aggregated Ethernet load-balancing solutions are mutually exclusive. When more than one of the load-balancing solutions is configured, the solution that is configured last overrides the previously configured one. You can verify the load-balancing solution being implemented by issuing the `show interfaces ae`*x*` aggregated-ether-options load-balance` command.

Topology

In this topology, two aggregated Ethernet bundles - ae0 and ae1 - are configured on the links between the R2 and R3 routers.

Figure 8: Aggregated Ethernet Load Balancing



Configuration

IN THIS SECTION

CLI Quick Configuration

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

R1

```
set chassis aggregated-devices ethernet device-count 12
set interfaces xe-0/0/0 unit 0 family inet address 120.168.1.1/30
set interfaces xe-0/0/0 unit 0 family iso
set interfaces xe-0/0/0 unit 0 family mpls
set interfaces xe-0/0/1 unit 0 family inet address 120.168.2.1/30
set interfaces xe-0/0/1 unit 0 family iso
set interfaces xe-0/0/1 unit 0 family mpls
set interfaces ge-1/0/0 unit 0 family inet address 120.168.100.2/30
set interfaces ge-1/0/0 unit 0 family iso
set interfaces ge-1/0/0 unit 0 family mpls
set interfaces ge-1/0/1 unit 0 family inet address 120.168.101.2/30
set interfaces ge-1/0/1 unit 0 family iso
set interfaces ge-1/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 120.168.0.2/32
set interfaces lo0 unit 0 family iso address 49.0001.1201.6800.0002.00
set routing-options router-id 120.168.0.2
set routing-options autonomous-system 55
set protocols rsvp interface ge-1/0/0.0
set protocols rsvp interface ge-1/0/1.0
set protocols mpls label-switched-path videl-to-sweets to 120.168.0.9
set protocols mpls label-switched-path v-2-s-601 to 60.0.1.0
set protocols mpls label-switched-path v-2-s-601 primary v-2-s-601-primary hop-limit 5
set protocols mpls label-switched-path v-2-s-602 to 60.0.2.0
set protocols mpls label-switched-path v-2-s-602 primary v-2-s-602-primary hop-limit 5
set protocols mpls label-switched-path v-2-s-603 to 60.0.3.0
set protocols mpls label-switched-path v-2-s-604 to 60.0.4.0
set protocols mpls path v-2-s-601-primary 120.168.100.1 strict
set protocols mpls path v-2-s-601-primary 120.168.104.2 strict
set protocols mpls path v-2-s-602-primary 120.168.101.1 strict
set protocols mpls path v-2-s-602-primary 120.168.105.2 strict
set protocols mpls interface ge-1/0/0.0
set protocols mpls interface ge-1/0/1.0
set protocols bgp group pe-routers type internal
set protocols bgp group pe-routers local-address 120.168.0.2
```

```
set protocols bgp group pe-routers family inet unicast
set protocols bgp group pe-routers family inet-vpn unicast
set protocols bgp group pe-routers neighbor 120.168.0.9
set protocols isis traffic-engineering family inet shortcuts
set protocols isis level 1 disable
set protocols isis interface ge-1/0/0.0
set protocols isis interface ge-1/0/1.0
set protocols isis interface lo0.0
set policy-options policy-statement nhs then next-hop self
set policy-options policy-statement vpn-m5-export term 1 from protocol bgp
set policy-options policy-statement vpn-m5-export term 1 from protocol direct
set policy-options policy-statement vpn-m5-export term 1 then community add vpn-m5-target
set policy-options policy-statement vpn-m5-export term 1 then accept
set policy-options policy-statement vpn-m5-export term 2 then reject
set policy-options policy-statement vpn-m5-import term 1 from protocol bgp
set policy-options policy-statement vpn-m5-import term 1 from community vpn-m5-target
set policy-options policy-statement vpn-m5-import term 1 then accept
set policy-options policy-statement vpn-m5-import term 2 then reject
set policy-options community vpn-m5-target members target:55:100
set routing-instances vpn-m5 instance-type vrf
set routing-instances vpn-m5 interface xe-0/0/0.0
set routing-instances vpn-m5 interface xe-0/0/1.0
set routing-instances vpn-m5 route-distinguisher 120.168.0.2:1
set routing-instances vpn-m5 vrf-import vpn-m5-import
set routing-instances vpn-m5 vrf-export vpn-m5-export
set routing-instances vpn-m5 protocols bgp group ce type external
set routing-instances vpn-m5 protocols bgp group ce peer-as 100
set routing-instances vpn-m5 protocols bgp group ce as-override
set routing-instances vpn-m5 protocols bgp group ce neighbor 120.168.1.2
set routing-instances vpn-m5 protocols bgp group ce neighbor 120.168.2.2
set routing-instances vpn-m5 protocols ospf domain-id 1.0.0.0
set routing-instances vpn-m5 protocols ospf export vpn-m5-import
set routing-instances vpn-m5 protocols ospf area 0.0.0.0 interface xe-0/0/1.0
set routing-instances vpn-m5 protocols ospf area 0.0.0.0 interface xe-0/0/0.0
```

R2

```
set chassis aggregated-devices ethernet device-count 5
set interfaces ge-1/2/0 unit 0 family inet address 120.168.100.1/30
set interfaces ge-1/2/0 unit 0 family iso
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces ge-1/2/1 unit 0 family inet address 120.168.101.1/30
```

```
set interfaces ge-1/2/1 unit 0 family iso
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces ge-1/3/0 gigether-options 802.3ad ae0
set interfaces ge-1/3/1 gigether-options 802.3ad ae0
set interfaces ge-1/3/2 gigether-options 802.3ad ae0
set interfaces ge-1/3/3 gigether-options 802.3ad ae0
set interfaces ge-1/3/4 gigether-options 802.3ad ae0
set interfaces ge-2/2/1 gigether-options 802.3ad ae1
set interfaces ge-2/2/2 gigether-options 802.3ad ae1
set interfaces ge-2/2/3 gigether-options 802.3ad ae1
set interfaces ge-2/2/4 gigether-options 802.3ad ae1
set interfaces ge-2/2/5 gigether-options 802.3ad ae1
set interfaces ge-2/2/6 gigether-options 802.3ad ae1
set interfaces ge-2/2/7 gigether-options 802.3ad ae1
set interfaces ge-2/2/8 gigether-options 802.3ad ae1
set interfaces ae0 aggregated-ether-options load-balance adaptive tolerance 10
set interfaces ae0 aggregated-ether-options link-speed 1g
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 unit 0 family inet address 120.168.104.1/30
set interfaces ae0 unit 0 family iso
set interfaces ae0 unit 0 family mpls
set interfaces ae1 aggregated-ether-options load-balance adaptive tolerance 10
set interfaces ae1 aggregated-ether-options link-speed 1g
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 unit 0 family inet address 120.168.105.1/30
set interfaces ae1 unit 0 family iso
set interfaces ae1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 120.168.0.4/32
set interfaces lo0 unit 0 family iso address 49.0001.1201.6800.0004.00
set accounting-options selective-aggregate-interface-stats disable
set protocols rsvp interface ge-1/2/0.0
set protocols rsvp interface ge-1/2/1.0
set protocols rsvp interface ae0.0
set protocols rsvp interface ae1.0
set protocols mpls interface ge-1/2/0.0
set protocols mpls interface ge-1/2/1.0
set protocols mpls interface ae0.0
set protocols mpls interface ae1.0
set protocols isis traffic-engineering family inet shortcuts
set protocols isis level 1 disable
set protocols isis interface ge-1/2/0.0
set protocols isis interface ge-1/2/1.0
set protocols isis interface ae0.0
```

```
set protocols isis interface ae1.0
set protocols isis interface lo0.0
```

R3

```
set chassis aggregated-devices ethernet device-count 5
set interfaces xe-4/0/0 unit 0 family inet address 120.168.9.1/30
set interfaces xe-4/0/0 unit 0 family mpls
set interfaces xe-4/0/1 unit 0 family inet address 120.168.10.1/30
set interfaces xe-4/0/1 unit 0 family mpls
set interfaces ge-5/0/1 gigether-options 802.3ad ae1
set interfaces ge-5/0/2 gigether-options 802.3ad ae1
set interfaces ge-5/0/3 gigether-options 802.3ad ae1
set interfaces ge-5/0/4 gigether-options 802.3ad ae1
set interfaces ge-5/0/5 gigether-options 802.3ad ae1
set interfaces ge-5/0/6 gigether-options 802.3ad ae1
set interfaces ge-5/0/7 gigether-options 802.3ad ae1
set interfaces ge-5/0/8 gigether-options 802.3ad ae1
set interfaces ge-5/3/0 gigether-options 802.3ad ae0
set interfaces ge-5/3/1 gigether-options 802.3ad ae0
set interfaces ge-5/3/2 gigether-options 802.3ad ae0
set interfaces ge-5/3/3 gigether-options 802.3ad ae0
set interfaces ge-5/3/4 gigether-options 802.3ad ae0
set interfaces ae0 aggregated-ether-options link-speed 1g
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 unit 0 family inet address 120.168.104.2/30
set interfaces ae0 unit 0 family iso
set interfaces ae0 unit 0 family mpls
set interfaces ae1 aggregated-ether-options link-speed 1g
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 unit 0 family inet address 120.168.105.2/30
set interfaces ae1 unit 0 family iso
set interfaces ae1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 120.168.0.9/32
set interfaces lo0 unit 0 family iso address 49.0001.1201.6800.0009.00
set routing-options router-id 120.168.0.9
set routing-options autonomous-system 55
set protocols rsvp interface xe-4/0/0.0
set protocols rsvp interface xe-4/0/1.0
set protocols rsvp interface ae0.0
set protocols rsvp interface ae1.0
set protocols mpls label-switched-path to-videl to 120.168.0.2
```

```
set protocols mpls interface xe-4/0/0.0
set protocols mpls interface xe-4/0/1.0
set protocols mpls interface ae0.0
set protocols mpls interface ae1.0
set protocols bgp group pe-routers type internal
set protocols bgp group pe-routers local-address 120.168.0.9
set protocols bgp group pe-routers family inet unicast
set protocols bgp group pe-routers family inet-vpn unicast
set protocols bgp group pe-routers neighbor 120.168.0.2
set protocols isis traffic-engineering family inet shortcuts
set protocols isis level 1 disable
set protocols isis interface ae0.0
set protocols isis interface ae1.0
set protocols isis interface lo0.0
set policy-options policy-statement nhs then next-hop self
set policy-options policy-statement vpn-m5-export term 1 from protocol bgp
set policy-options policy-statement vpn-m5-export term 1 from protocol direct
set policy-options policy-statement vpn-m5-export term 1 then community add vpn-m5-target
set policy-options policy-statement vpn-m5-export term 1 then accept
set policy-options policy-statement vpn-m5-export term 2 then reject
set policy-options policy-statement vpn-m5-import term 1 from protocol bgp
set policy-options policy-statement vpn-m5-import term 1 from protocol direct
set policy-options policy-statement vpn-m5-import term 1 from community vpn-m5-target
set policy-options policy-statement vpn-m5-import term 1 then accept
set policy-options policy-statement vpn-m5-import term 2 then reject
set policy-options community vpn-m5-target members target:55:100
set routing-instances vpn-m5 instance-type vrf
set routing-instances vpn-m5 interface xe-4/0/0.0
set routing-instances vpn-m5 interface xe-4/0/1.0
set routing-instances vpn-m5 route-distinguisher 120.168.0.9:1
set routing-instances vpn-m5 vrf-import vpn-m5-import
set routing-instances vpn-m5 vrf-export vpn-m5-export
set routing-instances vpn-m5 protocols bgp group ce type external
set routing-instances vpn-m5 protocols bgp group ce peer-as 100
set routing-instances vpn-m5 protocols bgp group ce as-override
set routing-instances vpn-m5 protocols bgp group ce neighbor 120.168.9.2
set routing-instances vpn-m5 protocols bgp group ce neighbor 120.168.10.2
set routing-instances vpn-m5 protocols ospf domain-id 1.0.0.0
set routing-instances vpn-m5 protocols ospf export vpn-m5-import
set routing-instances vpn-m5 protocols ospf area 0.0.0.0 interface xe-4/0/0.0
set routing-instances vpn-m5 protocols ospf area 0.0.0.0 interface xe-4/0/1.0
```

Configuring Adaptive Load Balancing

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure the R2 router:

NOTE: Repeat this procedure for the other routers, after modifying the appropriate interface names, addresses, and any other parameters for each router.

1. Specify the number of aggregated Ethernet interfaces to be created.

```
[edit chassis]
user@R2# set aggregated-devices ethernet device-count 5
```

2. Configure the Gigabit Ethernet interface link connecting R2 to R1.

```
[edit interfaces]
user@R2# set ge-1/2/0 unit 0 family inet address 120.168.100.1/30
user@R2# set ge-1/2/0 unit 0 family iso
user@R2# set ge-1/2/0 unit 0 family mpls
user@R2# set ge-1/2/1 unit 0 family inet address 120.168.101.1/30
user@R2# set ge-1/2/1 unit 0 family iso
user@R2# set ge-1/2/1 unit 0 family mpls
user@R2# set lo0 unit 0 family inet address 120.168.0.4/32
user@R2# set lo0 unit 0 family iso address 49.0001.1201.6800.0004.00
```

3. Configure the five member links of the ae0 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ge-1/3/0 gigether-options 802.3ad ae0
user@R2# set ge-1/3/1 gigether-options 802.3ad ae0
user@R2# set ge-1/3/2 gigether-options 802.3ad ae0
user@R2# set ge-1/3/3 gigether-options 802.3ad ae0
user@R2# set ge-1/3/4 gigether-options 802.3ad ae0
```

4. Configure the eight member links of the ae1 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ge-2/2/1 gigether-options 802.3ad ae1
user@R2# set ge-2/2/2 gigether-options 802.3ad ae1
user@R2# set ge-2/2/3 gigether-options 802.3ad ae1
user@R2# set ge-2/2/4 gigether-options 802.3ad ae1
user@R2# set ge-2/2/5 gigether-options 802.3ad ae1
user@R2# set ge-2/2/6 gigether-options 802.3ad ae1
user@R2# set ge-2/2/7 gigether-options 802.3ad ae1
user@R2# set ge-2/2/8 gigether-options 802.3ad ae1
```

5. Enable aggregate Ethernet load balancing on ae0 of R2.

```
[edit interfaces]
user@R2# set ae0 aggregated-ether-options load-balance adaptive tolerance 10
```

6. Configure the link speed for the ae0 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ae0 aggregated-ether-options link-speed 1g
```

7. Configure LACP on the ae0 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ae0 aggregated-ether-options lacp active
```

8. Configure the interface parameters for the ae0 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ae0 unit 0 family inet address 120.168.104.1/30
user@R2# set ae0 unit 0 family iso
user@R2# set ae0 unit 0 family mpls
```

9. Enable aggregate Ethernet load balancing on ae1 of R2.

```
[edit interfaces]
user@R2# set ae1 aggregated-ether-options load-balance adaptive tolerance 10
```

10. Configure the link speed for the ae1 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ae1 aggregated-ether-options link-speed 1g
```

11. Configure LACP on the ae1 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ae1 aggregated-ether-options lacp active
```

12. Configure the interface parameters for the ae1 aggregated Ethernet bundle.

```
[edit interfaces]
user@R2# set ae1 unit 0 family inet address 120.168.105.1/30
user@R2# set ae1 unit 0 family iso
user@R2# set ae1 unit 0 family mpls
```

13. Disable selective aggregate Ethernet statistics.

```
[edit accounting-options]
user@R2# set selective-aggregate-interface-stats disable
```

14. Configure RSVP on all the interfaces of R2 and on the AE bundles.

```
[edit protocols]
user@R2# set rsvp interface ge-1/2/0.0
user@R2# set rsvp interface ge-1/2/1.0
user@R2# set rsvp interface ae0.0
user@R2# set rsvp interface ae1.0
```

15. Configure MPLS on all the interfaces of R2 and on the AE bundles.

```
[edit protocols]
user@R2# set mpls interface ge-1/2/0.0
user@R2# set mpls interface ge-1/2/1.0
user@R2# set mpls interface ae0.0
user@R2# set mpls interface ae1.0
```

16. Configure IS-IS on all the interfaces of R2 and on the AE bundles.

```
[edit protocols]
user@R2# set isis traffic-engineering family inet shortcuts
user@R2# set isis level 1 disable
user@R2# set isis interface ge-1/2/0.0
user@R2# set isis interface ge-1/2/1.0
user@R2# set isis interface ae0.0
user@R2# set isis interface ae1.0
user@R2# set isis interface lo0.0
```

Results

From configuration mode, confirm your configuration by entering the `show chassis`, `show interfaces`, `show accounting-options`, and `show protocols` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R2# show chassis
aggregated-devices {
    ethernet {
        device-count 5;
    }
}
```

```
user@R2# show interfaces
ge-1/2/0 {
    unit 0 {
        family inet {
            address 120.168.100.1/30;
        }
```

```
        family iso;
        family mpls;
    }
}
ge-1/2/1 {
    unit 0 {
        family inet {
            address 120.168.101.1/30;
        }
        family iso;
        family mpls;
    }
}
ge-1/3/0 {
    gigether-options {
        802.3ad ae0;
    }
}
ge-1/3/1 {
    gigether-options {
        802.3ad ae0;
    }
}
ge-1/3/2 {
    gigether-options {
        802.3ad ae0;
    }
}
ge-1/3/3 {
    gigether-options {
        802.3ad ae0;
    }
}
ge-1/3/4 {
    gigether-options {
        802.3ad ae0;
    }
}
ge-2/2/1 {
    gigether-options {
        802.3ad ae1;
    }
}
```

```
ge-2/2/2 {
    gigether-options {
        802.3ad ae1;
    }
}
ge-2/2/3 {
    gigether-options {
        802.3ad ae1;
    }
}
ge-2/2/4 {
    gigether-options {
        802.3ad ae1;
    }
}
ge-2/2/5 {
    gigether-options {
        802.3ad ae1;
    }
}
ge-2/2/6 {
    gigether-options {
        802.3ad ae1;
    }
}
ge-2/2/7 {
    gigether-options {
        802.3ad ae1;
    }
}
ge-2/2/8 {
    gigether-options {
        802.3ad ae1;
    }
}
ae0 {
    aggregated-ether-options {
        load-balance {
            adaptive tolerance 10;
        }
        link-speed 1g;
        lacp {
            active;
```

```
        }
    }
    unit 0 {
        family inet {
            address 120.168.104.1/30;
        }
        family iso;
        family mpls;
    }
}
ae1 {
    aggregated-ether-options {
        load-balance {
            adaptive tolerance 10;
        }
        link-speed 1g;
        lacp {
            active;
        }
    }
    unit 0 {
        family inet {
            address 120.168.105.1/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 120.168.0.4/32;
        }
        family iso {
            address 49.0001.1201.6800.0004.00;
        }
```

```
    }
}
```

```
user@R2# show accounting-options
selective-aggregate-interface-stats disable;
```

```
user@R2# show protocols
rsvp {
    interface ge-1/2/0.0;
    interface ge-1/2/1.0;
    interface ae0.0;
    interface ae1.0;
}
mpls {
    interface ge-1/2/0.0;
    interface ge-1/2/1.0;
    interface ae0.0;
    interface ae1.0;
}
isis {
    traffic-engineering {
        family inet {
            shortcuts;
        }
    }
    level 1 disable;
    interface ge-1/2/0.0;
    interface ge-1/2/1.0;
    interface ae0.0;
    interface ae1.0;
    interface lo0.0;
}
```

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Adaptive Load Balancing on ae0

Purpose

Verify that packets received on the ae0 aggregated Ethernet bundle are load-balanced among the five member links.

Action

From operational mode, run the `show interfaces ae0 extensive` command.

```
user@R2> show interfaces ae0 extensive
Logical interface ae0.0 (Index 325) (SNMP ifIndex 917) (Generation 134)
    Flags: SNMP-Traps 0x4004000 Encapsulation: ENET2
    Statistics        Packets        pps         Bytes          bps
    Bundle:
        Input :        848761          9      81247024         7616
        Output:  166067308909    3503173 126900990064983  21423804256
    Adaptive Statistics:
        Adaptive Adjusts:        264
        Adaptive Scans  :      27682
        Adaptive Updates:         10
    Link:
      ge-1/3/0.0
        Input :        290888          5      29454436         3072
        Output:   33183442699     704569 25358563587277   4306031760
      ge-1/3/1.0
        Input :        162703          1      14806325          992
        Output:   33248375409     705446 25406995966732   4315342152
      ge-1/3/2.0
```

```
    Input :        127448          1      12130566          992
    Output:   33184552729     697572 25354827700261   4267192376
  ge-1/3/3.0
    Input :        121044          1      11481262         1280
    Output:   33245875402     697716 25405953405192   4265750584
  ge-1/3/4.0
    Input :        146678          1      13374435         1280
    Output:   33205071207     697870 25374651121458   4269487384
```

Meaning

The member links of the ae0 aggregated Ethernet bundle are fully utilized with adaptive load balancing.

Platform-Specific Aggregated Ethernet Load Balancing Behavior

IN THIS SECTION

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

Use the following table to review platform-specific behaviors for your platform.

Platform-Specific Aggregated Ethernet Load Balancing Behavior

Platform	Difference
ACX Series	• On ACX7000 Series of devices, ae member interfaces do not load balance egress traffic. • On ACX7000 Series of devices, you must configure the `set forwarding-options hash-key` statement to use all available member interfaces for load balancing.
EX Series	• You can configure up 480 LAG bundles on EX9200 switches.

(Continued)

Platform	Difference
MX Series	• You can configure up 480 LAG bundles on MX Series routers that support this feature. • You can perform uniform load balancing and rebalancing on MX Series routers with MPCs that support this feature. Rebalancing is not supported when load-balancing is skewed or distorted owing to a change in the number of flows.
PTX Series	• Adaptive load balancing is not supported on PTX Series devices if the VLAN ID is configured on the aggregated Ethernet interface. • The `pps` and `scan-interval` optional keywords are supported on PTX Series Packet Transport Routers only.
QFX Series	• Adaptive load balancing is not supported on QFX10000 switches if the VLAN ID is configured on the aggregated Ethernet interface.

Change History Table

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

Release	Description
14.1	Starting with Junos OS Release 14.1, aggregated Ethernet load balancing is enhanced to provide two solutions for resolving genuine traffic imbalance on aggregated Ethernet bundles on PICs or FPCs of PTX Series Packet Transport Routers.
13.3	Starting with Junos OS Release 13.3, aggregated Ethernet load balancing is enhanced to provide two solutions for resolving genuine traffic imbalance on aggregated Ethernet bundles on MICs or MPCs of MX Series routers.
10.1	Starting with Junos OS Release 10.1, you can also configure the load balancing hash key for Layer 2 traffic to use fields in the Layer 3 and Layer 4 headers using the `payload` statement.

PART 6

Configuring Graceful Routing Engine Switchover (GRES)

Understanding Graceful Routing Switchover

IN THIS SECTION

Understanding Graceful Routing Engine Switchover

IN THIS SECTION

This topic contains the following sections:

Graceful Routing Engine Switchover Concepts

The graceful Routing Engine switchover (GRES) feature in Junos OS and Junos OS Evolved enables a device with redundant Routing Engines to continue forwarding packets even if one Routing Engine fails. GRES preserves interface and kernel information, and traffic is not interrupted. However, GRES does not preserve the control plane.

Neighboring devices detect that the device has experienced a restart and react to the event in a manner prescribed by individual routing protocol specifications.

To preserve routing during a switchover, GRES must be combined with either:

- Graceful restart protocol extensions

- *Nonstop active routing* (NSR)

Any updates to the primary Routing Engine are replicated to the backup Routing Engine as soon as they occur.

NOTE: Because of its synchronization requirements and logic, NSR/GRES performance is limited by the slowest Routing Engine in the system.

Primary Role switches to the backup Routing Engine if:

- The primary Routing Engine kernel stops operating.
- The primary Routing Engine experiences a hardware failure.
- The administrator initiates a manual switchover.

NOTE: To quickly restore or to preserve routing protocol state information during a switchover, GRES must be combined with either graceful restart or nonstop active routing, respectively. For more information about graceful restart, see Graceful Restart Concepts. For more information about nonstop active routing, see Nonstop Active Routing Concepts.

If the backup Routing Engine does not receive a keepalive from the primary Routing Engine after 2 seconds, it determines that the primary Routing Engine has failed; and assumes primary role.

The Packet Forwarding Engine:

- Seamlessly disconnects from the old primary Routing Engine
- Reconnects to the new primary Routing Engine
- Does not reboot
- Does not interrupt traffic

The new primary Routing Engine and the Packet Forwarding Engine then become synchronized. If the new primary Routing Engine detects that the Packet Forwarding Engine state is not up to date, it resends state update messages.

Note the following GRES behaviors, recommendations, or requirements:

- Starting with Junos OS Release 12.2, if adjacencies between the restarting device and the neighboring peer 'helper' devices time out, graceful restart protocol extensions are unable to notify the peer 'helper' devices about the impending restart. Graceful restart can then stop and cause interruptions in traffic.

To ensure that these adjacencies are maintained, change the `hold-time` for IS-IS protocols from the default of 27 seconds to a value higher than 40 seconds.

- Successive Routing Engine switchover events must be a minimum of 240 seconds (4 minutes) apart after both Routing Engines have come up.

  If the device displays a warning message similar to:

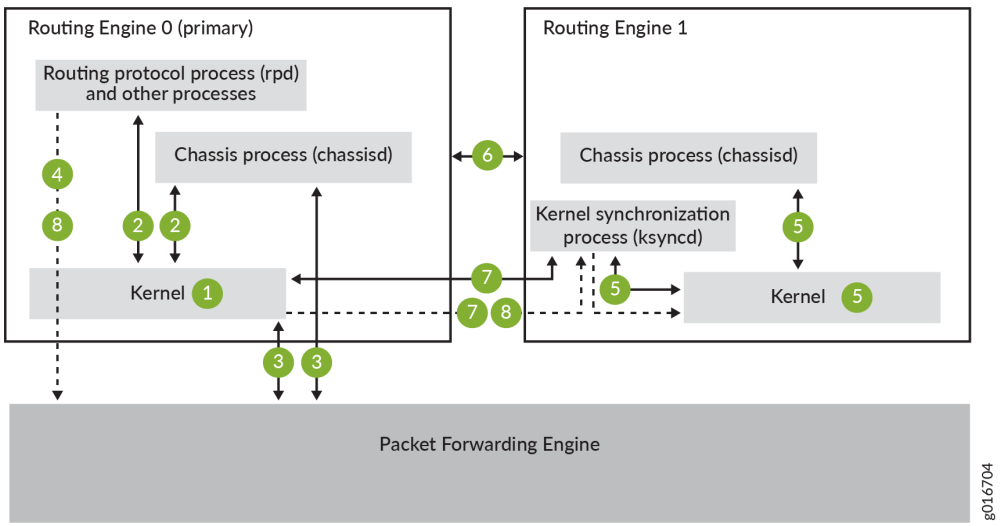
  ```
  Standby Routing Engine is not ready for graceful switchover. Packet Forwarding Engines that
  are not ready for graceful switchover might be reset
  ```

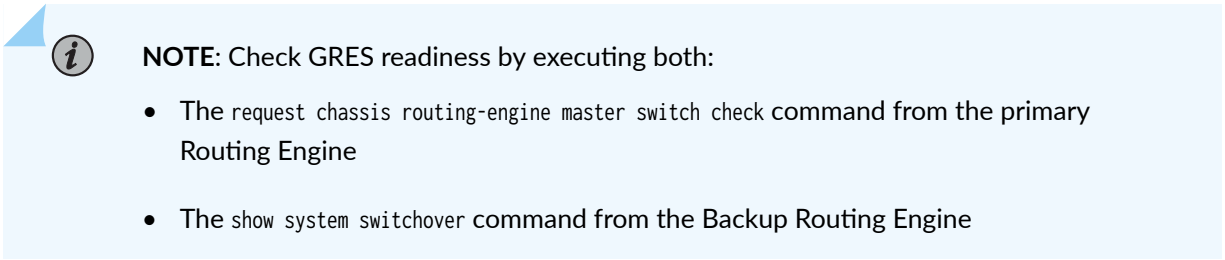
  then do not attempt a switchover. If you choose to proceed with switchover, the device resets only the Packet Forwarding Engines that were not ready for graceful switchover. None of the FPCs should spontaneously restart. We recommend that you wait until the warning no longer appears and then proceed with the switchover.

- We do *not* recommend:
  - Doing a commit operation on the backup Routing Engine when GRES is enabled on the device.
  - Enabling GRES on the backup Routing Engine in *any* scenario.

Figure 9 on page 197 shows the system architecture of graceful Routing Engine switchover and the process a routing platform follows to prepare for a switchover.

Figure 9: Preparing for a Graceful Routing Engine Switchover





NOTE: Check GRES readiness by executing both:

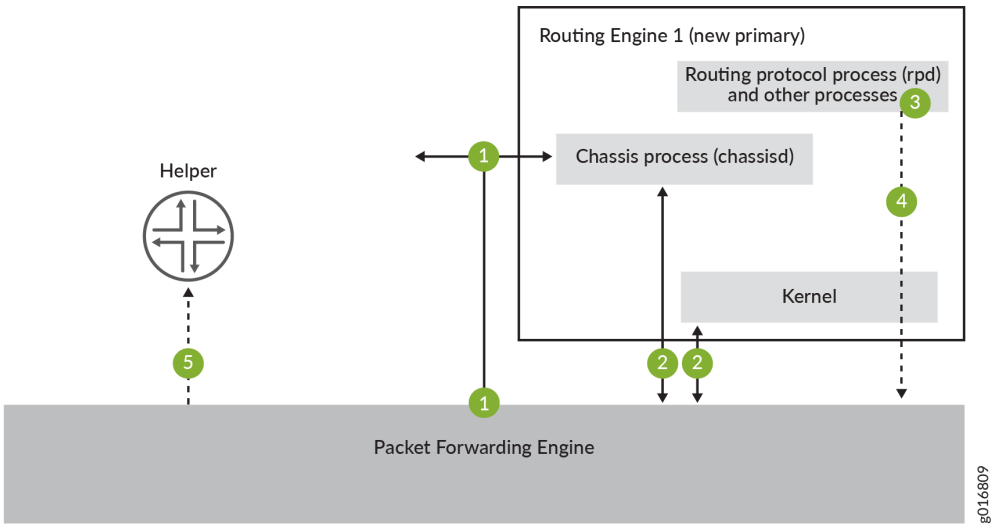
- The `request chassis routing-engine master switch check` command from the primary Routing Engine
- The `show system switchover` command from the Backup Routing Engine

The switchover preparation process for GRES is as follows:

1. The primary Routing Engine starts.
2. The routing platform processes (such as the chassis process [chassisd]) start.
3. The Packet Forwarding Engine starts and connects to the primary Routing Engine.
4. All state information is updated in the system.
5. The backup Routing Engine starts.
6. The system determines whether GRES has been enabled.
7. The kernel synchronization process (ksyncd) synchronizes the backup Routing Engine with the primary Routing Engine.
8. After ksyncd completes the synchronization, all state information and the forwarding table are updated.

Figure 10 on page 199 shows the effects of a switchover on the routing (or switching)platform.

Figure 10: Graceful Routing Engine Switchover Process



A switchover process comprises the following steps:

1. When keepalives from the primary Routing Engine are lost, the system switches over gracefully to the backup Routing Engine.
2. The Packet Forwarding Engine connects to the backup Routing Engine, which becomes the new primary.
3. Routing platform processes that are not part of GRES (such as the routing protocol process rpd) restart.
4. State information learned from the point of the switchover is updated in the system.
5. If configured, graceful restart protocol extensions collect and restore routing information from neighboring peer *helper* devices.

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

Review the "Platform-Specific GRES Behavior" on page 206 section for notes related to your platform.

Effects of a Routing Engine Switchover

Table 7 on page 200 describes the effects of a Routing Engine switchover when different features are enabled:

- No high availability features

- Graceful Routing Engine switchover
- Graceful restart
- Nonstop active routing

Table 7: Effects of a Routing Engine Switchover

Feature	Benefits	Considerations
Dual Routing Engines only (no features enabled)	• When the switchover to the new primary Routing Engine is complete, routing convergence takes place and traffic is resumed.	• All physical interfaces are taken offline. • Packet Forwarding Engines restart. • The backup Routing Engine restarts the routing protocol process (rpd). • All hardware and interfaces are discovered by the new primary Routing Engine. • The switchover takes several minutes. • All of the device's adjacencies are aware of the physical (interface alarms) and routing (topology) changes.
GRES enabled	• During the switchover, interface and kernel information is preserved. • The switchover is faster because the Packet Forwarding Engines are not restarted.	• The new primary Routing Engine restarts the routing protocol process (rpd). • All hardware and interfaces are acquired by a process that is similar to a warm restart. • All adjacencies are aware of the device's change in state.

Table 7: Effects of a Routing Engine Switchover *(Continued)*

Feature	Benefits	Considerations
GRES *and* NSR enabled	• Traffic is not interrupted during the switchover. • Interface and kernel information are preserved.	• Unsupported protocols must be refreshed using the normal recovery mechanisms inherent in each protocol.
GRES *and* graceful restart enabled	• Traffic is not interrupted during the switchover. • Interface and kernel information are preserved. • Graceful restart protocol extensions quickly collect and restore routing information from the neighboring devices.	• Neighbors are required to support graceful restart, and a wait interval is required. • The routing protocol process (rpd) restarts. • For certain protocols, a significant change in the network can cause graceful restart to stop. • Starting with Junos OS Release 12.2, if adjacencies between the restarting device and the neighboring peer 'helper' devices time out, graceful restart can stop and cause interruptions in traffic.

Graceful Routing Engine Switchover on Aggregated Services Interfaces

If a graceful Routing Engine switchover (GRES) is triggered by an operational mode command, the device does not preserve the state of aggregated services interfaces (ASIs). For example:

```
request interface <switchover | revert> asi-interface
```

However, if GRES is triggered by a CLI commit or FPC restart or crash, the backup Routing Engine updates the ASI state. For example:

```
set interface si-x/y/z disable
commit
```

Or:

```
request chassis fpc restart
```

SEE ALSO

Understanding High Availability Features on Juniper Networks Routers | 2

Graceful Routing Engine Switchover System Requirements

Configuring Graceful Routing Engine Switchover | 208

Configuring Graceful Routing Engine Switchover in a Virtual Chassis

Configuring Graceful Routing Engine Switchover in a Virtual Chassis

Requirements for Routers with a Backup Router Configuration

Example: Configuring IS-IS for GRES with Graceful Restart

Graceful Routing Engine Switchover System Requirements

IN THIS SECTION

Graceful Routing Engine switchover is supported on all routing (or switching) platforms that contain dual Routing Engines. All Routing Engines configured for graceful Routing Engine switchover must run the

same Junos OS release. Hardware and software support for graceful Routing Engine switchover is described in the following sections:

Graceful Routing Engine Switchover Platform Support

To enable graceful Routing Engine switchover, your system must meet these minimum requirements:

- MX960 router—Junos OS Release 8.3 or later
- MX480 router—Junos OS Release 8.4 or later (8.4R2 recommended)
- MX240 router—Junos OS Release 9.0 or later
- PTX5000 router—Junos OS Release 12.1X48 or later
- EX Series switches with dual Routing Engines or in a Virtual Chassis — Junos OS Release 9.2 or later for EX Series switches
- QFX Series switches in a Virtual Chassis —Junos OS Release 13.2 or later for the QFX Series
- EX Series or QFX Series switches in a Virtual Chassis Fabric —Junos OS Release 13.2X51-D20 or later for the EX Series and QFX Series switches

For more information about support for graceful Routing Engine switchover, see the sections that follow.

Graceful Routing Engine Switchover Feature Support

Graceful Routing Engine switchover supports most Junos OS features in Release 5.7 and later. Particular Junos OS features require specific versions of Junos OS. See Table 8 on page 203.

Table 8: Graceful Routing Engine Switchover Feature Support

Application	Junos OS Release
Aggregated Ethernet interfaces with Link Aggregation Control Protocol (LACP) and aggregated SONET interfaces	6.2
Asynchronous Transfer Mode (ATM) virtual circuits (VCs)	6.2

Table 8: Graceful Routing Engine Switchover Feature Support *(Continued)*

Application	Junos OS Release
Logical systems **NOTE**: In Junos OS Release 9.3 and later, the logical router feature is renamed to logical system.	6.3
Multicast	6.4 (7.0 for TX Matrix router)
Multilink Point-to-Point Protocol (MLPPP) and Multilink Frame Relay (MLFR)	7.0
Automatic Protection Switching (APS)—The current active interface (either the designated working or the designated protect interface) remains the active interface during a Routing Engine switchover.	7.4
Point-to-multipoint Multiprotocol Label Switching MPLS LSPs (transit only)	7.4
Compressed Real-Time Transport Protocol (CRTP)	7.6
Virtual private LAN service (VPLS)	8.2
Ethernet Operation, Administration, and Management (OAM) as defined by IEEE 802.3ah	8.5
Extended DHCP relay agent	8.5
Ethernet OAM as defined by IEEE 802.1ag	9.0
Packet Gateway Control Protocol (PGCP) process (pgcpd) on Multiservices 500 PICs on T640 routers.	9.0
Subscriber access	9.4
Layer 2 Circuit and LDP-based VPLS pseudowire redundant configuration	9.6

The following constraints apply to graceful Routing Engine switchover feature support:

- When graceful Routing Engine switchover and aggregated Ethernet interfaces are configured in the same system, the aggregated Ethernet interfaces must not be configured for fast-polling LACP. When fast polling is configured, the LACP polls time out at the remote end during the Routing Engine primary-role switchover. When LACP polling times out, the aggregated link and interface are disabled. The Routing Engine primary role change is fast enough that standard and slow LACP polling do not time out during the procedure.

  NOTE: MACSec sessions will flap upon Graceful Routing Engine switchover.

  Starting with Junos OS Release 13.2, when a graceful Routing Engine switchover occurs, the VRRP state does not change. VRRP is supported by graceful Routing Engine switchover only in the case that PPM delegation is enabled (which the default).

Graceful Routing Engine Switchover and Subscriber Access

Graceful Routing Engine switchover currently supports most of the features directly associated with dynamic DHCP and dynamic PPPoE subscriber access. Graceful Routing Engine switchover also supports the unified in-service software upgrade (ISSU) for the DHCP access model and the PPPoE access model used by subscriber access.

NOTE: When graceful Routing Engine switchover is enabled for subscriber management, all Routing Engines in the router must have the same amount of DRAM for stable operation.

Graceful Routing Engine Switchover PIC Support

Graceful Routing Engine switchover is supported on most PICs, except for the services PICs listed in this section. The PIC must be on a supported routing platform running the appropriate version of Junos OS. For information about FPC types, FPC/PIC compatibility, and the initial Junos OS Release in which an FPC supported a particular PIC, see the PIC guide for your router platform.

The following constraints apply to graceful Routing Engine switchover support for services PICs:

- You can include the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level on a router with Adaptive Services, Multiservices, and Tunnel Services PICs configured on it and successfully commit the configuration. However, all services on these PICs—except the Layer 2 service packages and extension-provider and SDK applications on Multiservices PICs—are reset during a switchover.

- Graceful Routing Engine switchover is not supported on any Monitoring Services PICs or Multilink Services PICs. If you include the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level on a router with either of these PIC types configured on it and issue the `commit` command, the commit fails.
- Graceful Routing Engine switchover is not supported on Multiservices 400 PICs configured for monitoring services applications. If you include the `graceful-switchover` statement, the commit fails.

NOTE: When an unsupported PIC is online, you cannot enable graceful Routing Engine switchover. If graceful Routing Engine switchover is already enabled, an unsupported PIC cannot come online.

SEE ALSO

Understanding High Availability Features on Juniper Networks Routers | 2

Understanding Graceful Routing Engine Switchover

Configuring Graceful Routing Engine Switchover | 208

Configuring Graceful Routing Engine Switchover in a Virtual Chassis

Requirements for Routers with a Backup Router Configuration

Platform-Specific GRES Behavior

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

Use the following table to review platform-specific behaviors for your platform:

Platform	Difference
MX Series	• When you perform GRES on MX Series routers, you must execute the `clear synchronous-ethernet wait-to-restore` operational mode command on the new primary Routing Engine to clear the wait-to-restore timer on it. This is because the `clear synchronous-ethernet wait-to-restore` operational mode command clears the wait-to-restore timer only on the local Routing Engine. • For MX Series routers using enhanced subscriber management, the new backup Routing Engine (the former primary Routing Engine) will reboot when a graceful Routing Engine switchover is performed. This cold restart resynchronizes the backup Routing Engine state with that of the new primary Routing Engine, preventing discrepancies in state that might have occurred during the switchover. • MX Series Routers that have distributed periodic packet management (PPM) enabled can configure graceful Routing Engine switchover and have aggregated Ethernet interfaces configured for fast-polling LACP on the same device. • Graceful Routing Engine switchover supports all Dense Port Concentrators (DPCs) on the MX Series 5G Universal Routing Platforms running the appropriate version of Junos OS as shown in "Platform-Specific GRES Behavior" on page 206.
PTX Series	• On PTX10004, PTX10008, and PTX10016 devices running Junos OS Evolved, GRES is enabled by default and cannot be disabled.
QFX Series	• When you enable nonstop routing with GRES on switches in the QFX10000 line that have redundant Routing Engines, we strongly recommend that you configure the `nsr-phantom-holdtime` *seconds* statement at the `[edit routing-options]` hierarchy level. Doing so helps to prevent traffic loss during a switchover. If you configure this statement, phantom IP addresses remain in the kernel during the switchover until the specified hold-time interval expires. After the interval expires, the device adds the corresponding routes to the appropriate routing tables. In an Ethernet VPN (EVPN)-VXLAN environment, we recommend that you specify a hold-time value of 300 seconds (5 minutes). This option doesn't apply to QFX10002 switches, which don't have redundant Routing Engines and don't support GRES.

Change History Table

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

Release	Description
13.2	Starting with Junos OS Release 13.2, when a graceful Routing Engine switchover occurs, the VRRP state does not change.
12.2	Starting with Junos OS Release 12.2, if adjacencies between the restarting device and the neighboring peer 'helper' devices time out, graceful restart protocol extensions are unable to notify the peer 'helper' devices about the impending restart.
12.2	Starting with Junos OS Release 12.2, if adjacencies between the restarting device and the neighboring peer 'helper' devices time out, graceful restart can stop and cause interruptions in traffic.

Configuring Graceful Routing Engine Switchover

SUMMARY

Learn how to configure Graceful Routing Engine Switchover (GRES) with the following steps and examples.

IN THIS SECTION

Requirements for Routers with a Backup Router Configuration

If your Routing Engine configuration includes a `backup-router` statement or an `inet6-backup-router` statement, you can also use the `destination` statement to specify a subnet address or multiple subnet addresses for the backup router. Include destination subnets for the backup Routing Engine at the `[edit system (backup-router | inet6-backup-router) address]` hierarchy level.

NOTE: If you have a backup router configuration in which multiple static routes point to a gateway from the management Ethernet interface, you must configure prefixes that are more specific than the static routes or include the **retain** flag at the `[edit routing-options static route]` hierarchy level.

For example, if you configure the static route 172.16.0.0/12 from the management Ethernet interface for management purposes, you must specify the backup router configuration as follows:

```
backup-router 172.29.201.62 destination [172.16.0.0/13 172.16.128.0/13]
```

Enabling Graceful Routing Engine Switchover

In most cases, graceful Routing Engine switchover (GRES) is disabled by default. To configure GRES, include the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level.

```
[edit chassis redundancy]
graceful-switchover;
```

NOTE: GRES is enabled by default on Junos OS Evolved devices with dual Routing Engines. GRES configuration is only required for devices where GRES is disabled by default.

When you enable GRES, the command-line interface (CLI) indicates which Routing Engine you are using. For example:

```
{master} [edit]
user@host#
```

To disable GRES, delete the `graceful-switchover` statement from the `[edit chassis redundancy]` hierarchy level.

Configuring Graceful Routing Engine Switchover with Graceful Restart

When using GRES with Graceful Restart, if adjacencies between the Routing Engine and the neighboring peer 'helper' routers time out, graceful restart protocol extensions are unable to notify the peer 'helper' routers about the impending restart. Graceful restart can then stop and cause interruptions in traffic.

To ensure that these adjacencies are kept, change the *hold-time* for IS-IS protocols from the default of 27 seconds to a value higher than 40 seconds.

Synchronizing the Routing Engine Configuration

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the primary Routing Engine configuration.

When you configure GRES, you can bring the backup Routing Engine online after the primary Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Only when you enable the graceful Routing Engine switchover, you can copy the running Junos OS version of the primary Routing Engine to the backup Routing Engine.

NOTE: If the system is in ISSU state, you cannot copy the running Junos OS version of the primary Router Engine.

You can enable automatic synchronization of the primary Routing Engine configuration with the backup Routing Engine by including the events CHASSISD_SNMP_TRAP7 statement at the [edit event-options policy *policy-name*] hierarchy level.

CHASSISD_SNMP_TRAP7 is a system event logging message that the chassis process (chassisd) generates a Simple Network Management Protocol (SNMP) trap with the seven indicated argument-

value pairs. An example of an event script to trigger automatic synchronization of primary to the backup Routing Engine is as follows:

```
[edit event-options]
policy UPGRADE-BACKUPRE {
events CHASSISD_SNMP_TRAP7;
attributes-match {
CHASSISD_SNMP_TRAP7.value5 matches "Routing Engine";
CHASSISD_SNMP_TRAP7.trap matches "Fru Online";
CHASSISD_SNMP_TRAP7.argument5 matches jnxFruName;
}
then {
event-script auto-image-upgrade.slax {
arguments {
trap "{$$.trap}";
value5 "{$$.value5}";
argument5 "{$$.argument5}";
}
}
}
}
event-script {
file auto-image-upgrade.slax;
}
```

After receiving this event, the event policy on the primary Router Engine is triggered and the image available in the */var/sw/pkg* path is pushed to the backup Router Engine upgrade. During script execution, the image is copied to the backup Routing Engine's */var/sw/pkg* path.

NOTE: If the image is not available in the */var/sw/pkg* path, the script is terminated with an appropriate syslog message.

The Junos automation scripts is synchronized automatically.

After the primary Router Engine is rebooted, the event script available at the */usr/libexec/scripts/event/auto-image-upgrade.slax* must be copied to the */var/db/scripts/event path*.

NOTE: For devices that support enhanced subscriber management, the new backup Routing Engine (the former primary Routing Engine) will reboot when a graceful Routing Engine switchover is performed. This cold restart resynchronizes the backup Routing

Engine state with that of the new primary Routing Engine, preventing discrepancies in state that might have occurred during the switchover.

Verifying Graceful Routing Engine Switchover Operation

To verify whether GRES is enabled on the backup Routing Engine, issue the `show system switchover` command. When the output of the command indicates that the **Graceful switchover** field is set to **On**, GRES is operational. The status of the kernel database and configuration database synchronization between Routing Engines is also provided. For example:

```
Graceful switchover: On
Configuration database: Ready
Kernel database: Ready
Peer state: Steady state
```

NOTE: You must issue the `show system switchover` command on the backup Routing Engine. This command is not supported on the primary Routing Engine.

For more information about the `show system switchover` command, see the CLI Explorer.

Configuring Graceful Routing Engine Switchover in a Virtual Chassis

In a Virtual Chassis, one member switch is assigned the primary role and has the primary Routing Engine. Another member switch is assigned the backup role and has the backup Routing Engine. Graceful Routing Engine switchover (GRES) enables the primary and backup Routing Engines in a Virtual Chassis configuration to switch from the primary to backup without interruption to packet forwarding as a hitless failover solution. When you configure graceful Routing Engine switchover, the backup Routing Engine automatically synchronizes with the primary Routing Engine to preserve kernel state information and the forwarding state.

To set up the Virtual Chassis configuration to use graceful Routing Engine switchover (GRES):

1. Set up a minimum of two switches in a Virtual Chassis configuration with primary-role priority of 255:

```
[edit]
user@switch# set virtual-chassis member 0 mastership-priority 255

[edit]
user@switch# set virtual-chassis member 1 mastership-priority 255
```

2. Set up graceful Routing Engine switchover:

```
[edit]
user@switch# set chassis redundancy graceful-switchover
```

Commit the configuration.

NOTE: We recommend that you use the `commit synchronize` command to save any configuration changes that you make to a multimember Virtual Chassis.

Preventing Graceful Routing Engine Switchover in the Case of Slow Disks

Unexpected slow disk access can happen for various reasons—a faulty or bad sector, for example—causing a hold up of the normal operation of processes such as the routing process (rpd). Eventually, the router's performance will be impacted. Under these circumstances, it may take longer for the typical failover mechanism to be triggered.

Juniper Networks has introduced a disk monitoring daemon to solve this dilemma. The daemon detects slow disk access and initiates failover. Failover can minimize the traffic impact and ease the load on the original primary Routing Engine for its backlog clean up.

However, there are instances when you might not want failover to occur. You might commit a large set of changes or even minor changes that might lead to a series of updates on the routing topology. Such activity could lead to extensive disk access delay and, therefore, trigger failover. For expected disk access delays like this, where you do not want to trigger failover, you can choose to not have failover occur by setting the `chassis redundancy failover not-on-disk-underperform` configuration command. Another way is to disable the disk monitoring daemon completely by setting the `system processes gstatd disable` command.

To prevent failovers in the case of slow disks in the Routing Engine:

- Set the option for preventing gstatd from initiating failovers in response to slow disks at the `[edit chassis redundancy failover]` hierarchy level.

```
[edit]
user@host# set chassis redundancy failover not-on-disk-underperform
```

Resetting Local Statistics

When you enable graceful Routing Engine switchover, the primary Routing Engine configuration is copied and loaded to the backup Routing Engine. User files, accounting information, and trace options information are not replicated to the backup Routing Engine.

When a graceful Routing Engine switchover occurs, local statistics such as process statistics and networking statistics are displayed as a cumulative value from the time the process first came online. Because processes on the primary Routing Engine can start at different times from the processes on the backup Routing Engine, the statistics on the two Routing Engines for the same process might differ. After a graceful Routing Engine switchover, we recommend that you issue the **clear interface statistics (*interface-name* | all)** command to reset the cumulative values for local statistics. Forwarding statistics are not affected by graceful Routing Engine switchover.

For information about how to use the **clear** command to clear statistics and protocol database information, see the CLI Explorer.

NOTE: The **clear firewall** command cannot be used to clear the Routing Engine filter counters on a backup Routing Engine that is enabled for graceful Routing Engine switchover.

Example: Configuring IS-IS for GRES with Graceful Restart

IN THIS SECTION

This example shows how to configure the Routing Engine's graceful restart protocol extensions using the intermediate system to intermediate system (IS-IS) interior gateway protocol (IGP) to successfully enable graceful Routing Engine switchover (GRES) with graceful restart.

Requirements

GRES prevents interruptions in network traffic if the primary Routing Engine fails when combined with either:

- Graceful restart
- Nonstop active routing (NSR)

Before you follow the directions here to configure graceful restart, be sure you have enabled GRES, which is disabled by default. See "Configuring Graceful Routing Engine Switchover" on page 208 for more information.

Overview

If adjacencies between the Routing Engine and the neighboring peer 'helper' routers time out, graceful restart protocol extensions are unable to notify the peer 'helper' routers about the impending restart. Graceful restart can then stop and cause interruptions in traffic.

To ensure that these adjacencies are kept, change the hold-time for IS-IS protocols from the default of 27 seconds to a value higher than 40 seconds.

If your system uses the open shortest pathway first (OSPF) protocol instead of IS-IS, see *Example: Configuring OSPF Timers* for configuration information.

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure the hold-time, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the different hierarchy levels shown.

Each interface must be set individually, with a value for each level that the routing device operates on. The minimum recommended value of 41 seconds is used in this example, your system may require a higher value based on size and traffic.

Level 1 and level 2 can be set to different values.

[edit protocols]

```
set protocols isis interface ge-1/2/0 level 1 hold-time 41
set protocols isis interface ge-1/2/0 level 2 hold-time 41
```

[edit logical-systems logical-system-name}

```
set protocols isis interface ge-1/2/0 level 1 hold-time 41
set protocols isis interface ge-1/2/0 level 2 hold-time 41
```

[edit logical-systems logical-system-name routing-instances routing-instance-name]

```
set protocols isis interface ge-1/2/0 level 1 hold-time 41
set protocols isis interface ge-1/2/0 level 2 hold-time 41
```

[edit routing-instances routing-instance-name]

```
set protocols isis interface ge-1/2/0 level 1 hold-time 41
set protocols isis interface ge-1/2/0 level 2 hold-time 41
```

Configuring the IS-IS Protocol Hold Time for Graceful Restart

Step-by-Step Procedure

To configure the IS-IS hold-time for graceful restart:

1. Locate or set the interfaces.

```
set protocols isis interface interface-name
```

2. Set the network level and the hold-time in seconds for that level.

```
set protocols isis interface interface-name level 1 hold-time 41
```

3. If the routing device functions on more than one level, set the value for the other level.

```
set protocols isis interface interface-name level 2 hold-time 41
```

4. If you are done configuring the routing device, commit the configuration.

NOTE: Repeat the entire configuration on all routing devices in a shared network.

Results

Verification

IN THIS SECTION

Verifying the IS-IS Protocol Hold Time for Graceful Restart

Purpose

Verify that the IS-IS protocol hold-time is set to 41 seconds or greater to ensure that graceful restart is enabled.

Action

Confirm your configuration by entering the `show isis adjacency brief` command from operational mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

Meaning

A high enough IS-IS protocol hold-time value allows your system configuration to restart and ensures that even if a Routing Engine fails, traffic continues.

RELATED DOCUMENTATION

Understanding Graceful Routing Engine Switchover

graceful-switchover

hold-time

Configuring Ethernet Automatic Protection Switching

SUMMARY

Learn how to configure Ethernet automatic protection switching (APS) for high availability.

IN THIS SECTION

Ethernet Automatic Protection Switching Overview

IN THIS SECTION

Ethernet automatic protection switching (APS) is a linear protection scheme designed to protect VLAN based Ethernet networks.

With Ethernet APS, a protected domain is configured with two paths, a working path and a protection path. Both working and protection paths can be monitored using an Operations Administration Management (OAM) protocol like Connectivity Fault Management (CFM). Normally, traffic is carried on the working path (that is, the working path is the active path), and the protection path is disabled. If the working path fails, its protection status is marked as degraded (DG) and APS switches the traffic to the protection path, then the protection path becomes the active path.

APS uses two modes of operation, linear 1+1 protection switching architecture and linear 1:1 protection switching architecture. The linear 1+1 protection switching architecture operates with either unidirectional or bidirectional switching. The linear 1:1 protection switching architecture operates with bidirectional switching.

In the linear 1+1 protection switching architecture, the normal traffic is copied and fed to both working and protection paths with a permanent bridge at the source of the protected domain. The traffic on the working and protection transport entities is transmitted simultaneously to the sink of the protected domain, where a selection between the working and protection transport entities is made.

In the linear 1:1 protection switching architecture, the normal traffic is transported on either the working path or on the protection path using a selector bridge at the source of the protection domain. The selector at the sink of the protected domain selects the entity that carries the normal traffic.

Unidirectional and Bidirectional Switching

Unidirectional switching utilizes fully independent selectors at each end of the protected domain. Bidirectional switching attempts to configure the two end points with the same bridge and selector

settings, even for a unidirectional failure. Unidirectional switching can protect two unidirectional failures in opposite directions on different entities.

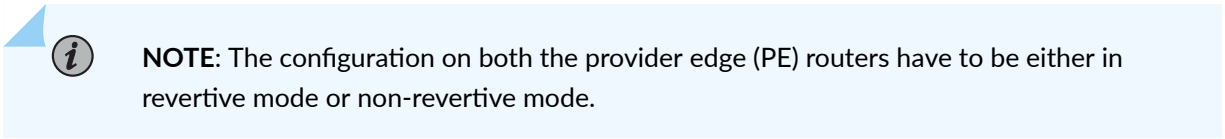
Selective and Merging Selectors

In the linear 1:1 protection switching architecture, where traffic is sent only on the active path, there are two different ways in which the egress direction (the direction out of the protected segment) data forwarding can act: selective selectors and merging selectors. A selective selector forwards only traffic that is received from both the paths regardless of which one is currently active. In other words, with a merging selector the selection of the currently active path only affects the ingress direction. Merging selectors minimize the traffic loss during a protection switch, but they do not guarantee the delivery of the data packets in order.

Revertive and Nonrevertive Switching

For revertive switching, traffic is restored to the working path after the conditions causing the switch have cleared.

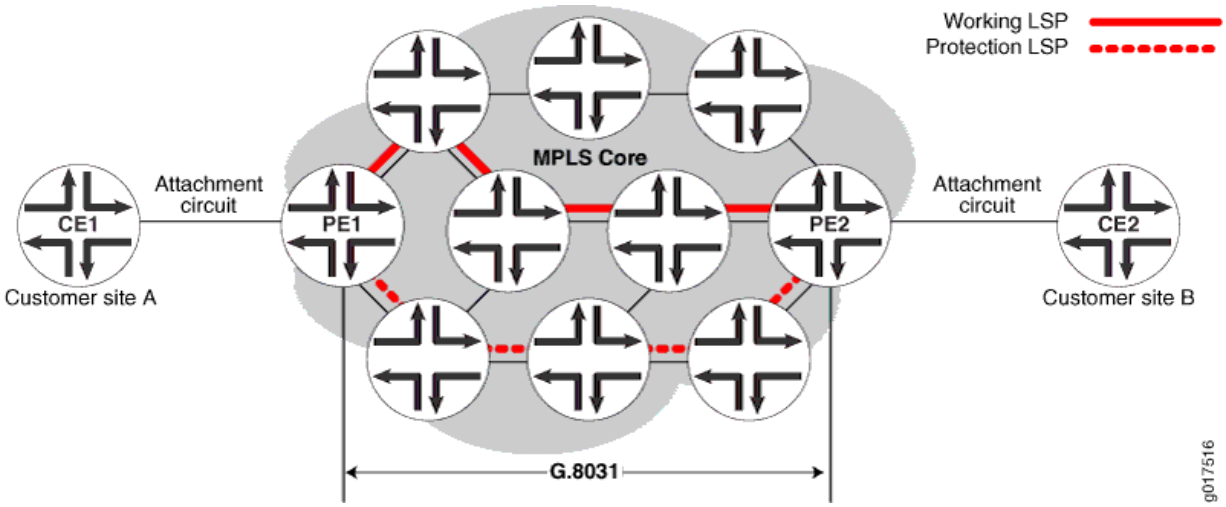
For nonrevertive switching, traffic is allowed to remain on the protection path even after the conditions causing the switch have cleared.



NOTE: The configuration on both the provider edge (PE) routers have to be either in revertive mode or non-revertive mode.

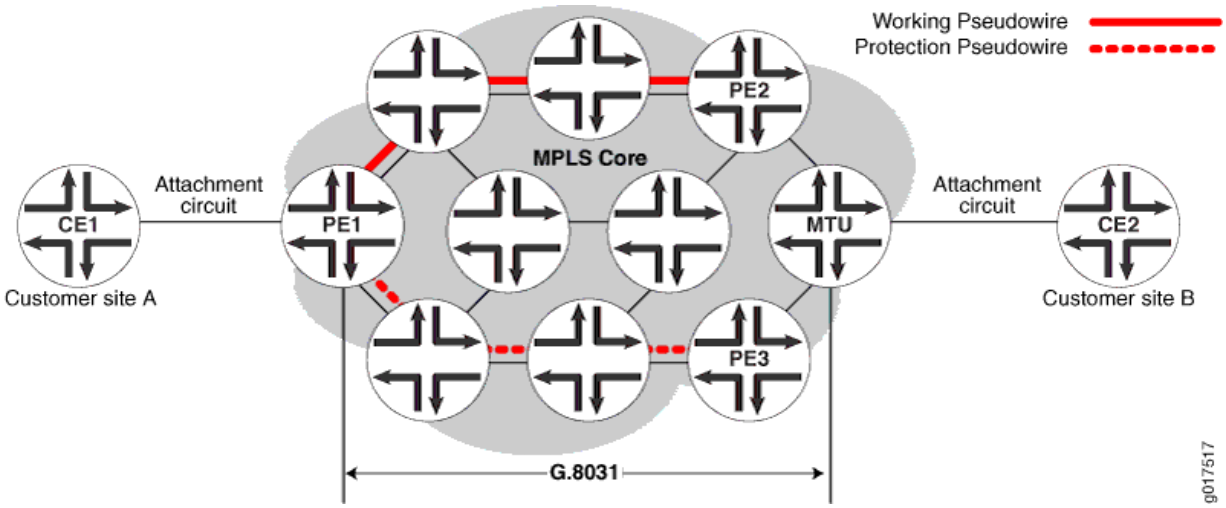
Protection Switching Between VPWS Pseudowires

Figure 11: Connections Terminating on Single PE



In the scenario diagramed in Figure 11 on page 221, a *Virtual Private Wire Service* (*VPWS*) is provisioned between customer sites A and B using a single pseudowire (layer 2 circuit) in the core network, and two Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) are provisioned, one for the working path and the other one for the protection path. CFM CCM will be used to monitor the status of each LSP. Provider edge routers PE1 and PE2 run G.8031 Ethernet APS to select one of the LSPs as the active path. Once the active path is elected at the source end of the protection group, PE1 forwards to traffic from site A to the elected active path. At the sink end of the protection group, PE2 implements a merging selector, meaning it forwards the traffic coming from both the LSPs to the customer site B.

Figure 12: Connections Terminating on a Different PE



In the scenario represented in Figure 12 on page 222, a VPWS is provisioned between customer sites A and B using two pseudowires (layer 2 circuit) in the core network, one for the working path and the other for the protection path. CFM CCM will be used to monitor the status of each pseudowire.

Provider edge router PE1 and MTU run G.8031 Ethernet APS to select one of the pseudowires as the active path. Once the active path is elected at the source end of the protection group, PE1 forwards the traffic from site A to the elected active path. At the sink end of the protection group, MTU implements a merging selector, meaning it forwards the traffic coming from both the pseudowires to customer site B.

CLI Configuration Statements

```
[edit protocols protection-group]
ethernet-aps profile1{
    protocol g8031;
    revert-time seconds;
    hold-time 0-10000ms;
    local-request lockout;
}
```

`revert-time`- By default, protection logic restores the use of the working path once it recovers. The revert-time statement specifies how much time should elapse before the path for data should be switched from Protection to Working once recovery for Working has occurred. A revert-time of zero indicates no reversion. It will default to 300 sec (5 minutes) if not configured.

`hold-time`- Once a failure is detected, APS waits until this timer expires before initiating the protection switch. The range of the hold-time timer is 0 to 10,000 milliseconds. It will default to zero if not configured.

`local-request`- Configuring this value to lockout or force-switch will trigger lockout or force-switch operation on the protection groups using this profile.

Mapping of CCM Defects to APS Events

The continuity check message (CCM) engine marks the status of working and protected transport entities as either Down, Degraded, or Up.

`Down`—The monitored path is declared down if any of the following Multiple End Point (MEP) defects occur:

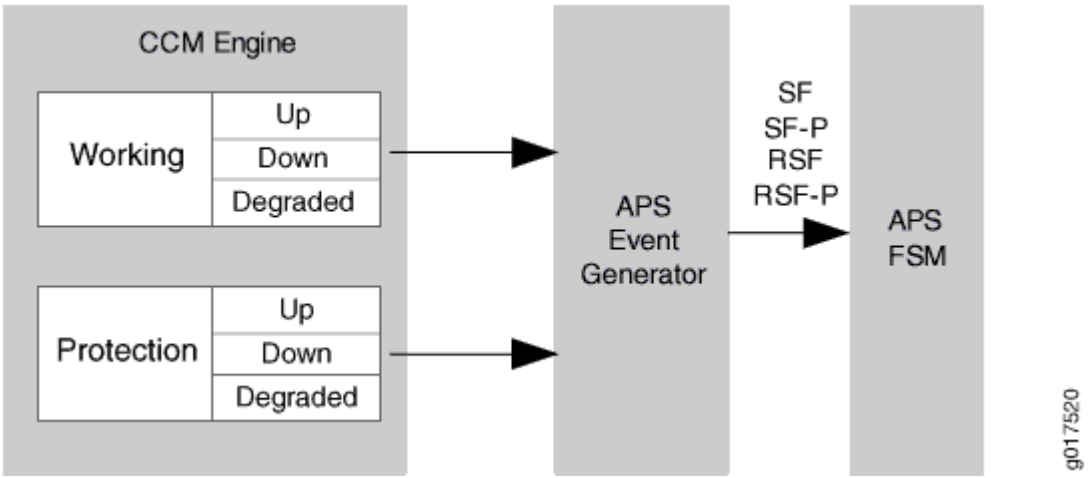
- Interface down
- CCM expiry
- RDI indicating signal failure

`Degraded`—The monitored path is declared degraded if any of the following MEP defects occur:

- FRR on
- FRR-ACK on

`Up`—The monitored path is declared up in the absence of any of the above events.

Figure 13: Understanding APS Events



As show in Figure 13 on page 223, the APS event generator generates the following APS events based on the status of the working and protection paths:

- SF—Signal failure on working path
- RSF—Working path recovers from signal failure
- SF-P—Signal failure on protection path
- RSF-P—Protection path recovers from signal failure

Example: Configuring Protection Switching Between Psuedowires

IN THIS SECTION

Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.2 or later
- 2 MX Series PE routers

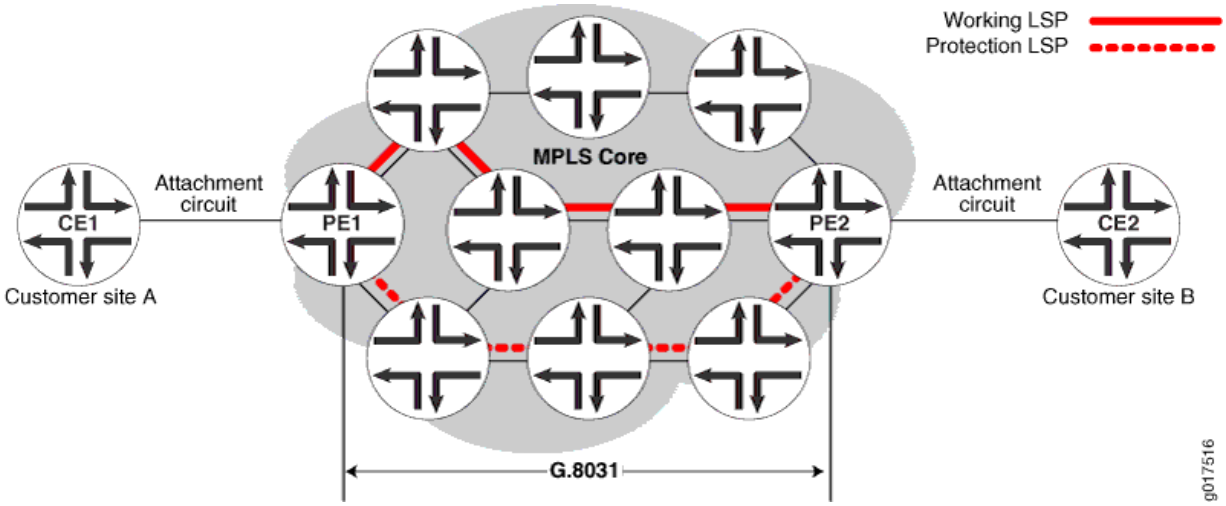
Overview and Topology

IN THIS SECTION

The physical topology of the protection switching between psuedowires example is shown in Figure 14 on page 225.

Topology

Figure 14: Topology of a Network Using VPWS Psuedowires



The following definitions describe the meaning of the device abbreviations used in Figure 14 on page 225.

- Customer edge (CE) device—A device at the customer site that provides access to the service provider's VPN over a data link to one or more provider edge (PE) routers.
- Provider edge (PE) device—A device, or set of devices, at the edge of the provider network that presents the provider's view of the customer site.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure protection switching between psuedowires, perform these tasks:

1. Configure automatic protection switching.

```
protocols {
    protection-group {
        ethernet-aps {
            profile-1 {
                protocol g8031;
                hold-time 1000s;
                revert-time 5m;
            }
        }
    }
```

2. Configure the connectivity fault management.

```
ethernet {
    oam {
        connectivity-fault-management {
            maintenance-domain md1 {
            level 5;
```

3. Configure the continuity check message for the working path.

```
maintenance-association W {
    protect maintenance-association P {
        aps-profile profile-1;
    }
    continuity-check {
        interval 1s;
    }
    mep 100 {
        interface ge-1/0/0.0 working;
        direction down;
        auto-discovery;
    }
}
```

4. Configure the continuity check message for the protection path.

```
maintenance-association P {
    continuity-check {
        interval 1s;
    }
    mep 100 {
        interface ge-1/0/0.0 protect;
        direction down;
        auto-discovery;
    }
}
```

Results

Check the results of the configuration:

```
protocols {
    protection-group {
        ethernet-aps {
            profile-1 {
                protocol g8031;
                hold-time 1000s;
                revert-time 5m;
            }
        }
    }
    ethernet {
        oam {
            connectivity-fault-management {
                maintenance-domain md1 {
                    level 5;
                    maintenance-association W {
                        protect maintenance-association P {
                            aps-profile profile-1;
                        }
                        continuity-check {
                            interval 1s;
                        }
                        mep 100 {
                            interface ge-1/0/0.0 working;
```

```
                        direction down;
                        auto-discovery;
                    }
                }
                maintenance-association P {
                    continuity-check {
                        interval 1s;
                    }
                    mep 100 {
                        interface ge-1/0/0.0 protect;
                        direction down;
                        auto-discovery;
                    }
                }
            }
        }
    }
}
```

PART 7

Configuring Ethernet Ring Protection Switching

Understanding Ethernet Ring Protection Switching

SUMMARY

Ethernet ring protection switching (ERPS) helps to prevent fatal loops from disrupting a network. ERPS is similar to spanning-tree protocols, but ERPS is more efficient because it is customized for ring topologies.

IN THIS SECTION

Ethernet Ring Protection Switching Overview

Ethernet ring protection switching (ERPS) helps achieve high reliability and network stability. Links in the ring will never form loops that fatally affect the network operation and services availability. The basic idea of an Ethernet ring is to use one specific link to protect the whole ring. This special link is called a *ring protection link (RPL)*. If no failure happens in other links of the ring, the RPL blocks the traffic and is not used. The RPL is controlled by a special node called an *RPL owner*. There is only one RPL owner in a ring. The RPL owner is responsible for blocking traffic over the RPL. Under ring failure conditions, the RPL owner is responsible for unblocking traffic over the RPL. A ring failure results in protection switching of the RPL traffic. An automatic protection switching (APS) protocol is used to coordinate the protection actions over the ring. Protection switching blocks traffic on the failed link and unblocks the traffic on the RPL. When the failure clears, revertive protection switching blocks traffic over the RPL and unblocks traffic on the link on which the failure is cleared.

NOTE: ERPS on AE interfaces is not supported on ACX Series routers except on ACX5000 and ACX7100 Series routers.

The following standards provide detailed information on Ethernet ring protection switching:

- ITU-T Recommendation G.8032/Y.1344 version 1 and 2, *Ethernet Ring protection switching*. G.8032v1 supports a single ring topology and G.8032v2 supports multiple rings and ladder topology.

  All devices with Ethernet ring protection switching support G.8032v1. MX Series and ACX Series routers also support G.8032v2.

- ITU-T Y.1731, *OAM functions and mechanisms for Ethernet-based networks*

For additional information on configuring Ethernet ring protection switching on EX Series switches, see *Example: Configuring Ethernet Ring Protection Switching on EX Series Switches*.

Starting in Junos OS Evolved release 24.4R1, the ERPS feature support provided on the four platforms ACX7100-32C, ACX7100-48L, ACX7509, and ACX7024, includes:

- Ethernet ring uses one specific link to protect the whole ring called a ring protection link (RPL). The RPL is controlled by a special node called an RPL owner.
- A ring failure results in protection switching of the RPL traffic. An automatic protection switching (APS) protocol is used to coordinate the protection actions over the ring. Protection switching blocks traffic on the failed link and unblocks the traffic on the RPL.
- ITU-T Y.1731, OAM functions and mechanisms for Ethernet-based networks.

For additional information on configuring Ethernet ring protection switching on MX Series routers, see the *Layer 2 Configuration Guide* for a complete example of Ethernet rings and information about STP loop avoidance and prevention.

Understanding Ethernet Ring Protection Switching Functionality

IN THIS SECTION

Acronyms

The following acronyms are used in the discussion about Ethernet ring protection switching (ERPS):

- MA—Maintenance association
- MEP—Maintenance association end point
- OAM—Operations, administration, and management (Ethernet ring protection switching uses connectivity fault management daemon)
- FDB—MAC forwarding database
- STP—Spanning Tree Protocol
- RAPS—Ring automatic protection switching
- WTB—Wait to block. Note that WTB is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting on EX2300 and EX3400 switches has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect on EX2300 and EX3400 switches.
- WTR—Wait to restore. Note that on EX2300 and EX3400 switches only, the WTR configuration must be 5-12 minutes.
- RPL—Ring protection link

Ring Nodes

Multiple nodes are used to form a ring. There are two different node types:

- Normal node—The node has no special role on the ring.
- RPL owner node—The node owns the RPL and blocks or unblocks traffic over the RPL.

Ring Node States

The following are the different states for each node of a specific ring:

- init—Not a participant of a specific ring.
- idle—No failure on the ring; the node is performing normally. For a normal node, traffic is unblocked on both ring ports. For the RPL owner or RPL neighbor, traffic is blocked on the ring port that connects to the RPL and unblocked on the other ring port.
- protection—A failure occurred on the ring. For a normal node, traffic is blocked on the ring port that connects to the failing link and unblocked on working ring ports. For the RPL owner, traffic is unblocked on both ring ports if they connect to non-failure links.
- pending—The node is recovering from failure or its state after a `clear` command is used to remove the previous manual command. When a protection group is configured, the node enters the pending state. When a node is in pending state, the WTR or WTB timer will be running. All nodes are in pending state till WTR or WTB timer expiry.
- force switch—A force switch is issued. When a force switch is issued on a node in the ring all nodes in the ring will move into the force switch state.

  NOTE: EX2300 and EX3400 switches do not support force switch.

- manual switch—A manual switch is issued. When a manual switch is issued on a node in the ring all nodes in the ring will move into the manual switch state.

  NOTE: EX2300 and EX3400 switches do not support manual switch.

There can be only one RPL owner for each ring. The user configuration must guarantee this, because the APS protocol cannot check this.

Default Logging of Basic State Transitions on EX Series Switches

Starting with Junos OS Release 14.1X53-D15, EX Series switches automatically log basic state transitions for the ERPS protocol. Starting with Junos OS Release 18.2R1, EX2300 and EX3400 switches automatically log basic state transitions for the ERPS protocol. No configuration is required to initiate this logging. Basic state transitions include ERPS interface transitions from up to down, and down to up; and ERPS state transitions from idle to protection, and protection to idle.

The basic state transitions are logged in a single file named **erp-default**, which resides in the **/var/log** directory of the switch. The maximum size of this file is 15 MB.

Default logging for ERPS can capture initial ERPS interface and state transitions, which can help you troubleshoot issues that occur early in the ERPS protocol startup process. However, if more robust logging is needed, you can enable traceoptions for ERPS by entering the `traceoptions` statement in the `[edit protocols protection-group]` hierarchy.

Be aware that for ERPS, only default logging or traceoptions can be active at a time on the switch. That is, default logging for ERPS is automatically enabled and if you enable traceoptions for ERPS, the switch automatically disables default logging. Conversely, if you disable traceoptions for ERPS, the switch automatically enables default logging.

Logical Ring

You can define multiple logical-ring instances on the same physical ring. The logical ring feature currently supports only the physical ring, which means that two adjacent nodes of a ring must be physically connected and the ring must operate on the physical interface, not the VLAN. Multiple ring instances are usually defined with trunk mode ring interfaces.

FDB Flush

When ring protection switching occurs, normally an *FDB flush* is executed. The Ethernet ring control module uses the same mechanism as the STP to trigger the FDB flush. The Ethernet ring control module controls the ring port physical interface's default STP index to execute the FDB flush.

NOTE: Optimized flushing is not supported on EX2300 and EX3400 switches.

Starting with Junos OS Release 14.2, the FDB flush depends on the RAPS messages received on the both the ports of the ring node.

Traffic Blocking and Forwarding

Ethernet ring control uses the same mechanism as the STP to control forwarding or discarding of user traffic. The Ethernet ring control module sets the ring port physical interface default STP index state to forwarding or discarding in order to control user traffic.

RPL Neighbor Node

Starting with Junos OS Release 14.2, ring protection link neighbor nodes are supported. An RPL neighbor node is adjacent to the RPL and is not the RPL owner. If a node is configured with one interface as the protection-link-end and no protection-link-owner is present in its configuration, the node is an RPL neighbor node.

NOTE: RPL neighbor node is not supported on EX2300 and EX3400 switches.

RAPS Message Blocking and Forwarding

The router or switch treats the ring automatic protection switching (RAPS) message the same as it treats user traffic for forwarding RAPS messages between two ring ports. The ring port physical interface default STP index state also controls forwarding RAPS messages between the two ring ports. Other than forwarding RAPS messages between the two ring ports, as shown in Figure 15 on page 235, the system also needs to forward the RAPS message between the CPU (Ethernet ring control module) and the ring port. This type of forwarding does not depend on the ring port physical interfaces' STP index state. The RAPS message is always sent by the router or switch through the ring ports, as shown in Figure 16 on page 235. A RAPS message received from a discarding ring port is sent to the Ethernet ring control module, but is not sent to the other ring port.

Figure 15: Protocol Packets from the Network to the Router

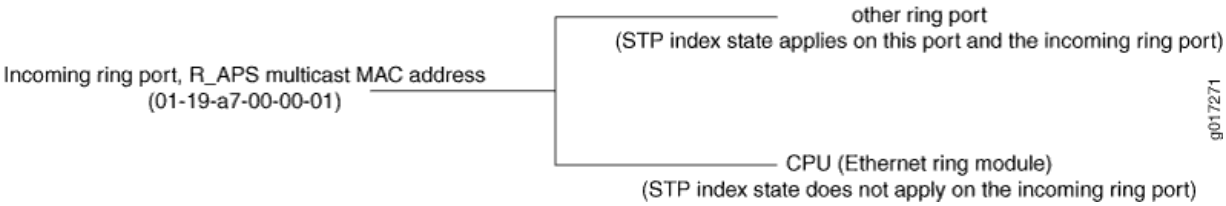
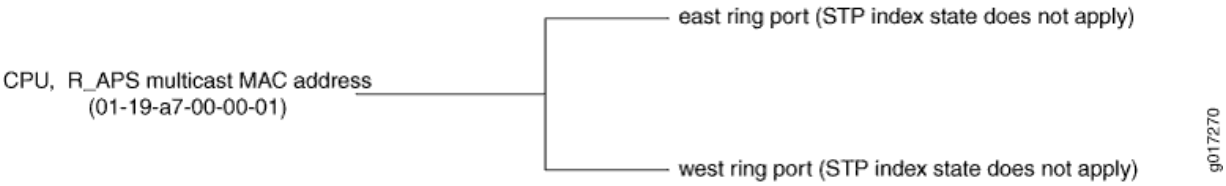


Figure 16: Protocol Packets from the Router or Switch to the Network



Juniper Networks switches and Juniper Networks routers use different methods to achieve these routes.

The switches use forwarding database entries to direct the RAPS messages. The forwarding database entry (keyed by the RAPS multicast address and VLAN) has a composite next hop associated with it—the composite next hop associates the two ring interfaces with the forwarding database entry and uses the split horizon feature to prevent sending the packet out on the interface that it is received on. This is an

example of the forwarding database entry relating to the RAPS multicast MAC (a result of the `show ethernet-switching table detail` command):

```
VLAN: v1, Tag: 101, MAC: 01:19:a7:00:00:01, Interface: ERP
Interfaces:                  ge-0/0/9.0, ge-0/0/3.0
Type: Static
Action: Mirror
Nexthop index: 1333
```

The routers use an implicit filter to achieve ERP routes. Each implicit filter binds to a bridge domain. Therefore, the east ring port control channel and the west ring port control channel of a particular ring instance must be configured to the same bridge domain. For each ring port control channel, a filter term is generated to control RAPS message forwarding. The filter number is the same as the number of bridge domains that contain the ring control channels. If a bridge domain contains control channels from multiple rings, the filter related to this bridge domain will have multiple terms and each term will relate to a control channel. The filter has command parts and control-channel related parts, as follows:

- Common terms:
  - ```
    term 1: if [Ethernet type is not OAM Ethernet type (0x8902)
    ]             { accept packet }
    ```
  - ```
    term 2: if [source MAC address belongs to this bridge]
                { drop packet, our packet loop through the ring and come back
                   to home}
    ```
  - ```
    term 3: if [destination is the RAPS PDU multicast address(0x01,0x19,0xa7,
                    0x00,0x00,0x01] AND[ring port STP status is DISCARDING]
                { send to CPU }
    ```
- Control channel related terms:
  - ```
          if [destination is the RAPS PDU multicast address(0x01,0x19,0xa7,0x00,0x00,
             0x01] AND[ring port STP status is FORWARDING] AND [Incoming interface
              IFL equal to control channel IFL]
             { send packet to CPU and send to the other ring port }
          default term: accept packet.
    ```

Dedicated Signaling Control Channel

For each ring port, a dedicated signaling control channel with a dedicated VLAN ID must be configured. In Ethernet ring configuration, only this control *logical interface* is configured and the underlying physical interface is the physical ring port. Each ring requires that two control physical interfaces be configured. These two logical interfaces must be configured in a bridge domain for routers (or the same VLAN for switches) in order to forward RAPS protocol data units (PDUs) between the two ring control physical interfaces. If the router control channel logical interface is not a trunk port, only control logical interfaces will be configured in ring port configuration. If this router control channel logical interface is a trunk port, in addition to the control channel logical interfaces, a dedicated VLAN ID must be configured for routers. For switches, always specify either a VLAN name or VLAN ID for all links.

RAPS Message Termination

The RAPS message starts from the originating node, travels through the entire ring, and terminates in the originating node unless a failure is present in the ring. The originating node must drop the RAPS message if the source MAC address in the RAPS message belongs to itself. The source MAC address is the node's node ID.

Revertive and Non-revertive Modes

In revertive operation, once the condition causing a switch has cleared, traffic is blocked on the RPL and restored to the working transport entity. In nonrevertive operation, traffic is allowed to use the RPL if it has not failed, even after a switch condition has cleared.

NOTE: Non-revertive mode is not supported on EX2300 and EX3400 switches.

Multiple Rings

The Ethernet ring control module supports multiple rings in each node (two logical interfaces are part of each ring). The ring control module also supports the interconnection of multiple rings. Interconnection of two rings means that two rings might share the same link or share the same node. Ring interconnection is supported only using non-virtual-channel mode. Ring interconnection using virtual channel mode is not supported.

NOTE: Interconnection of multiple rings is not supported on EX2300 and EX3400 switches.

Node ID

For each node in the ring, a unique *node ID* identifies each node. The node ID is the node's MAC address.

For routers only, you can configure this node ID when configuring the ring on the node or automatically select an ID like STP does. In most cases, you will not configure this and the router will select a node ID, like STP does. It should be the manufacturing MAC address. The ring node ID should not be changed, even if you change the manufacturing MAC address. Any MAC address can be used if you make sure each node in the ring has a different node ID. The node ID on switches is selected automatically and is not configurable.

Ring ID

The ring ID is used to determine the value of the last octet of the MAC destination address field of the RAPS protocol data units (PDUs) generated by the ERP control process. The ring ID is also used to discard any RAPS PDU, received by this ERP control process with a non-matching ring ID. Ring ID values 1 through 239 are supported.

Bridge Domains with the Ring Port (MX Series Routers Only)

On the routers, the protection group is seen as an abstract logical port that can be configured to any bridge domain. Therefore, if you configure one ring port or its logical interface in a bridge domain, you must configure the other related ring port or its logical interface to the same bridge domain. The bridge domain that includes the ring port acts as any other bridge domain and supports the IRB Layer 3 interface.

Wait-to-Block Timer

The RPL owner node uses a delay timer before initiating an RPL block in revertive mode of operation or before reverting to IDLE state after clearing manual commands. The Wait-to-Block (WTB) timer is used when clearing `force switch` and `manual switch` commands. As multiple `force switch` commands are allowed to coexist in an Ethernet ring, the WTB timer ensures that clearing of a single `force switch` command does not trigger the re-blocking of the RPL. When clearing a `manual switch` command, the WTB timer prevents the formation of a closed loop due to a possible timing anomaly where the RPL Owner Node receives an outdated remote `manual switch` request during the recovery process.

When recovering from a `manual switch` command, the delay timer must be long enough to receive any latent remote `force switch`, signal failure, or `manual switch` commands. This delay timer is called the WTB timer and is defined to be 5 seconds longer than the guard timer. This delay timer is activated on the RPL Owner Node. When the WTB timer expires, the RPL Owner Node initiates the reversion process by

transmitting an RAPS (NR, RB) message. The WTB timer is deactivated when any higher-priority request preempts it.

NOTE: The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.

Adding and Removing a Node

Starting with Junos OS Release 14.2, you can add or remove a node between two nodes in an Ethernet ring. Nodes are added or removed using the `force switch` command.

NOTE: EX2300 and EX3400 switches do not support force switch.

Change History Table

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

Release	Description
18.2R1	Starting with Junos OS Release 18.2R1, EX2300 and EX3400 switches automatically log basic state transitions for the ERPS protocol.
14.2	Starting with Junos OS Release 14.2, the FDB flush depends on the RAPS messages received on the both the ports of the ring node.
14.2	Starting with Junos OS Release 14.2, ring protection link neighbor nodes are supported.
14.2	Starting with Junos OS Release 14.2, you can add or remove a node between two nodes in an Ethernet ring.
14.1X53-D15	Starting with Junos OS Release 14.1X53-D15, EX Series switches automatically log basic state transitions for the ERPS protocol.

Configuring Ethernet Ring Protection Switching

SUMMARY

Follow the steps below to configure Ethernet ring protection switching (ERPS) on your device.

IN THIS SECTION

Configuring Ethernet Ring Protection Switching

The inheritance model follows:

```
[edit protocols]
protection-group {
    ethernet-ring ring-name (
        node-id mac-address;
        ring-protection-link-owner;
        east-interface {
            control-channel channel-name {
            ring-protection-link-end;
        }
        west-interface {
            node-id mac-address;
            control-channel channel-name {
            ring-protection-link-end;
        }
        data-channel {
            vlan number;
        }
        guard-interval number;
        restore-interval number;
    }
}
```

For each ring, a protection group must be configured. There may be several rings in each node, so there should be multiple protection groups corresponding to the related Ethernet rings.

Three interval parameters (`restore-interval`, `guard-interval`, and `hold-interval`) can be configured at the protection group level. These configurations are global configurations and apply to all Ethernet rings if the Ethernet ring doesn't have a more specific configuration for these values. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value.

Example: Ethernet Ring Protection Switching Configuration on MX Routers

IN THIS SECTION

This example describes how to configure Ethernet ring protection switching on an MX Series router:

Requirements

This example uses the following hardware and software components:

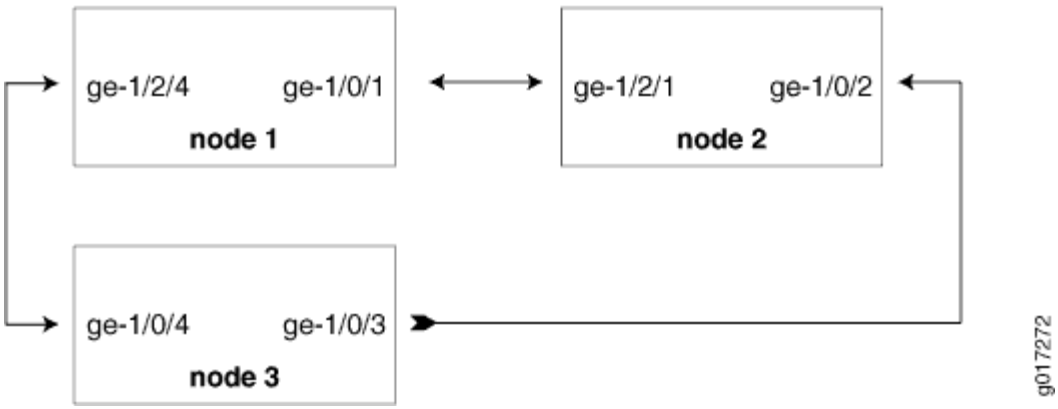
- Router node 1 running Junos OS with two Gigabit Ethernet interfaces.
- Router node 2 running Junos OS with two Gigabit Ethernet interfaces.
- Router node 3 running Junos OS with two Gigabit Ethernet interfaces.

Ethernet Ring Overview and Topology

IN THIS SECTION

This section describes a configuration example for a three-node ring. The ring topology is shown in Figure 17 on page 242.

Figure 17: Example of a Three-Node Ring Topology



Topology

The configuration in this section is only for the RAPS channel. The bridge domain for user traffic is the same as the normal bridge domain. The only exception is if a bridge domain includes a ring port, then it must also include the other ring port of the same ring.

Configuring a Three-Node Ring

IN THIS SECTION

To configure Ethernet Ring Protection Switching on a three-node ring, perform these tasks:

Configuring Ethernet Ring Protection Switching on a Three-Node Ring

Step-by-Step Procedure

1. Configuring Node 1

```
interfaces {
     ge-1/0/1 {
          vlan-tagging;
          encapsulation flexible-ethernet-services;
          unit 1 {
               encapsulation vlan-bridge;
               vlan-id 100;
          }
     }
     ge-1/2/4 {
          vlan-tagging;
          encapsulation flexible-ethernet-services;
          unit 1 {
               encapsulation vlan-bridge;
               vlan-id 100;
          }
     }
}
bridge-domains {
    bd1 {
         domain-type bridge;
         interface ge-1/2/4.1;
         interface ge-1/0/1.1;
     }
}
protocols {
    protection-group {
        ethernet-ring pg101 {
        node-id 00:01:01:00:00:01;
        ring-protection-link-owner;
            east-interface {
                 control-channel ge-1/0/1.1;
                 ring-protection-link-end;
            }
            west-interface {
                control-channel ge-1/2/4.1;
```

```
            }
        }
    }
}
protocols {
    oam {
          ethernet {
              connectivity-fault-management {
                  action-profile rmep-defaults {
                      default-action {
                           interface-down;
                      }
                  }

                  maintenance-domain d1 {
                       level 0;
                       maintenance-association 100 {
                            mep 1 {
                                interface ge-1/0/1;
                                remote-mep 2 {
                                    action-profile rmep-defaults;
                                }
                            }
                       }
                  }
                  maintenance-domain d2 {
                      level 0;
                      maintenance-association 100 {
                          mep 1 {
                               interface ge-1/2/4;
                               remote-mep 2 {
                                   action-profile rmep-defaults;
                                }
                           }
                      }
                  }
              }
        }
    }
}
```

2. Configuring Node 2

```
interfaces {
    ge-1/0/2 {
        vlan-tagging;
        encapsulation flexible-ethernet-services;
        unit 1 {
             encapsulation vlan-bridge;
             vlan-id 100;
        }
    }

    ge-1/2/1 {
        vlan-tagging;
        encapsulation flexible-ethernet-services;
        unit 1 {
            encapsulation vlan-bridge;
            vlan-id 100;
        }
    }
}

bridge-domains {
    bd1 {
        domain-type bridge;
        interface ge-1/2/1.1;
        interface ge-1/0/2.1;
    }
}


protocols {
    protection-group {
        ethernet-ring pg102 {
            east-interface {
                control-channel ge-1/0/2.1;
            }
            west-interface {
                control-channel ge-1/2/1.1;
            }
        }
    }
```

```
}


protocols {
    oam {
        ethernet {
            connectivity-fault-management {
                action-profile rmep-defaults {
                    default-action {
                        interface-down;
                    }
                }
                maintenance-domain d1 {
                    level 0;
                    maintenance-association 100 {
                        mep 2 {
                            interface ge-1/2/1;
                            remote-mep 1 {
                                action-profile rmep-defaults;
                            }
                        }
                    }
                }
                maintenance-domain d3 {
                    level 0;
                    maintenance-association 100 {
                        mep 1 {
                            interface ge-1/0/2;
                            remote-mep 2 {
                                action-profile rmep-defaults;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

3. Configuring Node 3

```
interfaces {
    ge-1/0/4 {
        vlan-tagging;
        encapsulation flexible-ethernet-services;
        unit 1 {
             encapsulation vlan-bridge;
             vlan-id 100;
        }
    }

    ge-1/0/3 {
        vlan-tagging;
        encapsulation flexible-ethernet-services;
        unit 1 {
            encapsulation vlan-bridge;
            vlan-id 100;
        }
    }
}

bridge-domains {
    bd1 {
        domain-type bridge;
        interface ge-1/0/4.1;
        interface ge-1/0/3.1;
    }
}


protocols {
    protection-group {
        ethernet-ring pg103 {
            east-interface {
                control-channel ge-1/0/3.1;
            }
            west-interface {
                control-channel ge-1/0/4.1;
            }
        }
    }
```

```
}


protocols {
    oam {
        ethernet {
            connectivity-fault-management {
                action-profile rmep-defaults {
                    default-action {
                        interface-down;
                    }
                }
                maintenance-domain d2 {
                    level 0;
                    maintenance-association 100 {
                        mep 2 {
                            interface ge-1/0/4;
                            remote-mep 1 {
                                action-profile rmep-defaults;
                            }
                        }
                    }
                }
                maintenance-domain d3 {
                    level 0;
                    maintenance-association 100 {
                        mep 2 {
                            interface ge-1/0/3;
                            remote-mep 1 {
                                action-profile rmep-defaults;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Examples: Ethernet RPS Output

This section provides output examples based on the configuration shown in "Example: Ethernet Ring Protection Switching Configuration on MX Routers" on page 241. The show commands used in these examples can help verify configuration and correct operation.

Normal Situation—RPL Owner Node

If the ring has no failure, the `show` command will have the following output for Node 1:

```
user@node1> show protection-group ethernet-ring aps

Ethernet Ring Name  Request/state  No Flush  Ring Protection Link Blocked
pg101               NR             No        Yes

Originator  Remote Node ID
Yes

user@node1> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg101

Interface    Control Channel  Forward State  Ring Protection Link End
ge-1/0/1     ge-1/0/1.1       discarding     Yes
ge-1/2/4     ge-1/2/4.1       forwarding     No

Signal Failure  Admin State
Clear           IFF ready
Clear           IFF ready

user@node1> show protection-group ethernet-ring node-state
Ethernet ring    APS State    Event         Ring Protection Link Owner
pg101            idle         NR-RB         Yes

Restore Timer  Quard Timer  Operation state
disabled       disabled    operational
user@node1> show protection-group ethernet-ring statistics group-name pg101
Ethernet Ring statistics for PG pg101
RAPS sent                       : 1
RAPS received                   : 0
Local SF happened:              : 0
Remote SF happened:             : 0
```

```
NR event happened:               : 0
NR-RB event happened:            : 1
```

Normal Situation—Other Nodes

For Node 2 and Node 3, the outputs should be the same:

```
user@node2> show protection-group ethernet-ring aps
Ethernet Ring Name  Request/state  No Flush  Ring Protection Link Blocked
pg102               NR             No        Yes

Originator  Remote Node ID
No          00:01:01:00:00:01

user@node2> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg102

Interface    Control Channel  Forward State  Ring Protection Link End
ge-1/2/1     ge-1/2/1.1       forwarding     No
ge-1/0/2     ge-1/0/2.1       forwarding     No

Signal Failure  Admin State
Clear           IFF ready
Clear           IFF ready

user@node2> show protection-group ethernet-ring node-state
Ethernet ring    APS State    Event         Ring Protection Link Owner
pg102              idle         NR-RB         No

Restore Timer  Quard Timer  Operation state
disabled       disabled    operational

user@node2> show protection-group ethernet-ring statistics group-name pg102
Ethernet Ring statistics for PG pg101
RAPS sent                        : 0
RAPS received                    : 1
Local SF happened:               : 0
Remote SF happened:              : 0
NR event happened:               : 0
NR-RB event happened:            : 1
```

Failure Situation—RPL Owner Node

If the ring has a link failure between Node 2 and Node 3, the `show` command will have the following outputs for Node 1:

```
user@node1> show protection-group ethernet-ring aps
Ethernet Ring Name  Request/state  No Flush  Ring Protection Link Blocked
pg101               SF             NO        No

Originator  Remote Node ID
No          00:01:02:00:00:01

user@node1> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg101

Interface    Control Channel  Forward State  Ring Protection Link End
ge-1/0/1     ge-1/0/1.1       forwarding     Yes
ge-1/2/4     ge-1/2/4.1       forwarding     No

Signal Failure  Admin State
Clear           IFF ready
Clear           IFF ready

user@node1> show protection-group ethernet-ring node-state
Ethernet ring    APS State    Event         Ring Protection Link Owner
pg101            protected    SF            Yes

Restore Timer  Quard Timer  Operation state
disabled       disabled     operational

user@node1> show protection-group ethernet-ring statistics group-name pg101
Ethernet Ring statistics for PG pg101
RAPS sent                        : 1
RAPS received                    : 1
Local SF happened:               : 0
Remote SF happened:              : 1
NR event happened:               : 0
NR-RB event happened:            : 1
```

Failure Situation—Other Nodes

For Node 2 and Node 3, the outputs should be the same:

```
user@node2> show protection-group ethernet-ring aps
Ethernet Ring Name  Request/state  No Flush  Ring Protection Link Blocked
pg102               SF             No        No

Originator  Remote Node ID
Yes         00:00:00:00:00:00

user@node2> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg102

Interface    Control Channel  Forward State  Ring Protection Link End
ge-1/2/1     ge-1/2/1.1       forwarding     No
ge-1/0/2     ge-1/0/2.1       discarding     No

Signal Failure  Admin State
Clear           IFF ready
set             IFF ready

user@node2> show protection-group ethernet-ring node-state
Ethernet ring    APS State    Event         Ring Protection Link Owner
pg102             idle         NR-RB         No

Restore Timer  Quard Timer  Operation state
disabled       disabled     operational

user@node2> show protection-group ethernet-ring statistics group-name pg102
Ethernet Ring statistics for PG pg101
RAPS sent                        : 1
RAPS received                    : 1
Local SF happened:               : 1
Remote SF happened:              : 0
NR event happened:               : 0
NR-RB event happened:            : 1
```

8

PART

Configuring Nonstop Bridging

Understanding Nonstop Bridging

SUMMARY

Nonstop bridging (NSB) helps preserve interface and kernel information on Routing Engine switchover, and synchronizes all protocol information for NSB-supported Layer 2 protocols between the primary and backup Routing Engines.

IN THIS SECTION

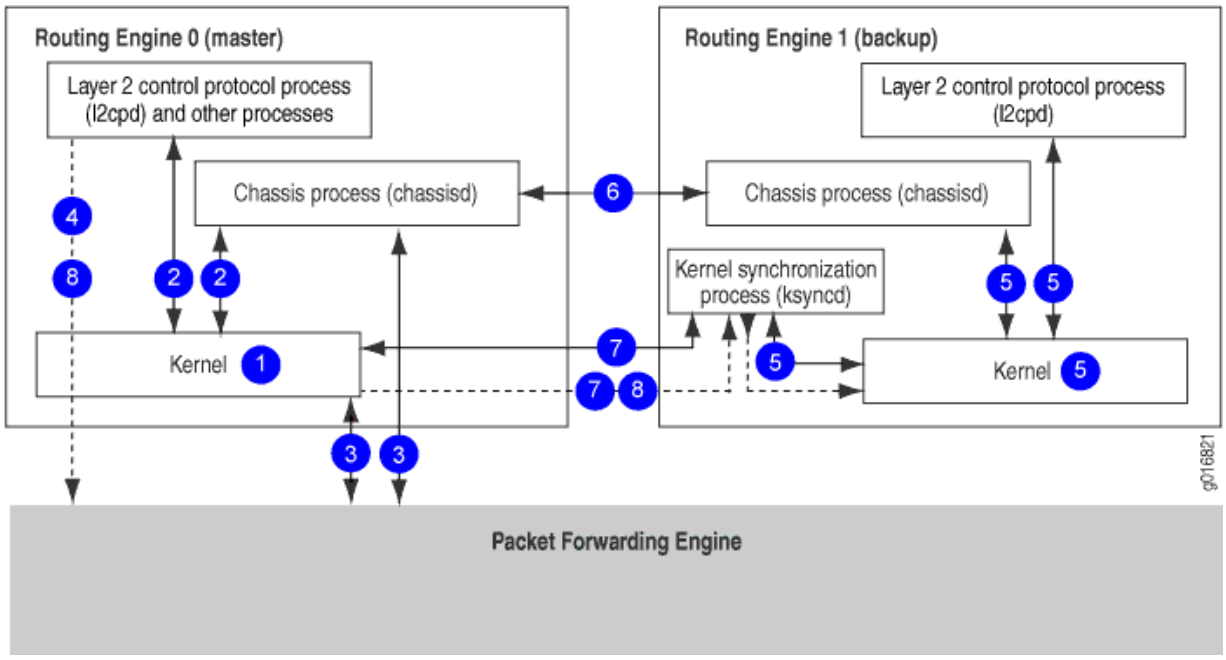
Nonstop Bridging Concepts

Nonstop bridging uses the same infrastructure as *graceful Routing Engine switchover* (GRES) to preserve interface and kernel information. However, nonstop bridging also saves Layer 2 Control Protocol (L2CP) information by running the Layer 2 Control Protocol process (l2cpd) on the backup Routing Engine.

NOTE: To use nonstop bridging, you must first enable graceful Routing Engine switchover on your routing (or switching) platform. For more information about graceful Routing Engine switchover, see Understanding Graceful Routing Engine Switchover.

Figure 18 on page 255 shows the system architecture of nonstop bridging and the process a routing (or switching) platform follows to prepare for a switchover.

Figure 18: Nonstop Bridging Switchover Preparation Process

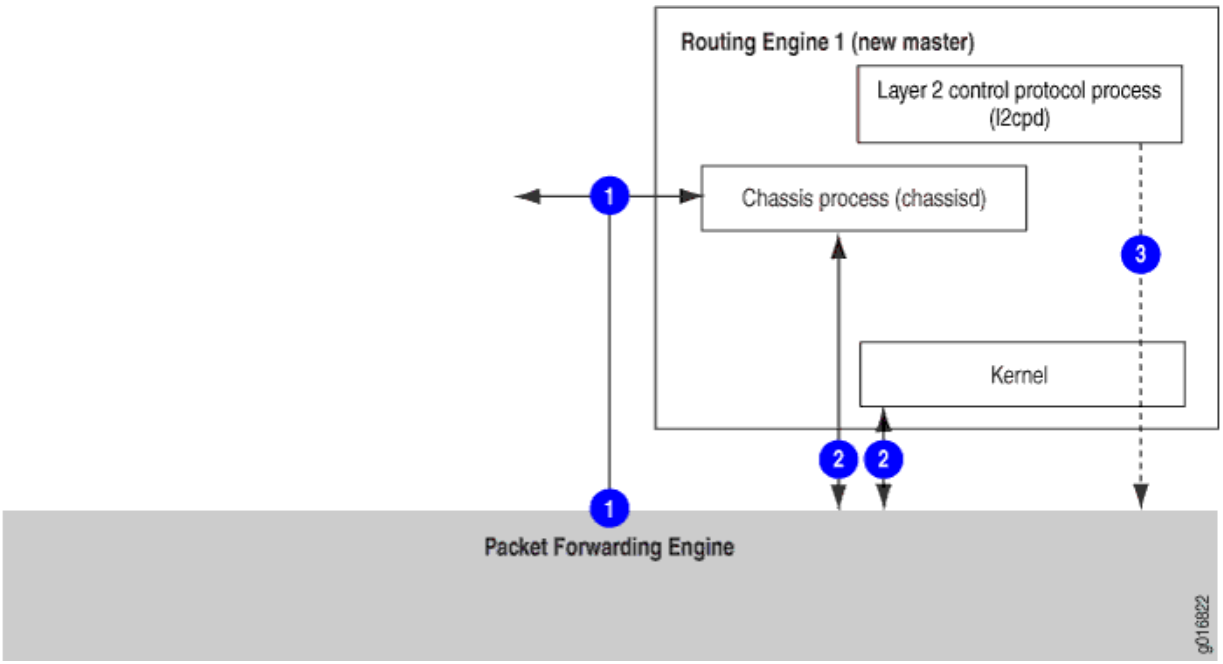


The switchover preparation process for nonstop bridging follows these steps:

1. The primary Routing Engine starts.
2. The routing platform processes on the primary Routing Engine (such as the chassis process [chassisd] and the Layer 2 Control Protocol process [l2cpd]) start.
3. The Packet Forwarding Engine starts and connects to the primary Routing Engine.
4. All state information is updated in the system.
5. The backup Routing Engine starts, including the chassis process (chassisd) and the Layer 2 Control Protocol process (l2cpd).
6. The system determines whether graceful Routing Engine switchover and nonstop bridging have been enabled.
7. The kernel synchronization process (ksyncd) synchronizes the backup Routing Engine with the primary Routing Engine.
8. For supported protocols, state information is updated directly between the l2cpds on the primary and backup Routing Engines.

Figure 19 on page 256 shows the effects of a switchover on the routing platform.

Figure 19: Nonstop Bridging During a Switchover



The switchover process follows these steps:

1. When keepalives from the primary Routing Engine are lost, the system switches over gracefully to the backup Routing Engine.
2. The Packet Forwarding Engine connects to the backup Routing Engine, which becomes the new primary. Because the Layer 2 Control Protocol process (l2cpd) and chassis process (chassisd) are already running, these processes do not need to restart.
3. State information learned from the point of the switchover is updated in the system. Forwarding and bridging are continued during the switchover, resulting in minimal packet loss.

Understanding Nonstop Bridging on EX Series Switches

You can configure nonstop bridging (NSB) to provide resilience for Layer 2 protocol sessions on a Juniper Networks EX Series Ethernet Switch or on an EX Series *Virtual Chassis* with redundant Routing Engines.

NSB operates by synchronizing all protocol information for NSB-supported Layer 2 protocols between the primary and backup Routing Engines. If the switch has a Routing Engine switchover, the NSB-supported Layer 2 protocol sessions remain active because all session information is already synchronized to the backup Routing Engine. Traffic disruption for the NSB-supported Layer 2 protocol is minimal or nonexistent as a result of the switchover. The Routing Engine switchover is transparent to

neighbor devices, which do not detect any changes related to the NSB-supported Layer 2 protocol sessions on the switch.

For a list of the EX Series switches and Layer 2 protocols that support NSB, see EX Series Switch Software Features Overview and EX Series Virtual Chassis Software Features Overview.

NOTE: Nonstop bridging provides a transparent switchover mechanism only for Layer 2 protocol sessions. *Nonstop active routing* (NSR) provides a similar mechanism for Layer 3 protocol sessions.

Nonstop Bridging System Requirements

IN THIS SECTION

This topic contains the following sections:

Platform Support

Nonstop bridging is supported on MX Series 5G Universal Routing Platforms. Your system must be running Junos OS Release 8.4 or later.

Nonstop bridging is supported on EX Series switches with redundant Routing Engines in a Virtual Chassis or in a Virtual Chassis Fabric.

Nonstop bridging is supported on QFX Series switches in a Virtual Chassis or in a Virtual Chassis Fabric.

For a list of the EX Series switches and Layer 2 protocols that support nonstop bridging, see EX Series Switch Software Features Overview.

NOTE: All Routing Engines configured for nonstop bridging must be running the same Junos OS release.

Protocol Support

Nonstop bridging is supported for the following Layer 2 control protocols:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Multiple Spanning Tree Protocol (MSTP)
- VLAN Spanning Tree Protocol (VSTP)

RELATED DOCUMENTATION

Configuring Nonstop Bridging | 258

Configuring Nonstop Bridging

SUMMARY

You can configure nonstop bridging by following the steps below.

IN THIS SECTION

Enabling Nonstop Bridging

Nonstop bridging requires you to configure graceful Routing Engine switchover (GRES). To enable graceful Routing Engine switchover, include the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level:

```
[edit chassis redundancy]
graceful-switchover;
```

By default, nonstop bridging is disabled. To enable nonstop bridging, include the `nonstop-bridging` statement at the `[edit protocols layer2-control]` hierarchy level:

```
[edit protocols layer2-control]
nonstop-bridging;
```

To disable nonstop active routing, remove the `nonstop-bridging` statement from the `[edit protocols layer2-control]` hierarchy level.

Synchronizing the Routing Engine Configuration

When you configure nonstop bridging, you must also include the `commit synchronize` statement at the `[edit system]` hierarchy level so that, by default, when you issue the `commit` command, the configuration changes are synchronized on both Routing Engines. If you issue the `commit synchronize` command at the `[edit]` hierarchy level on the backup Routing Engine, the Junos OS displays a warning and commits the candidate configuration.

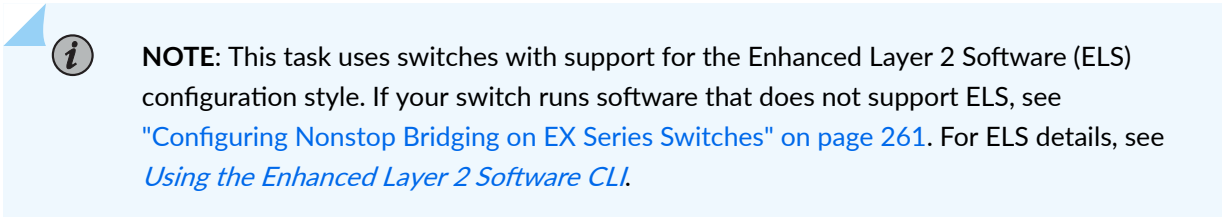
NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the primary Routing Engine configuration.

When you configure nonstop bridging, you can bring the backup Routing Engine online after the primary Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

Verifying Nonstop Bridging Operation

When you enable nonstop bridging, you can issue Layer 2 Control Protocol-related operational mode commands on the backup Routing Engine. However, the output of the commands might not match the output of the same commands issued on the primary Routing Engine.

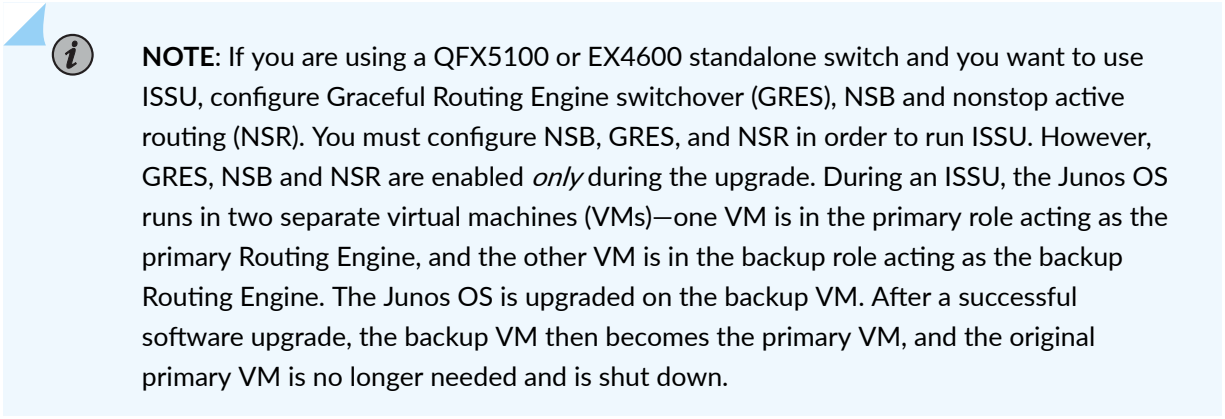
Configuring Nonstop Bridging on Switches (CLI Procedure)



NOTE: This task uses switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Nonstop Bridging on EX Series Switches" on page 261. For ELS details, see *Using the Enhanced Layer 2 Software CLI*.

You can configure nonstop bridging (NSB) to provide resilience for Layer 2 protocol sessions on a Juniper Networks EX Series switch with multiple Routing Engines or an EX Series or QFX Series switch in a Virtual Chassis or Virtual Chassis Fabric configuration. Limited support for NSB is also provided on QFX5100 and EX4600 standalone switches, but NSB is enabled *only* during an ISSU.

NSB operates by synchronizing all protocol information for NSB-supported Layer 2 protocols between the primary and backup Routing Engines. If the switch has a Routing Engine switchover, the NSB-supported Layer 2 protocol sessions remain active because they are already synchronized on the backup Routing Engine. The Routing Engine switchover is transparent to neighbor devices, which do not detect any changes related to the Layer 2 protocol sessions. The neighboring devices and other devices on the network do not, therefore, have to resynchronize their Layer 2 protocol states to respond to the downtime on the switch—a process that adds network overhead and risks disrupting network performance—when a Routing Engine switchover occurs when NSB is enabled.



NOTE: If you are using a QFX5100 or EX4600 standalone switch and you want to use ISSU, configure Graceful Routing Engine switchover (GRES), NSB and nonstop active routing (NSR). You must configure NSB, GRES, and NSR in order to run ISSU. However, GRES, NSB and NSR are enabled *only* during the upgrade. During an ISSU, the Junos OS runs in two separate virtual machines (VMs)—one VM is in the primary role acting as the primary Routing Engine, and the other VM is in the backup role acting as the backup Routing Engine. The Junos OS is upgraded on the backup VM. After a successful software upgrade, the backup VM then becomes the primary VM, and the original primary VM is no longer needed and is shut down.

To configure NSB:

1. Enable graceful Routing Engine switchover (GRES):

```
[edit chassis redundancy]
user@switch# set graceful-switchover
```

2. Enable NSB:

```
[edit protocols layer2-control]
user@switch# set nonstop-bridging
```

3. Synchronize configuration changes between the Routing Engines:

```
[edit system]
user@switch# set commit synchronize
```

   If you try to commit a configuration that includes NSB without including the `commit synchronize` statement, the commit fails.

NOTE: There is no requirement to start the two Routing Engines simultaneously. If the backup Routing Engine is not up when you use the `commit synchronize` statement, the candidate configuration is committed in the primary Routing Engine. When the backup Routing Engine comes online, its configuration is automatically synchronized with that of the primary.

BEST PRACTICE: After a graceful Routing Engine switchover, we recommend that you issue the `clear interface statistics (`*`interface-name`*` | all)` command to reset the cumulative values for local statistics on the new primary Routing Engine.

Configuring Nonstop Bridging on EX Series Switches (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring Nonstop Bridging on Switches" on page 260.

You can configure nonstop bridging (NSB) to provide resilience for Layer 2 protocol sessions on an EX Series switch with redundant Routing Engines.

Nonstop bridging operates by synchronizing all protocol information for NSB-supported Layer 2 protocols between the primary and backup Routing Engines. If the switch has a Routing Engine switchover, the NSB-supported Layer 2 protocol sessions remain active because they are already synchronized on the backup Routing Engine. The Routing Engine switchover is transparent to neighbor devices, which do not detect any changes related to the Layer 2 protocol sessions on the switch.

To configure nonstop bridging:

1. Enable graceful Routing Engine switchover (GRES):

```
[edit chassis redundancy]
user@switch# set graceful-switchover
```

2. Configure the switch to always synchronize configuration changes between the Routing Engines:

```
[edit system]
user@switch# set commit synchronize
```

   If you try to commit a configuration in which nonstop bridging is configured but synchronization of configuration changes is not configured, the configuration is not committed.

3. Enable nonstop bridging:

```
[edit ethernet-switching-options]
user@switch# set nonstop-bridging
```

NOTE: There is no requirement to start both Routing Engines simultaneously. If the backup Routing Engine is not up when you commit the configuration, the candidate configuration is committed in the primary Routing Engine. When the backup Routing Engine comes online, the configuration is automatically synchronized.

RELATED DOCUMENTATION

Understanding Nonstop Bridging | 254

nonstop-bridging

PART 9

Configuring Nonstop Active Routing (NSR)

Understanding Nonstop Active Routing

SUMMARY

Nonstop active routing (NSR) enables the transparent switchover of the Routing Engines in the event that one of the Routing Engines goes down.

IN THIS SECTION

Nonstop Active Routing Concepts

Nonstop active routing (NSR) uses the same infrastructure as *graceful Routing Engine switchover* (GRES) to preserve interface and kernel information. However, NSR also saves routing protocol information by running the routing protocol process (rpd) on the backup Routing Engine. By saving this additional information, NSR is self-contained and does not rely on helper routers (or switches) to assist the routing platform in restoring routing protocol information. NSR is advantageous in networks in which neighbor routers (or switches) do not support graceful restart protocol extensions. As a result of this enhanced functionality, NSR is a natural replacement for graceful restart.

Starting with Junos OS Release 15.1R1, if you have NSR configured, it is never valid to issue the `restart routing` command in any form on the NSR primary Routing Engine. Doing so results in a loss of protocol adjacencies and neighbors and a drop in traffic.

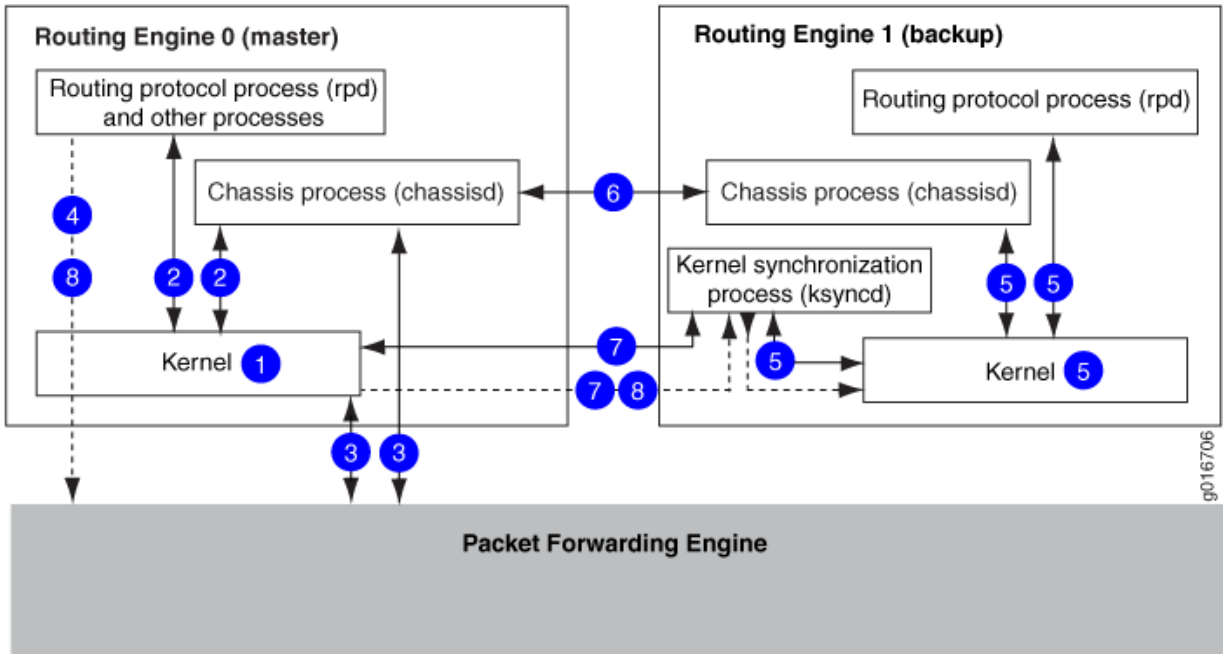
Use Feature Explorer to confirm platform and release support for specific features.

NOTE: To use NSR, you must first enable GRES on your routing (or switching) platform. For more information about GRES, see Understanding Graceful Routing Engine Switchover.

NOTE: If NSR is enabled, certain system log (syslog) messages are sent from the backup Routing Engine if the configured syslog host is reachable through the fxp0 interface.

Figure 20 on page 265 shows the system architecture of nonstop active routing and the process a routing (or switching) platform follows to prepare for a switchover.

Figure 20: Nonstop Active Routing Switchover Preparation Process

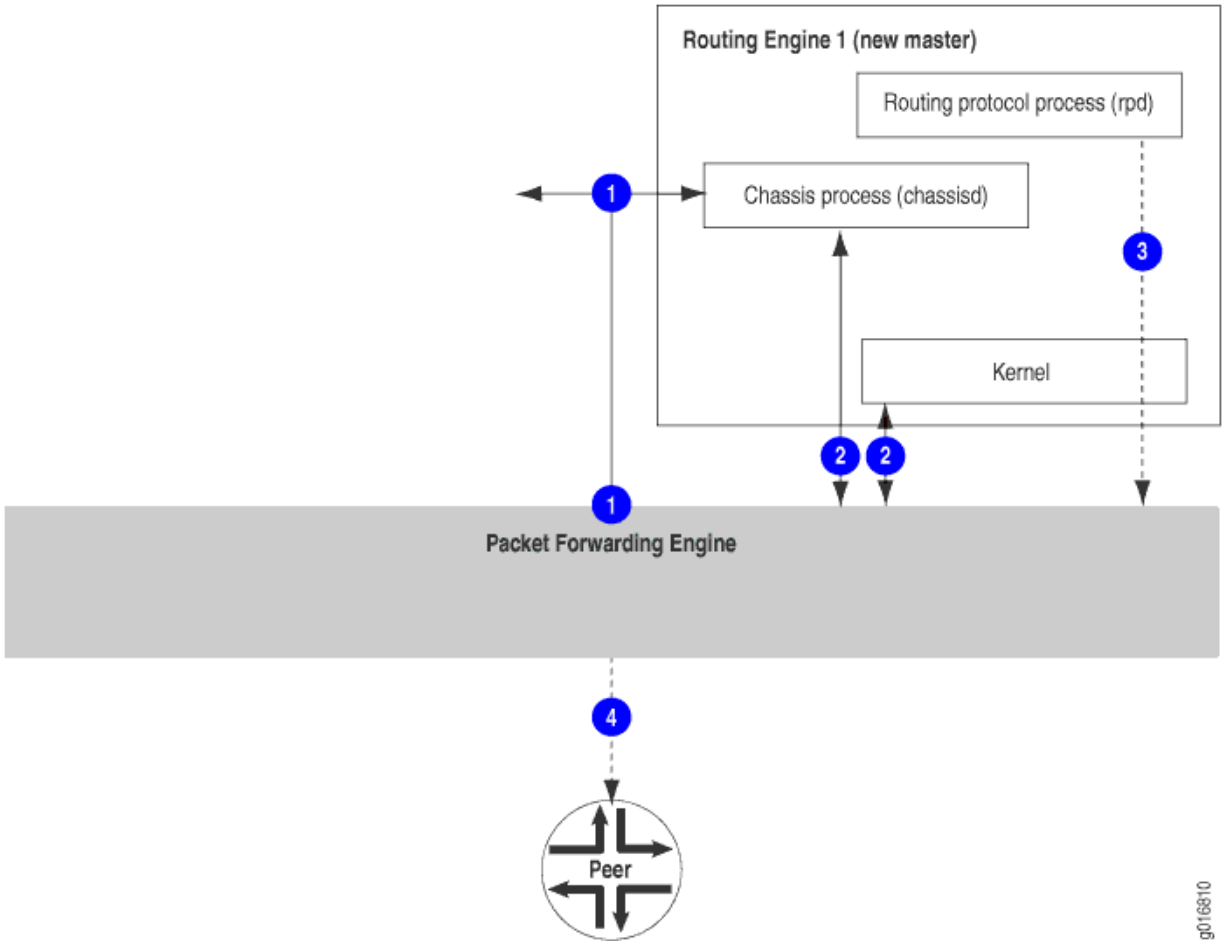


The switchover preparation process for NSR comprises the following steps:

1. The primary Routing Engine starts.
2. The routing (or switching) platform processes on the primary Routing Engine (such as the chassis process [chassisd] and the routing protocol process [rpd]) start.
3. The Packet Forwarding Engine starts and connects to the primary Routing Engine.
4. All state information is updated in the system.
5. The backup Routing Engine starts, including the chassis process (chassisd) and the routing protocol process (rpd).
6. The system determines whether GRES and NSR have been enabled.
7. The kernel synchronization process (ksyncd) synchronizes the backup Routing Engine with the primary Routing Engine.
8. For supported protocols, state information is updated directly between the routing protocol processes on the primary and backup Routing Engines.

Figure 21 on page 266 shows the effects of a switchover on the routing platform.

Figure 21: Nonstop Active Routing During a Switchover



The switchover process comprises the following steps:

1. When keepalives from the primary Routing Engine are lost, the system switches over gracefully to the backup Routing Engine.
2. The Packet Forwarding Engine connects to the backup Routing Engine, which becomes the new primary. Because the routing protocol process (rpd) and chassis process (chassisd) are already running, these processes do not need to restart.
3. State information learned from the point of the switchover is updated in the system. Forwarding and routing are continued during the switchover, resulting in minimal packet loss.
4. Peer routers (or switches) continue to interact with the routing platform as if no change had occurred. Routing adjacencies and session state relying on underlying routing information are preserved and not reset.

CAUTION: We recommend that you do not restart the routing protocol process (rpd) on primary Routing Engine after enabling NSR, as it disrupts the protocol adjacency/peering sessions, resulting in traffic loss.

SEE ALSO

Understanding High Availability Features on Juniper Networks Routers | 2

Nonstop Active Routing System Requirements

Configuring Nonstop Active Routing | 280

Configuring Nonstop Active Routing on Switches

Understanding Nonstop Active Routing on EX Series Switches

You can configure *nonstop active routing* (NSR) on an EX Series switch with redundant Routing Engines or on an EX Series *Virtual Chassis* to enable the transparent switchover of the Routing Engines in the event that one of the Routing Engines goes down.

Nonstop active routing provides high availability for Routing Engines by enabling transparent switchover of the Routing Engines without requiring restart of supported routing protocols. Both Routing Engines are fully active in processing protocol sessions, and so each can take over for the other. The switchover is transparent to neighbor routing devices, which do not detect that a change has occurred.

Enable nonstop active routing when neighbor routing devices are not configured to support graceful restart of protocols or when you want to ensure graceful restart of protocols for which graceful restart is not supported—such as PIM.

You do not need to start the two Routing Engines simultaneously to synchronize them for nonstop active routing. If both Routing Engines are not present or not up when you issue a `commit synchronize` statement, the candidate configuration is committed in the primary Routing Engine and when the backup Routing Engine is inserted or comes online, its configuration is automatically synchronized with that of the primary.

Nonstop active routing uses the same infrastructure as *graceful Routing Engine switchover* (GRES) to preserve interface and kernel information. However, nonstop active routing also saves routing protocol information by running the routing protocol process (**rpd**) on the backup Routing Engine. By saving this additional information, nonstop active routing does not rely on other routing devices to assist in restoring routing protocol information.

NOTE: After a graceful Routing Engine switchover, we recommend that you issue the `clear interface statistics (interface-name | all)` command to reset the cumulative values for local statistics on the new primary Routing Engine.

If you suspect a problem with the synchronization of Routing Engines when nonstop active routing is enabled, you can gather troubleshooting information using trace options. For example, if certain protocols lose connectivity with neighbors after a graceful Routing Engine switchover with NSR enabled, you can use trace options to help isolate the problem. See "Tracing Nonstop Active Routing Synchronization Events" on page 1186.

NOTE: Graceful restart and nonstop active routing are mutually exclusive. You will receive an error message upon commit if both are configured.

NOTE: Nonstop active routing provides a transparent switchover mechanism only for Layer 3 protocol sessions. Nonstop bridging (NSB) provides a similar mechanism for Layer 2 protocol sessions. See "Understanding Nonstop Bridging on EX Series Switches" on page 256.

SEE ALSO

Configuring Nonstop Active Routing on Switches

Example: Configuring Nonstop Active Routing on Switches | 288

Nonstop Active Routing System Requirements

IN THIS SECTION

This section contains the following topics:

Nonstop Active Routing Protocol and Feature Support

The following protocols are supported by nontop active routing:

- Aggregated Ethernet interfaces with Link Aggregation Control Protocol (LACP)
- Bidirectional Forwarding Detection (BFD)

  For more information, see "Nonstop Active Routing BFD Support" on page 272.
- BGP

  For more information, see "Nonstop Active Routing BGP Support" on page 272.
- EVPN
  - EVPN with ingress replication for BUM traffic
  - EVPN-ETREE
  - EVPN-VPWS
  - EVPN -VXLAN
  - PBB-EVPN
  - EVPN with P2MP mLDP replication for BUM traffic starting in Junos OS release 18.2R1

  For more information, please see *NSR and Unified ISSU Support for EVPN* .
- Labeled BGP (PTX Series Packet Transport Routers: only)
- IS-IS
- LDP
- LDP-based virtual private LAN service (VPLS)
- LDP OAM (operation, administration, and management) features
- LDP (PTX Series Packet Transport Routers only)

Nonstop active routing support for LDP includes:

- LDP unicast transit LSPs
- LDP egress LSPs for labeled internal BGP (IBGP) and external BGP (EBGP)
- LDP over RSVP transit LSPs
- LDP transit LSPs with indexed next hops
- LDP transit LSPs with unequal cost load balancing
- LDP Point-to-Multipoint LSPs
- LDP ingress LSPs

- Layer 2 circuits
- Layer 2 VPNs
- Layer 2 VPNs (PTX Series Packet Transport Routers only)

  NOTE: Nonstop active routing is not supported for Layer 2 interworking (Layer 2 stitching).

- Layer 3 VPNs (does not include dynamic GRE tunnels, multicast VPNs, or BGP flow routes.)

  Nonstop active routing support for Layer 3 VPNs include:

  - IPv4 labeled-unicast (ingress or egress)
  - IPv4-vpn unicast (ingress or egress)
  - IPv6 labeled-unicast (ingress or egress)
  - IPv6-vpn unicast (ingress or egress)

- Logical System support (Nonstop active routing support for logical systems to preserve interface and kernel information).
- Multicast Source Discovery Protocol (MSDP)

  For more information, see "Nonstop Active Routing MSDP Support" on page 277.

- OSPF/OSPFv3

NOTE: OSPFv3 neighbors enabled with IPSEC authentication are not supported with NSR.

- Protocol Independent Multicast (PIM)

  For more information, see "Nonstop Active Routing PIM Support" on page 274.

- RIP and RIP next generation (RIPng)
- RSVP (PTX Series Packet Transport Routers only)

  Nonstop active routing support for RSVP includes:

  - Point-to-Multipoint LSPs
    - RSVP Point-to-Multipoint ingress, transit, and egress LSPs using existing non-chained next hop.
    - RSVP Point-to-Multipoint transit LSPs using composite next hops for Point-to-Multipoint label routes.
  - Point-to-Point LSPs
    - RSVP Point-to-Point ingress, transit, and egress LSPs using non-chained next hops.
    - RSVP Point-to-Point transit LSPs using chained composite next hops.

- RSVP-TE LSP

  For more information, see "Nonstop Active Routing Support for RSVP-TE LSPs" on page 277.

- VPLS
- VRRP
- VRRP

If you configure a protocol that is not supported by nonstop active routing, the protocol operates as usual. When a switchover occurs, the state information for the unsupported protocol is not preserved and must be refreshed using the normal recovery mechanisms inherent in the protocol.

On routers that have logical systems configured on them, NSR is only supported in the main instance.

In a Virtual Chassis environment configured with OSPF and NSR, any failure or restart of the backup device can lead to longer global convergence times compared to environments where NSR is not configured.

Nonstop Active Routing BFD Support

Nonstop active routing supports the Bidirectional Forwarding Detection (BFD) protocol, which uses the topology discovered by routing protocols to monitor neighbors. The BFD protocol is a simple hello mechanism that detects failures in a network. Because BFD is streamlined to be efficient at fast liveness detection, when it is used in conjunction with routing protocols, routing recovery times are improved. With nonstop active routing enabled, BFD session states are not restarted when a Routing Engine switchover occurs.

NOTE: BFD session states are saved only for clients using aggregate or static routes or for BGP, IS-IS, OSPF/OSPFv3, PIM, or RSVP.

When a BFD session is distributed to the Packet Forwarding Engine, BFD packets continue to be sent during a Routing Engine switchover. If nondistributed BFD sessions are to be kept alive during a switchover, you must ensure that the session failure detection time is greater than the Routing Engine switchover time. The following BFD sessions are not distributed to the Packet Forwarding Engine: multihop sessions, tunnel-encapsulated sessions, and sessions over integrated routing and bridging (IRB) interfaces.

NOTE: BFD is an intensive protocol that consumes system resources. Specifying a minimum interval for BFD less than 100 ms for Routing Engine-based sessions and 10 ms for distributed BFD sessions can cause undesired BFD flapping. The `minimum-interval` configuration statement is a BFD liveness detection parameter.

Depending on your network environment, these additional recommendations might apply:

- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 ms for Routing Engine-based sessions, and 100 ms for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing is configured, specify a minimum interval of 2.5 seconds for Routing Engine-based sessions. For distributed BFD sessions with nonstop active routing configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

Nonstop Active Routing BGP Support

Nonstop active routing BGP support is subject to the following conditions:

- You must include the `path-selection external-router-ID` statement at the `[edit protocols bgp]` hierarchy level to ensure consistent path selection between the primary and backup Routing Engines during and after the nonstop active routing switchover.
- You must include the `advertise-from-main-vpn-tables` statement at the `[edit protocols bgp]` hierarchy level to prevent BGP sessions from going down when route reflector (RR) or autonomous system border router (ASBR) functionality is enabled or disabled on a routing device that has VPN address families configured.
- BGP session uptime and downtime statistics are not synchronized between the primary and backup Routing Engines during Nonstop Active Routing and ISSU. The backup Routing Engine maintains its own session uptime based on the time when the backup first becomes aware of the established sessions. For example, if the backup Routing Engine is rebooted (or if you run `restart routing` on the backup Routing Engine), the backup's uptime is a short duration, because the backup has just learned about the established sessions. If the backup is operating when the BGP sessions first come up on the primary, the uptime on the primary and the uptime on the backup are almost the same duration. After a Routing Engine switchover, the new primary continues from the time left on the backup Routing Engine.
- If the BGP peer in the primary Routing Engine has negotiated address-family capabilities that are not supported for nonstop active routing, then the corresponding BGP neighbor state on the backup Routing Engine shows as idle. On switchover, the BGP session is reestablished from the new primary Routing Engine.

  Only the following address families are supported for nonstop active routing:

  - evpn-signaling
  - inet labeled-unicast
  - inet-mdt
  - inet multicast
  - inet-mvpn
  - inet unicast
  - inet-vpn unicast
  - inet6 labeled-unicast
  - inet6 multicast
  - inet6-mvpn
  - inet6 unicast

- inet6-vpn unicast
- iso-vpn
- l2vpn signaling
- route-target

> **NOTE**: Address families are supported only on the main instance of BGP. Only unicast is supported on VRF instances.

- BGP route dampening does not work on the backup Routing Engine when nonstop active routing is enabled.

Nonstop Active Routing Layer 2 Circuit and VPLS Support

Nonstop active routing supports Layer 2 circuit and VPLS on both LDP-based and RSVP-TE-based networks. Nonstop active routing support enables the backup Routing Engine to track the label advertised by Layer 2 circuit and VPLS on the primary Routing Engine, and to use the same label after the Routing Engine switchover.

Nonstop active routing supports Layer 2 circuit and LDP-based VPLS pseudowire redundant configurations.

Nonstop Active Routing PIM Support

Nonstop active routing supports Protocol Independent Multicast (PIM) with stateful replication on backup Routing Engines. State information replicated on the backup Routing Engine includes information about neighbor relationships, join and prune events, rendezvous point (RP) sets, synchronization between routes and next hops, multicast session states, and the forwarding state between the two Routing Engines.

Nonstop active routing for PIM is supported for IPv4 and IPv6. Junos OS also supports nonstop active routing for PIM on devices that have both IPv4 and IPv6 configured on them.

To configure nonstop active routing for PIM, include the same statements in the configuration as for other protocols: the `nonstop-routing` statement at the `[edit routing-options]` hierarchy level and the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level. To trace PIM nonstop active routing events, include the `flag nsr-synchronization` statement at the `[edit protocols pim traceoptions]` hierarchy level.

NOTE: The `clear pim join`, `clear pim register`, and `clear pim statistics` operational mode commands are not supported on the backup Routing Engine when nonstop active routing is enabled.

Nonstop active routing support varies for different PIM features. The features fall into the following three categories: supported features, unsupported features, and incompatible features.

Supported features:

- Auto-RP

  NOTE: Nonstop active routing PIM support on IPv6 does not support auto-RP because IPv6 does not support auto-RP.

- Bootstrap router (BSR)
- Static RPs
- Embedded RP on non-RP IPv6 routers
- Local RP

  NOTE: RP set information synchronization is supported for local RP and BSR (on IPv4 and IPv6), autoRP (on IPv4), and embedded RP (on IPv6).

- BFD
- Dense mode
- Sparse mode
- Source-specific multicast (SSM)
- Draft Rosen multicast VPNs (MVPNs)
- Anycast RP (anycast RP set information synchronization and anycast RP register state synchronization on IPv4 and IPv6 configurations)
- Flow maps
- Unified ISSU

- Policy features such as neighbor policy, bootstrap router export and import policies, scope policy, flow maps, and reverse path forwarding (RPF) check policies
- Upstream assert synchronization
- PIM join load balancing

Junos OS supports nonstop active routing PIM for draft Rosen MVPNs. Nonstop active routing PIM support for draft Rosen MVPNs enables nonstop active routing-enabled devices to preserve draft Rosen MPVN-related information—such as default and data multicast distribution tree (MDT) states—across switchovers.

The backup Routing Engine sets up the default MDT based on the configuration and the information it receives from the primary Routing Engine, and keeps updating the default MDT state information.

However, for data MDTs, the backup Routing Engine relies on the primary Routing Engine to provide updates when data MDTs are created, updated, or deleted. The backup Routing Engine neither monitors data MDT flow rates nor triggers a data MDT switchover based on variations in flow rates. Similarly, the backup Routing Engine does not maintain the data MDT delay timer or timeout timer. It does not send MDT join TLV packets for the data MDTs until it takes over as the primary Routing Engine. After the switchover, the new primary Routing Engine starts sending MDT join TLV packets for each data MDT, and also resets the data MDT timers. Note that the expiration time for the timers might vary from the original values on the previous primary Routing Engine.

Junos OS supports Protocol Independent Multicast (PIM) nonstop active routing on IGMP-only interfaces. Multicast joins on IGMP-only interfaces are mapped to PIM states, and these states are replicated on the backup Routing Engine. If the corresponding PIM states are available on the backup, the multicast routes are marked as forwarding on the backup Routing Engine. This enables uninterrupted traffic flow after a switchover. This support covers IGMPv2, IGMPv3, MLDv1, and MLDv2 reports and leaves.

Unsupported features: You can configure the following PIM features on a router along with nonstop active routing, but they function as if nonstop active routing is not enabled. In other words, during Routing Engine switchover and other outages, their state information is not preserved, and traffic loss is to be expected.

- Internet Group Management Protocol (IGMP) exclude mode
- IGMP snooping

Nonstop active routing is not supported for next-generation MVPNs with PIM provider tunnels. The commit operation fails if the configuration includes both nonstop active routing and next-generation MVPNs with PIM provider tunnels.

Junos OS provides a configuration statement that disables nonstop active routing for PIM only, so that you can activate incompatible PIM features and continue to use nonstop active routing for the other protocols on the router. Before activating an incompatible PIM feature, include the `nonstop-routing disable`

statement at the [edit protocols pim] hierarchy level. Note that in this case, nonstop active routing is disabled for all PIM features, not just incompatible features.

Nonstop Active Routing MSDP Support

Junos OS supports nonstop active routing for the Multicast Source Discovery Protocol (MSDP).

Nonstop active routing support for MSDP preserves the following MSDP-related information across the switchover:

- MSDP configuration and peer information
- MSDP peer socket information
- Source-active and related information

However, note that the following restrictions or limitations apply to nonstop active routing MSDP support:

- Because the backup Routing Engine learns the active source information by processing the source-active messages from the network, synchronizing of source active information between the primary and backup Routing Engines might take up to 60 seconds. So, no planned switchover is allowed within 60 seconds of the initial replication of the sockets.
- Similarly, Junos OS does not support two planned switchovers within 240 seconds of each other.

Junos OS enables you to trace MSDP nonstop active routing events by including the flag nsr-synchronization statement at the [edit protocols msdp traceoptions] hierarchy level.

Nonstop Active Routing Support for RSVP-TE LSPs

Junos OS supports nonstop active routing for label-switching routers (LSRs) and Layer 2 Circuits that are part of an RSVP-TE LSP. Nonstop active routing support on LSRs ensures that the primary to backup Routing Engine switchover on an LSR remains transparent to the network neighbors and that the LSP information remains unaltered during and after the switchover.

You can use the show rsvp version command to view the nonstop active routing mode and state on an LSR. Similarly, you can use the show mpls lsp and show rsvp session commands on the backup Routing Engine to view the state recreated on the backup Routing Engine.

The Junos OS nonstop active routing feature is also supported on RSVP point-to-multipoint LSPs. During the switchover, the LSP comes up on the backup Routing Engine that shares and synchronizes the state information with the primary Routing Engine before and after the switchover. Nonstop active routing support for point-to-multipoint transit and egress LSPs ensures that the switchover remains transparent to the network neighbors, and preserves the LSP information across the switchover.

Junos OS supports nonstop active routing for next-generation multicast VPNs (MVPNs).

The `show rsvp session detail` command enables you to check the point-to-multipoint LSP remerge state information (`P2MP LSP re-merge`; possible values are `head`, `member`, and `none`).

Junos OS supports nonstop active routing for point-to-multipoint LSPs used by VPLS and MVPN.

However, Junos OS does not support nonstop active routing for the following features:

- Generalized Multiprotocol Label Switching (GMPLS) and LSP hierarchy
- Interdomain or loose-hop expansion LSPs
- BFD liveness detection
- Setup protection

Nonstop active routing support for RSVP-TE LSPs is subject to the following limitations and restrictions:

- Detour LSPs are not maintained across a switchover and so, detour LSPs might fail to come back online after the switchover.
- Control plane statistics corresponding to the `show rsvp statistics` and `show rsvp interface detail | extensive` commands are not maintained across Routing Engine switchovers.
- Statistics from the backup Routing Engine are not reported for `show mpls lsp statistics` and `monitor mpls label-switched-path` commands. However, if a switchover occurs, the backup Routing Engine, after taking over as the primary, starts reporting statistics. Note that the `clear statistics` command issued on the old primary Routing Engine does not have any effect on the new primary Routing Engine, which reports statistics, including any uncleared statistics.
- State timeouts might take additional time during nonstop active routing switchover. For example, if a switchover occurs after a neighbor has missed sending two hello messages to the primary, the new primary Routing Engine waits for another three hello periods before timing out the neighbor.
- On the RSVP ingress router, if you configure auto-bandwidth functionality, the bandwidth adjustment timers are set in the new primary after the switchover. This causes a one-time increase in the length of time required for the bandwidth adjustment after the switchover occurs.
- Backup LSPs —LSPs that are established between the point of local repair (PLR) and the merge point after a node or link failure—are not preserved during a Routing Engine switchover.
- When nonstop active routing is enabled, graceful restart is not supported. However, graceful restart helper mode is supported.

SEE ALSO

Nonstop Active Routing Concepts

Configuring Nonstop Active Routing | 280

Configuring Nonstop Active Routing on Switches

Example: Configuring Nonstop Active Routing on Switches | 288

Platform-Specific NSR Behavior

Use the following table to review platform-specific behaviors for your platforms.

Platform	Difference
EX Series	On EX9214 switches, the VRRP primary state might change during graceful Routing Engine switchover, even when nonstop active routing is enabled.
MX Series	NSR is not supported during the Routing Engine reboot process on MX Series devices with the Next-Generation Routing Engine (NG-RE) installed. NSR will still work during the Routing Engine switchover process.
PTX Series	Nonstop active routing (NSR) switchover on PTX Series is supported only for the following MPLS and VPN protocols and applications using chained composite next hops: • Labeled BGP • Layer 2 VPNs excluding Layer 2 interworking (Layer 2 switching) • Layer 3 VPNs • LDP • RSVP

Change History Table

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

Release	Description
15.1R1	Starting with Junos OS Release 15.1R1, if you have NSR configured, it is never valid to issue the `restart routing` command in any form on the NSR primary Routing Engine.
12.3	Starting with Junos OS Release 12.3, because of its synchronization requirements and logic, NSR or GRES performance is limited by the slowest Routing Engine in the system.

Configuring Nonstop Active Routing

SUMMARY

Configure nonstop active routing on your device with the following steps and examples.

IN THIS SECTION

Enabling Nonstop Active Routing

Nonstop active routing (NSR) requires you to configure graceful Routing Engine switchover (GRES). To enable graceful Routing Engine switchover, include the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level:

```
[edit chassis redundancy]
graceful-switchover;
```

By default, nonstop active routing is disabled. To enable nonstop active routing, include the `nonstop-routing` statement at the `[edit routing-options]` hierarchy level:

```
[edit routing-options]
nonstop-routing;
```

To disable nonstop active routing, remove the `nonstop-routing` statement from the `[edit routing-options]` hierarchy level.

NOTE: When you enable nonstop active routing, you cannot enable automatic route distinguishers for multicast VPN routing instances. Automatic route distinguishers are enabled by configuring the `route-distinguisher-id` statement at the `[edit routing-instances` *`instance-name`*`]` hierarchy level; for more information, see the Junos OS VPNs Library for Routing Devices.

If the routing protocol process (rpd) on the NSR primary Routing Engine crashes, the primary Routing Engine simply restarts rpd (with no Routing Engine switchover), which impacts routing protocol adjacencies and neighbors and results in traffic loss. To prevent this negative impact on traffic flow, configure the `switchover-on-routing-crash` statement at the `[edit system]` hierarchy level. This configuration forces an NSR Routing Engine switchover if rpd on the primary Routing Engine crashes.

```
[edit system]
user@host# set switchover-on-routing-crash
```

To enable the routing platform to switch over to the backup Routing Engine when the routing protocol process (rpd) fails rapidly three times in succession, include the `other-routing-engine` statement at the `[edit system processes routing failover]` hierarchy level.

For more information about the `other-routing-engine` statement, see the Junos OS Administration Library for Routing Devices.

Synchronizing the Routing Engine Configuration

When you configure nonstop active routing, you must also include the `commit synchronize` statement at the `[edit system]` hierarchy level so that configuration changes are synchronized on both Routing Engines:

```
[edit system]commit synchronize;
```

If you try to commit the nonstop active routing configuration without including the `commit synchronize` statement, the commit fails.

If you configure the `commit synchronize` statement at the `[edit system]` hierarchy level and issue a commit in the primary Routing Engine, the primary configuration is automatically synchronized with the backup.

However, if the backup Routing Engine is down when you issue the commit, the Junos OS displays a warning and commits the candidate configuration in the primary Routing Engine. When the backup Routing Engine comes up, its configuration will automatically be synchronized with the primary.

NOTE: A newly inserted backup Routing Engine automatically synchronizes its configuration with the primary Routing Engine configuration.

When you configure nonstop active routing, you can bring the backup Routing Engine online after the primary Routing Engine is already running. There is no requirement to start the two Routing Engines simultaneously.

CAUTION: We recommend that you do not restart Routing Protocol Process (rpd) on primary Routing Engine after enabling nonstop active routing, as it disrupts the protocol adjacency/peering sessions, resulting in traffic loss.

Verifying Nonstop Active Routing Operation

To see whether or not nonstop active routing is enabled, issue the `show task replication` command. For BGP nonstop active routing, you must also issue the `show bgp replication` command.

CAUTION: If BGP is configured, before attempting nonstop active routing switchover, check the output of `show bgp replication` to confirm that BGP routing table synchronization has completed on the backup Routing Engine. The `complete` status in the output of `show task replication` only indicates that the socket replication has completed

and the BGP synchronization is in progress. To determine whether BGP synchronization is complete, you must check the `Protocol state` and `Synchronization state` fields in the output of `show bgp replication` on the primary Routing Engine. The `Protocol state` must be `idle` and the `Synchronization state` must be `complete`. If you perform NSR switchover before the BGP synchronization has completed, the BGP session might flap.

When you enable nonstop active routing or graceful Routing Engine switchover and issue routing-related operational mode commands on the backup Routing Engine (such as `show route`, `show bgp neighbor`, `show ospf database`, and so on), the output might not match the output of the same commands issued on the primary Routing Engine. For example, it is normal for the routing table on the backup Routing Engine to contain persistent phantom routes that are not present in the routing table on the primary Routing Engine.

To display BFD state replication status, issue the `show bfd session` command. The `replicated` flag appears in the output for this command when a BFD session has been replicated to the backup Routing Engine.

Configuring Nonstop Active Routing on Switches

Nonstop active routing (NSR) provides a mechanism for transparent switchover of the Routing Engines without necessitating restart of supported routing protocols. Both Routing Engines are fully active in processing protocol sessions, and so each can take over for the other. The switchover is transparent to neighbors.

You can configure NSR on an on a Juniper Networks EX Series switch with multiple Routing Engines or an EX Series or QFX Series switch in a Virtual Chassis or Virtual Chassis Fabric configuration.

To configure nonstop active routing:

1. Enable graceful Routing Engine switchover (GRES):

```
[edit chassis redundancy]
user@switch# set graceful-switchover
```

2. Enable nonstop active routing (by default, nonstop active routing is disabled):

```
[edit routing-options]
user@switch# set nonstop-routing
```

3. Synchronize configuration changes between the Routing Engines:

```
[edit system]
user@switch# set commit synchronize
```

   If you try to commit the nonstop active routing configuration without including the `commit synchronize` statement, the commit fails.

4.

NOTE: There is no requirement to start the two Routing Engines simultaneously. If the backup Routing Engine is not up when you issue the `commit synchronize` command , the candidate configuration is committed in the primary Routing Engine. When the backup Routing Engine is inserted or comes online, its configuration is automatically synchronized with that of the primary.

BEST PRACTICE: After a graceful Routing Engine switchover, we recommend that you issue the `clear interface statistics (`*`interface-name`*` | all)` command to reset the cumulative values for local statistics on the new primary Routing Engine.

To disable nonstop active routing:

```
[edit routing-options]
user@switch# delete nonstop-routing
```

Preventing Automatic Reestablishment of BGP Peer Sessions After NSR Switchovers

It is useful to prevent a BGP peer session from automatically being reestablished after a nonstop active routing (NSR) switchover when you have applied routing policies configured in the dynamic database. When NSR is enabled, the dynamic database is not synchronized with the backup Routing Engine. Therefore, when a switchover occurs, import and export policies configured in the dynamic database might no longer be available. For more information about configuring dynamic routing policies, see the Routing Policies, Firewall Filters, and Traffic Policers User Guide.

NOTE: The BGP established timers are not maintained across switchovers.

You can configure the routing device not to reestablish a BGP peer session after an NSR switchover either for a specified period or until you manually reestablish the session. Include the `idle-after-switch-over` statement at the `[edit protocols bgp]` hierarchy level:

```
idle-after-switch-over (forever | seconds);
```

For a list of hierarchy levels at which you can configure this statement, see the configuration statement summary for this statement.

For ***seconds***, specify a value from 1 through 4294967295. The BGP peer session is not reestablished until after the specified period. If you specify the **forever** option, the BGP peer session is not reestablished until you issue the `clear bgp neighbor` command.

Example: Configuring Nonstop Active Routing

The following example enables graceful Routing Engine switchover, nonstop active routing, and nonstop active routing trace options for BGP, IS-IS, and OSPF.

```
[edit]
system commit {
    synchronize;
}
chassis {
    redundancy {
        graceful-switchover; # This enables graceful Routing Engine switchover on
# the routing platform.
    }
}
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.0.1.1/30;
            }
            family iso;
        }
```

```
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.1.1.1/30;
            }
            family iso;
        }
    }
    so-0/0/2 {
        unit 0 {
            family inet {
                address 10.2.1.1/30;
            }
            family iso;
        }
    }
    so-0/0/3 {
        unit 0 {
            family inet {
                address 10.3.1.1/30;
            }
            family iso;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.2.1/32;
            }
            family iso {
                address 49.0004.1921.6800.2001.00;
            }
        }
    }
}
routing-options {
    nonstop-routing; # This enables nonstop active routing on the routing platform.
    router-id 192.168.2.1;
    autonomous-system 65432;
}
protocols {
    bgp {
```

```
        traceoptions {
            flag nsr-synchronization detail; # This logs nonstop active routing
# events for BGP.
        }
        advertise-from-main-vpn-tables;
        local-address 192.168.2.1;
        group external-group {
            type external;
            export BGP_export;
            neighbor 192.168.1.1 {
                family inet {
                    unicast;
                }
                peer-as 65103;
            }
        }
        group internal-group {
            type internal;
            neighbor 192.168.10.1;
            neighbor 192.168.11.1;
            neighbor 192.168.12.1;
        }
    }
    isis {
        traceoptions {
            flag nsr-synchronization detail; # This logs nonstop active routing events
# for IS-IS.
        }
        interface all;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
    ospf {
        traceoptions {
            flag nsr-synchronization detail; # This logs nonstop active routing events
# for OSPF.
        }
        area 0.0.0.0 {
            interface all;
```

```
            interface fxp0.0 {
                disable;
            }
            interface lo0.0 {
                passive;
            }
        }
    }
}
policy-options {
    policy-statement BGP_export {
        term direct {
            from {
                protocol direct;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```

Resetting Local Statistics

After a graceful Routing Engine switchover, we recommend that you issue the `clear interface statistics (`*`interface-name`*` | all)` command to reset the cumulative values for local statistics on the new primary Routing Engine.

Example: Configuring Nonstop Active Routing on Switches

IN THIS SECTION

Nonstop active routing (NSR) provides high availability for Routing Engines by enabling transparent switchover of the Routing Engines without necessitating restart of supported routing protocols. Both Routing Engines are fully active in processing protocol sessions, and so each can take over for the other. The switchover is transparent to neighbors.

This example describes how to configure nonstop active routing on switches with multiple Routing Engines or on an EX Series or a QFX series switch in a Virtual Chassis or Virtual Chassis Fabric configuration.

Requirements

This example uses the following hardware and software components:

- An EX Series with multiple Routing Engines or on an EX Series or a QFX series switch in a Virtual Chassis or Virtual Chassis Fabric configuration
- Junos OS Release 10.4 or later for EX Series switches
- Junos OS Release 13.2X51-D20 or later for QFX Series switches

Overview and Topology

Configure nonstop active routing on any EX Series with multiple Routing Engines or on an EX Series or a QFX series switch in a Virtual Chassis or Virtual Chassis Fabric configuration. Nonstop active routing is advantageous in networks where neighbor routing devices do not support graceful restart protocol extensions.

The topology used in this example consists of an EX8200 switch with redundant Routing Engines connected to neighbor routing devices that are not configured to support graceful restart of protocols.

Configuration

IN THIS SECTION

CLI Quick Configuration

To quickly configure nonstop active routing, copy the following commands and paste them into the switch terminal window:

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

Procedure

Step-by-Step Procedure

To configure nonstop active routing on a switch:

1. Enable graceful Routing Engine switchover (GRES):

   ```
   [edit chassis redundancy]
   user@switch# set graceful-switchover
   ```

2. Enable nonstop active routing (by default, nonstop active routing is disabled):

   ```
   [edit routing-options]
   user@switch# set nonstop-routing
   ```

3. Synchronize configuration changes between the Routing Engines:

   ```
   [edit system]
   user@switch# set commit synchronize
   ```

   If you try to commit the nonstop active routing configuration without including the `commit synchronize` statement, the commit fails.

NOTE: If the backup Routing Engine is down when you issue the commit, a warning is displayed and the candidate configuration is committed in the primary Routing Engine. When the backup Routing Engine comes up, its configuration is automatically synchronized with that of the primary. If you subsequently insert or bring up a backup Routing Engine, it automatically synchronizes its configuration with the primary Routing Engine configuration.

Results

Check the results of the configuration:

```
[edit]
user@switch# show
chassis {
    redundancy {
        graceful-switchover;
    }
routing-options {
    nonstop-routing;
}
system {
    commit synchronize;
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly, perform these tasks:

Verifying That Nonstop Active Routing Is Working Correctly on the Switch

Purpose

Verify that nonstop active routing is enabled.

Action

Issue the `show task replication` command:

```
user@switch# show task replication
Stateful Replication: Enabled
RE mode: Master

 Protocol                Synchronization Status
 OSPF                    Complete
 RIP                     Complete
 PIM                     Complete
 RSVP                    Complete
```

Meaning

This output shows that nonstop active routing (Stateful Replication) is enabled on primary routing engine. If nonstop routing is not enabled, instead of the output shown above:

- On the backup routing engine the following error message is displayed: "`error: the routing subsystem is not running`."
- On the primary routing engine, the following output is displayed if nonstop routing is not enabled:

```
Stateful Replication: Disabled
RE mode: Master
```

Troubleshooting

IN THIS SECTION

To troubleshoot nonstop active routing, perform these tasks:

Investigating Problems with Synchronization of Routing Engines When NSR Is Enabled

Problem

A protocol loses connectivity with neighbors after a graceful Routing Engine switchover (GRES) occurs with nonstop active routing (NSR) enabled.

Solution

Use trace options to help isolate the problem and gather troubleshooting information. Using the information gathered from trace options, you can confirm or eliminate the synchronization of the Routing Engines as the cause of the loss of connectivity for the protocol. See "Tracing Nonstop Active Routing Synchronization Events" on page 1186.

RELATED DOCUMENTATION

Nonstop Active Routing Concepts

Nonstop Active Routing System Requirements

Tracing Nonstop Active Routing Synchronization Events | **1186**

nonstop-routing

PART 10

Configuring Graceful Restart

Understanding Graceful Restart

SUMMARY

Graceful restart allows for uninterrupted packet forwarding and temporary suppression of all routing protocol updates during the restart process.

IN THIS SECTION

Graceful Restart Concepts

With routing protocols, any service interruption requires that an affected router recalculate adjacencies with neighboring routers, restore routing table entries, and update other protocol-specific information. An unprotected restart of a router can result in forwarding delays, route flapping, wait times stemming from protocol reconvergence, and even dropped packets. Some benefits of graceful restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful restart enables a router to pass through intermediate convergence states that are hidden from the rest of the network.

Three main types of graceful restart are available on Juniper Networks routing platforms:

- Graceful restart for aggregate and static routes and for routing protocols—Provides protection for aggregate and static routes and for Border Gateway Protocol (BGP), End System-to-Intermediate System (ES-IS), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), next-generation RIP (RIPng), and Protocol Independent Multicast (PIM) sparse mode routing protocols.

- Graceful restart for MPLS-related protocols—Provides protection for Label Distribution Protocol (LDP), Resource Reservation Protocol (RSVP), circuit cross-connect (CCC), and translational cross-connect (TCC). (Not supported on OCX Series switches.)
- Graceful restart for virtual private networks (VPNs)—Provides protection for Layer 2 and Layer 3 VPNs.

Graceful restart works similarly for routing protocols and MPLS protocols and combines components of these protocol types to enable graceful restart in VPNs. The main benefits of graceful restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful restart thus enables a router to pass through intermediate convergence states that are hidden from the rest of the network.

Most graceful restart implementations define two types of routers—the restarting router and the helper router. The restarting router requires rapid restoration of forwarding state information so it can resume the forwarding of network traffic. The helper router assists the restarting router in this process. Graceful restart configuration statements typically affect either the restarting router or the helper router.

Graceful Restart for Aggregate and Static Routes

When you include the `graceful-restart` statement at the `[edit routing-options]` hierarchy level, any static routes or aggregated routes that have been configured are protected. Because no helper router assists in the restart, these routes are retained in the forwarding table while the router restarts (rather than being discarded or refreshed).

Graceful Restart and Routing Protocols

IN THIS SECTION

This section covers the following topics:

BGP

When a router enabled for BGP graceful restart restarts, it retains BGP peer routes in its forwarding table and marks them as stale. However, it continues to forward traffic to other peers (or receiving peers) during the restart. To reestablish sessions, the restarting router sets the “restart state” bit in the BGP OPEN message and sends it to all participating peers. The receiving peers reply to the restarting router with messages containing end-of-routing-table markers. When the restarting router or switch receives all replies from the receiving peers, the restarting router performs route selection, the forwarding table is updated, and the routes previously marked as stale are discarded. At this point, all BGP sessions are reestablished and the restarting peer can receive and process BGP messages as usual.

While the restarting router does its processing, the receiving peers also temporarily retain routing information. When a receiving peer detects a TCP transport reset, it retains the routes received and marks the routes as stale. After the session is reestablished with the restarting router or switch , the stale routes are replaced with updated route information.

IS-IS

Normally, IS-IS routers move neighbor adjacencies to the down state when changes occur. However, a router enabled for IS-IS graceful restart sends out Hello messages with the Restart Request (RR) bit set in a restart type length value (TLV) message. This indicates to neighboring routers that a graceful restart is in progress and to leave the IS-IS adjacency intact. The neighboring routers must interpret and implement restart signaling themselves. Besides maintaining the adjacency, the neighbors send complete sequence number PDUs (CSNPs) to the restarting router and flood their entire database.

The restarting router never floods any of its own link-state PDUs (LSPs), including pseudonode LSPs, to IS-IS neighbors while undergoing graceful restart. This enables neighbors to reestablish their adjacencies without transitioning to the down state and enables the restarting router to reinitiate a smooth database synchronization.

OSPF and OSPFv3

When a router enabled for OSPF graceful restart restarts, it retains routes learned before the restart in its forwarding table. The router does not allow new OSPF link-state advertisements (LSAs) to update the routing table. This router continues to forward traffic to other OSPF neighbors (or helper routers), and sends only a limited number of LSAs during the restart period. To reestablish OSPF adjacencies with neighbors, the restarting router must send a grace LSA to all neighbors. In response, the helper routers enter helper mode and send an acknowledgement back to the restarting router. If there are no topology changes, the helper routers continue to advertise LSAs as if the restarting router had remained in continuous OSPF operation.

When the restarting router receives replies from all the helper routers, the restarting router selects routes, updates the forwarding table, and discards the old routes. At this point, full OSPF adjacencies are reestablished and the restarting router receives and processes OSPF LSAs as usual. When the helper routers no longer receive grace LSAs from the restarting router or the topology of the network changes, the helper routers also resume normal operation.

NOTE: For more information about the standard helper mode implementation, see RFC 3623, *Graceful OSPF Restart*.

Starting with Release 11.3, Junos OS supports the restart signaling-based helper mode for OSPF graceful restart configurations. The helper modes, both standard and restart signaling-based, are enabled by default. In restart signaling-based helper mode implementations, the restarting router relays the restart status to its neighbors only after the restart is complete. When the restart is complete, the restarting router sends hello messages to its helper routers with the **restart signal** (**RS**) bit set in the hello packet header. When a helper router receives a hello packet with the **RS** bit set in the header, the helper router returns a hello message to the restarting router. The reply hello message from the helper router contains the **ResyncState** flag and the **ResyncTimeout** timer that enable the restarting router to keep track of the helper routers that are syncing up with it. When all helpers complete the synchronization, the restarting router exits the restart mode.

NOTE:

For more information about restart signaling-based graceful restart helper mode implementation, see RFC 4811, *OSPF Out-of-Band Link State Database (LSDB) Resynchronization*, RFC 4812, *OSPF Restart Signaling*, and RFC 4813, *OSPF Link-Local Signaling*.

Restart signaling-based graceful restart helper mode is not supported for OSPFv3 configurations.

PIM Sparse Mode

PIM sparse mode uses a mechanism called a *generation identifier* to indicate the need for graceful restart. Generation identifiers are included by default in PIM hello messages. An initial generation identifier is created by each PIM neighbor to establish device capabilities. When one of the PIM neighbors restarts, it sends a new generation identifier to its neighbors. All neighbors that support graceful restart and are connected by point-to-point links assist by sending multicast updates to the restarting neighbor.

The restart phase completes when either the PIM state becomes stable or when the restart interval timer expires. If the neighbors do not support graceful restart or connect to each other using multipoint interfaces, the restarting router uses the restart interval timer to define the restart period.

RIP and RIPng

When a router enabled for RIP graceful restart restarts, routes that have been configured are protected. Because no helper router assists in the restart, these routes are retained in the forwarding table while the router restarts (rather than being discarded or refreshed).

Graceful Restart and MPLS-Related Protocols

IN THIS SECTION

This section contains the following topics:

LDP

LDP graceful restart enables a router whose LDP control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. It also enables a router on which helper mode is enabled to assist a neighboring router that is attempting to restart LDP.

During session initialization, a router advertises its ability to perform LDP graceful restart or to take advantage of a neighbor performing LDP graceful restart by sending the graceful restart TLV. This TLV contains two fields relevant to LDP graceful restart: the reconnect time and the recovery time. The values of the reconnect and recovery times indicate the graceful restart capabilities supported by the router.

The default reconnect time is configured in Junos OS as 60 seconds and is user-configurable. The reconnect time is how long the helper router waits for the restarting router to establish a connection. If the connection is not established within the reconnect interval, graceful restart for the LDP session is terminated. The default maximum reconnect time is 120 seconds and is user-configurable. The maximum reconnect time is the maximum value that a helper router accepts from its restarting neighbor.

When a router discovers that a neighboring router is restarting, it waits until the end of the recovery time before attempting to reconnect. The recovery time is the length of time a router waits for LDP to restart gracefully. The recovery time period begins when an initialization message is sent or received.

This time period is also typically the length of time that a neighboring router maintains its information about the restarting router, so it can continue to forward traffic.

You can configure LDP graceful restart both in the master instance for the LDP protocol and for a specific routing instance. You can disable graceful restart at the global level for all protocols, at the protocol level for LDP only, and for a specific routing instance only.

RSVP

RSVP graceful restart enables a router undergoing a restart to inform its adjacent neighbors of its condition. The restarting router requests a grace period from the neighbor or peer, which can then cooperate with the restarting router. The restarting router can still forward MPLS traffic during the restart period; convergence in the network is not disrupted. The restart is not visible to the rest of the network, and the restarting router is not removed from the network topology. RSVP graceful restart can be enabled on both transit routers and ingress routers. It is available for both point-to-point LSPs and point-to-multipoint LSPs.

CCC and TCC

CCC and TCC graceful restart enables Layer 2 connections between customer edge (CE) routers to restart gracefully. These Layer 2 connections are configured with the **remote-interface-switch** or `lsp-switch` statements. Because these CCC and TCC connections have an implicit dependency on RSVP LSPs, graceful restart for CCC and TCC uses the RSVP graceful restart capabilities.

RSVP graceful restart must be enabled on the provider edge (PE) routers and provider (P) routers to enable graceful restart for CCC and TCC. Also, because RSVP is used as the signaling protocol for signaling label information, the neighboring router must use helper mode to assist with the RSVP restart procedures.

Understanding Restart Signaling-Based Helper Mode Support for OSPF Graceful Restart

Starting with Release 11.4, Junos OS supports restart signaling-based helper mode for OSPF graceful restart configurations.

NOTE:

- Restart signaling-based graceful restart helper mode is not supported for OSPFv3 configurations.
- Junos OS releases prior to Release 11.4 and OSPFv3 configurations support only standard helper mode as defined in RFC 3623 . For more information about the standard helper mode implementation, see RFC 3623 and the *Junos OS High Availability Configuration Guide*.

Both standard and restart signaling-based helper modes are enabled by default, irrespective of the graceful-restart configuration status on the device.

In restart signaling-based helper mode implementations, the restarting router informs the restart status to its neighbors only after the restart is complete. When the restart is complete, the restarting router sends hello messages to its helper routers with the **restart signal** (**RS**) bit set in the hello packet header. When a helper router receives a hello packet with the **RS** bit set in the header, the helper router returns a hello message to the restarting router. The reply hello message from the helper router contains the **ResyncState** flag and the **ResyncTimeout** timer that enable the restarting router to keep track of the helper routers that are syncing up with it. When all helpers complete the synchronization, the restarting router exits the restart mode.

For more information about restart signaling-based graceful restart helper mode implementation, see RFC 4811, *OSPF Out-of-Band Link State Database (LSDB) Resynchronization*, RFC 4812, *OSPF Restart Signaling*and RFC 4813, *OSPF Link-Local Signaling*.

Graceful Restart and Layer 2 and Layer 3 VPNs

VPN graceful restart uses three types of restart functionality:

1. BGP graceful restart functionality is used on all PE-to-PE BGP sessions. This affects sessions carrying any service signaling data for network layer reachability information (NLRI), for example, an IPv4 VPN or Layer 2 VPN NLRI.
2. OSPF, IS-IS, LDP, or RSVP graceful restart functionality is used in all core routers. Routes added by these protocols are used to resolve Layer 2 and Layer 3 VPN NLRI.
3. Protocol restart functionality is used for any Layer 3 protocol (RIP, OSPF, LDP, and so on) used between the PE and customer edge (CE) routers. This does not apply to Layer 2 VPNs because Layer 2 protocols used between the CE and PE routers do not have graceful restart capabilities.

Before VPN graceful restart can work properly, all of the components must restart gracefully. In other words, the routers must preserve their forwarding states and request neighbors to continue forwarding

to the router in case of a restart. If all of the conditions are satisfied, VPN graceful restart imposes the following rules on a restarting router:

- The router must wait to receive all BGP NLRI information from other PE routers before advertising routes to the CE routers.
- The router must wait for all protocols in all routing instances to converge (or complete the restart process) before it sends CE router information to other PE routers. In other words, the router must wait for all instance information (whether derived from local configuration or advertisements received from a remote peer) to be processed before it sends this information to other PE routers.
- The router must preserve all forwarding state in the **instance.mpls.0** tables until the new labels and transit routes are allocated and announced to other PE routers (and CE routers in a carrier-of-carriers scenario).

  If any condition is not met, VPN graceful restart does not succeed in providing uninterrupted forwarding between CE routers across the VPN infrastructure.

Graceful Restart on Logical Systems

Graceful restart for a logical system functions much as graceful restart does in the main router. The only difference is the location of the `graceful-restart` statement:

- For a logical system, include the `graceful-restart` statement at the `[edit logical-systems` *`logical-system-name`* `routing-options]` hierarchy level.
- For a routing instance inside a logical system, include the `graceful-restart` statement at both the `[edit logical-systems` *`logical-system-name`* `routing-options]` and `[edit logical-systems` *`logical-system-name`* `routing-instances` *`instance-name`* `routing-options]` hierarchy levels.

Configuring Graceful Restart

SUMMARY

Follow these steps to configure graceful restart on your device.

IN THIS SECTION

Enabling Graceful Restart

Graceful restart is disabled by default. You must configure graceful restart at the `[edit routing-options]` or `[edit routing-instances *instance-name* routing-options]` hierarchy level to enable the feature globally.

For example:

```
routing-options {
    graceful-restart;
}
```

You can, optionally, modify the global settings at the individual protocol level or, as of Junos OS 15.1, at the individual routing instance level.

NOTE: If you configure graceful restart after a BGP or LDP session has been established, the BGP or LDP session restarts and the peers negotiate graceful restart capabilities.

To disable graceful restart, include the `disable` statement. You can do this globally for all protocols by including the `disable` statement at the `[edit routing-options]` hierarchy level, or you can disable graceful restart for a single protocol by including the disable statement at the `[edit protocols *protocol* graceful-restart]` hierarchy level. To configure a time period for complete restart, include the `restart-duration` statement. You can specify a number between 120 and 900.

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

When you include the `graceful-restart` statement at the `[edit routing-options]` hierarchy level, graceful restart is also enabled for aggregate and static routes.

Configuring Graceful Restart

To enable graceful restart, include the `graceful-restart` statement at the `[edit routing-instance` *`instance-name`* `routing-options]` or `[edit routing-options]` hierarchy level. This enables graceful restart globally for all routing protocols. You can, optionally, modify or supplement the global settings at the individual protocol level.

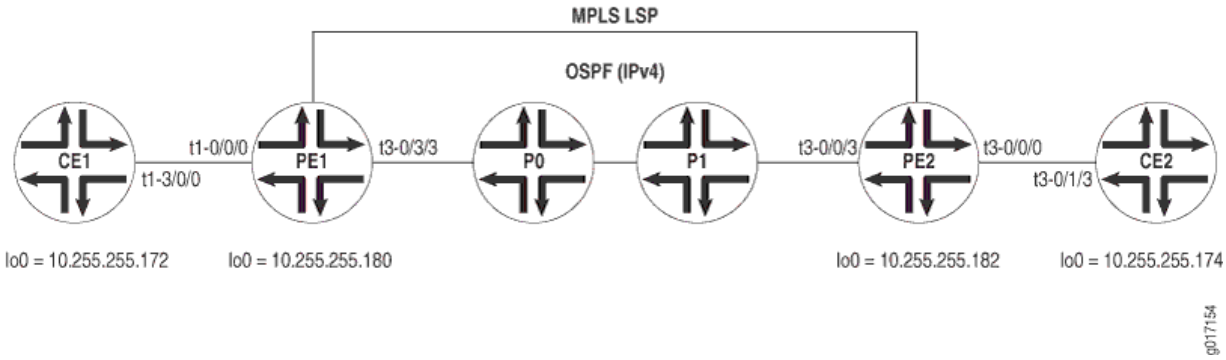
NOTE: When `set protocols bgp group` *`group-name`* `allow network` is configured to accept dynamic BGP sessions, `unconfigured-peer-graceful-restart` statement should be configured to avoid traffic drop during graceful restart or graceful Routing Engine switchover.

For example:

```
protocols {
    bgp {
        group ext {
            graceful-restart {
                restart-time 400;
            }
        }
    }
}
routing-options {
    graceful-restart;
}
```

Figure 22 on page 305 shows a standard MPLS VPN network. Routers CE1 and CE2 are customer edge routers, PE1 and PE2 are provider edge routers, and P0 is a provider core router. Several Layer 3 VPNs are configured across this network, as well as one Layer 2 VPN. Interfaces are shown in the diagram and are not included in the configuration example that follows.

Figure 22: Layer 3 VPN Graceful Restart Topology



Router CE1

On Router CE1, configure the following protocols on the logical interfaces of **t3-3/1/0**: OSPF on unit 101, RIP on unit 102, BGP on unit 103, and IS-IS on unit 512. Also configure graceful restart, BGP, IS-IS, OSPF, and RIP on the main instance to be able to connect to the routing instances on Router PE1.

```
[edit]
interfaces {
    t3-3/1/0 {
        encapsulation frame-relay;
        unit 100 {
            dlci 100;
            family inet {
                address 10.96.100.2/30;
            }
        }
        unit 101 {
            dlci 101;
            family inet {
                address 10.96.101.2/30;
            }
        }
        unit 102 {
            dlci 102;
            family inet {
                address 10.96.102.2/30;
            }
        }
        unit 103 {
            dlci 103;
            family inet {
```

```
                address 10.96.103.2/30;
            }
        }
        unit 512 {
            dlci 512;
            family inet {
                address 10.96.252.1/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.172/32;
                primary;
            }
            address 10.96.110.1/32;
            address 10.96.111.1/32;
            address 10.96.112.1/32;
            address 10.96.113.1/32;
            address 10.96.116.1/32;
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4172.00;
        }
    }
}
routing-options {
    graceful-restart;
    autonomous-system 65100;
}
protocols {
    bgp {
        group CE-PE-INET {
            type external;
            export BGP_INET_LB_DIRECT;
            neighbor 10.96.103.1 {
                local-address 10.96.103.2;
                family inet {
                    unicast;
                }
                peer-as 65103;
            }
```

```
        }
    }
    isis {
        export ISIS_L2VPN_LB_DIRECT;
        interface t3-3/1/0.512;
    }
    ospf {
        export OSPF_LB_DIRECT;
        area 0.0.0.0 {
            interface t3-3/1/0.101;
        }
    }
    rip {
        group RIP {
            export RIP_LB_DIRECT;
            neighbor t3-3/1/0.102;
        }
    }
}
policy-options {
    policy-statement OSPF_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.101.0/30 exact;
                route-filter 10.96.111.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement RIP_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.102.0/30 exact;
                route-filter 10.96.112.1/32 exact;
            }
            then accept;
        }
        term final {
```

```
            then reject;
        }
    }
    policy-statement BGP_INET_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.103.0/30 exact;
                route-filter 10.96.113.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement ISIS_L2VPN_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.116.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```

Router PE1

On Router PE1, configure graceful restart in the master instance, along with BGP, OSPF, MPLS, and LDP. Next, configure several protocol-specific instances of graceful restart. By including instances for BGP, OSPF, Layer 2 VPNs, RIP, and static routes, you can observe the wide range of options available when you implement graceful restart. Configure the following protocols in individual instances on the logical interfaces of **t3-0/0/0**: a static route on unit 100, OSPF on unit 101, RIP on unit 102, BGP on unit 103, and Frame Relay on unit 512 for the Layer 2 VPN instance.

```
[edit]
interfaces {
    t3-0/0/0 {
```

```
        dce;
        encapsulation frame-relay-ccc;
        unit 100 {
            dlci 100;
            family inet {
                address 10.96.100.1/30;
            }
            family mpls;
        }
        unit 101 {
            dlci 101;
            family inet {
                address 10.96.101.1/30;
            }
            family mpls;
        }
        unit 102 {
            dlci 102;
            family inet {
                address 10.96.102.1/30;
            }
            family mpls;
        }
        unit 103 {
            dlci 103;
            family inet {
                address 10.96.103.1/30;
            }
            family mpls;
        }
        unit 512 {
            encapsulation frame-relay-ccc;
            dlci 512;
        }
    }
    t1-0/1/0 {
        unit 0 {
            family inet {
                address 10.96.0.2/30;
            }
            family mpls;
        }
    }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.176/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4176.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    router-id 10.245.14.176;
    autonomous-system 69;
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        group PEPE {
            type internal;
            neighbor 10.245.14.182 {
                local-address 10.245.14.176;
                family inet-vpn {
                    unicast;
                }
                family l2vpn {
                    unicast;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/0.0;
            interface fxp0.0 {
                disable;
            }
            interface lo0.0 {
                passive;
            }
```

```
        }
    }
    ldp {
        interface all;
    }
}
policy-options {
    policy-statement STATIC-import {
        from community STATIC;
        then accept;
    }
    policy-statement STATIC-export {
        then {
            community add STATIC;
            accept;
        }
    }
    policy-statement OSPF-import {
        from community OSPF;
        then accept;
    }
    policy-statement OSPF-export {
        then {
            community add OSPF;
            accept;
        }
    }
    policy-statement RIP-import {
        from community RIP;
        then accept;
    }
    policy-statement RIP-export {
        then {
            community add RIP;
            accept;
        }
    }
    policy-statement BGP-INET-import {
        from community BGP-INET;
        then accept;
    }
    policy-statement BGP-INET-export {
        then {
```

```
            community add BGP-INET;
            accept;
        }
    }
    policy-statement L2VPN-import {
        from community L2VPN;
        then accept;
    }
    policy-statement L2VPN-export {
        then {
            community add L2VPN;
            accept;
        }
    }
    community BGP-INET members target:69:103;
    community L2VPN members target:69:512;
    community OSPF members target:69:101;
    community RIP members target:69:102;
    community STATIC members target:69:100;
}
routing-instances {
    BGP-INET {
        instance-type vrf;
        interface t3-0/0/0.103;
        route-distinguisher 10.245.14.176:103;
        vrf-import BGP-INET-import;
        vrf-export BGP-INET-export;
        routing-options {
            graceful-restart;
            autonomous-system 65103;
        }
        protocols {
            bgp {
                group BGP-INET {
                    type external;
                    export BGP-INET-import;
                    neighbor 10.96.103.2 {
                        local-address 10.96.103.1;
                        family inet {
                            unicast;
                        }
                        peer-as 65100;
                    }
```

```
            }
        }
    }
}
L2VPN {
    instance-type l2vpn;
    interface t3-0/0/0.512;
    route-distinguisher 10.245.14.176:512;
    vrf-import L2VPN-import;
    vrf-export L2VPN-export;
    protocols {# There is no graceful-restart statement for Layer 2 VPN instances.
        l2vpn {
            encapsulation-type frame-relay;
            site CE1-ISIS {
                site-identifier 512;
                interface t3-0/0/0.512 {
                    remote-site-id 612;
                }
            }
        }
    }
}
OSPF {
    instance-type vrf;
    interface t3-0/0/0.101;
    route-distinguisher 10.245.14.176:101;
    vrf-import OSPF-import;
    vrf-export OSPF-export;
    routing-options {
        graceful-restart;
    }
    protocols {
        ospf {
            export OSPF-import;
            area 0.0.0.0 {
                interface all;
            }
        }
    }
}
RIP {
    instance-type vrf;
    interface t3-0/0/0.102;
```

```
        route-distinguisher 10.245.14.176:102;
        vrf-import RIP-import;
        vrf-export RIP-export;
        routing-options {
            graceful-restart;
        }
        protocols {
            rip {
                group RIP {
                    export RIP-import;
                    neighbor t3-0/0/0.102;
                }
            }
        }
    }
    STATIC {
        instance-type vrf;
        interface t3-0/0/0.100;
        route-distinguisher 10.245.14.176:100;
        vrf-import STATIC-import;
        vrf-export STATIC-export;
        routing-options {
            graceful-restart;
            static {
                route 10.96.110.1/32 next-hop t3-0/0/0.100;
            }
        }
    }
}
```

Router P0

On Router P0, configure graceful restart in the main instance, along with OSPF, MPLS, and LDP. This allows the protocols on the PE routers to reach one another.

```
[edit]
interfaces {
    t3-0/1/3 {
        unit 0 {
            family inet {
                address 10.96.0.5/30;
            }
```

```
            family mpls;
        }
    }
    t1-0/2/0 {
        unit 0 {
            family inet {
                address 10.96.0.1/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.174/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4174.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    router-id 10.245.14.174;
    autonomous-system 69;
}
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/2/0.0;
            interface t3-0/1/3.0;
            interface fxp0.0 {
                disable;
            }
            interface lo0.0 {
                passive;
            }
        }
    }
```

```
    ldp {
        interface all;
    }
}
```

Router PE2

On Router PE2, configure BGP, OSPF, MPLS, LDP, and graceful restart in the master instance. Configure the following protocols in individual instances on the logical interfaces of **t1-0/1/3**: a static route on unit 200, OSPF on unit 201, RIP on unit 202, BGP on unit 203, and Frame Relay on unit 612 for the Layer 2 VPN instance. Also configure protocol-specific graceful restart in all routing instances, except the Layer 2 VPN instance.

```
[edit]
interfaces {
    t3-0/0/0 {
        unit 0 {
            family inet {
                address 10.96.0.6/30;
            }
            family mpls;
        }
    }
    t1-0/1/3 {
        dce;
        encapsulation frame-relay-ccc;
        unit 200 {
            dlci 200;
            family inet {
                address 10.96.200.1/30;
            }
            family mpls;
        }
        unit 201 {
            dlci 201;
            family inet {
                address 10.96.201.1/30;
            }
            family mpls;
        }
        unit 202 {
            dlci 202;
```

```
            family inet {
                address 10.96.202.1/30;
            }
            family mpls;
        }
        unit 203 {
            dlci 203;
            family inet {
                address 10.96.203.1/30;
            }
            family mpls;
        }
        unit 612 {
            encapsulation frame-relay-ccc;
            dlci 612;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.182/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4182.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    router-id 10.245.14.182;
    autonomous-system 69;
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        group PEPE {
            type internal;
            neighbor 10.245.14.176 {
                local-address 10.245.14.182;
                family inet-vpn {
```

```
                unicast;
            }
            family l2vpn {
                unicast;
            }
        }
    }
}
ospf {
    area 0.0.0.0 {
        interface t3-0/0/0.0;
        interface fxp0.0 {
            disable;
        }
        interface lo0.0 {
            passive;
        }
    }
}
ldp {
    interface all;
}
policy-options {
    policy-statement STATIC-import {
        from community STATIC;
        then accept;
    }
    policy-statement STATIC-export {
        then {
            community add STATIC;
            accept;
        }
    }
    policy-statement OSPF-import {
        from community OSPF;
        then accept;
    }
    policy-statement OSPF-export {
        then {
            community add OSPF;
            accept;
        }
    }
```

```
        policy-statement RIP-import {
            from community RIP;
            then accept;
        }
        policy-statement RIP-export {
            then {
                community add RIP;
                accept;
            }
        }
        policy-statement BGP-INET-import {
            from community BGP-INET;
            then accept;
        }
        policy-statement BGP-INET-export {
            then {
                community add BGP-INET;
                accept;
            }
        }
        policy-statement L2VPN-import {
            from community L2VPN;
            then accept;
        }
        policy-statement L2VPN-export {
            then {
                community add L2VPN;
                accept;
            }
        }
        community BGP-INET members target:69:103;
        community L2VPN members target:69:512;
        community OSPF members target:69:101;
        community RIP members target:69:102;
        community STATIC members target:69:100;
    }
    routing-instances {
        BGP-INET {
            instance-type vrf;
            interface t1-0/1/3.203;
            route-distinguisher 10.245.14.182:203;
            vrf-import BGP-INET-import;
            vrf-export BGP-INET-export;
```

```
            routing-options {
                graceful-restart;
                autonomous-system 65203;
            }
            protocols {
                bgp {
                    group BGP-INET {
                        type external;
                        export BGP-INET-import;
                        neighbor 10.96.203.2 {
                            local-address 10.96.203.1;
                            family inet {
                                unicast;
                            }
                            peer-as 65200;
                        }
                    }
                }
            }
        }
        L2VPN {
            instance-type l2vpn;
            interface t1-0/1/3.612;
            route-distinguisher 10.245.14.182:612;
            vrf-import L2VPN-import;
            vrf-export L2VPN-export;
            protocols {# There is no graceful-restart statement for Layer 2 VPN instances.
                l2vpn {
                    encapsulation-type frame-relay;
                    site CE2-ISIS {
                        site-identifier 612;
                        interface t1-0/1/3.612 {
                            remote-site-id 512;
                        }
                    }
                }
            }
        }
        OSPF {
            instance-type vrf;
            interface t1-0/1/3.201;
            route-distinguisher 10.245.14.182:201;
            vrf-import OSPF-import;
```

```
            vrf-export OSPF-export;
            routing-options {
                graceful-restart;
            }
            protocols {
                ospf {
                    export OSPF-import;
                    area 0.0.0.0 {
                        interface all;
                    }
                }
            }
        }
        RIP {
            instance-type vrf;
            interface t1-0/1/3.202;
            route-distinguisher 10.245.14.182:202;
            vrf-import RIP-import;
            vrf-export RIP-export;
            routing-options {
                graceful-restart;
            }
            protocols {
                rip {
                    group RIP {
                        export RIP-import;
                        neighbor t1-0/1/3.202;
                    }
                }
            }
        }
        STATIC {
            instance-type vrf;
            interface t1-0/1/3.200;
            route-distinguisher 10.245.14.182:200;
            vrf-import STATIC-import;
            vrf-export STATIC-export;
            routing-options {
                graceful-restart;
                static {
                    route 10.96.210.1/32 next-hop t1-0/1/3.200;
                }
            }
```

```
        }
    }
}
```

Router CE2

On Router CE2, complete the Layer 2 and Layer 3 VPN configuration by mirroring the protocols already set on Routers PE2 and CE1. Specifically, configure the following on the logical interfaces of **t1-0/0/3**: OSPF on unit 201, RIP on unit 202, BGP on unit 203, and IS-IS on unit 612. Finally, configure graceful restart, BGP, IS-IS, OSPF, and RIP on the main instance to be able to connect to the routing instances on Router PE2.

```
[edit]
interfaces {
    t1-0/0/3 {
        encapsulation frame-relay;
        unit 200 {
            dlci 200;
            family inet {
                address 10.96.200.2/30;
            }
        }
        unit 201 {
            dlci 201;
            family inet {
                address 10.96.201.2/30;
            }
        }
        unit 202 {
            dlci 202;
            family inet {
                address 10.96.202.2/30;
            }
        }
        unit 203 {
            dlci 203;
            family inet {
                address 10.96.203.2/30;
            }
        }
        unit 512 {
            dlci 512;
```

```
            family inet {
                address 10.96.252.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.180/32 {
                    primary;
                }
                address 10.96.210.1/32;
                address 10.96.111.1/32;
                address 10.96.212.1/32;
                address 10.96.213.1/32;
                address 10.96.216.1/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4180.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    autonomous-system 65200;
}
protocols {
    bgp {
        group CE-PE-INET {
            type external;
            export BGP_INET_LB_DIRECT;
            neighbor 10.96.203.1 {
                local-address 10.96.203.2;
                family inet {
                    unicast;
                }
                peer-as 65203;
            }
        }
    }
    isis {
        export ISIS_L2VPN_LB_DIRECT;
```

```
        interface t1-0/0/3.612;
    }
    ospf {
        export OSPF_LB_DIRECT;
        area 0.0.0.0 {
            interface t1-0/0/3.201;
        }
    }
    rip {
        group RIP {
            export RIP_LB_DIRECT;
            neighbor t1-0/0/3.202;
        }
    }
}
policy-options {
    policy-statement OSPF_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.201.0/30 exact;
                route-filter 10.96.211.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement RIP_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.202.0/30 exact;
                route-filter 10.96.212.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement BGP_INET_LB_DIRECT {
```

```
        term direct {
            from {
                protocol direct;
                route-filter 10.96.203.0/30 exact;
                route-filter 10.96.213.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement ISIS_L2VPN_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.216.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```

Router PE1 Status Before a Restart

The following example displays neighbor relationships on Router PE1 before a restart happens:

```
user@PE1> show bgp neighbor
Peer: 10.96.103.2+3785 AS 65100 Local: 10.96.103.1+179 AS 65103
  Type: External    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS Refresh>
  Address families configured: inet-unicast
  Local Address: 10.96.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.96.110.1      Local ID: 10.96.103.1      Active Holdtime: 90
  Keepalive Interval: 30
```

```
  Local Interface: t3-0/0/0.103
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI peer can save forwarding state: inet-unicast
  NLRI that peer saved forwarding for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Table BGP-INET.inet.0 Bit: 30001
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            0
    Received prefixes:          0
    Suppressed due to damping:  0
  Last traffic (seconds): Received 8    Sent 3    Checked 3
  Input messages:  Total 15     Updates 0       Refreshes 0     Octets 321
  Output messages: Total 18     Updates 2       Refreshes 0     Octets 450
  Output Queue[2]: 0

Peer: 10.245.14.182+4701 AS 69    Local: 10.245.14.176+179 AS 69
  Type: Internal    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.245.14.176 Holdtime: 90 Preference: 170
  Number of flaps: 1
  Peer ID: 10.245.14.182    Local ID: 10.245.14.176    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
```

```
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
  NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
  NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
  NLRI of all end-of-rib markers sent: inet-vpn-unicast l2vpn
  Table bgp.l3vpn.0 Bit: 10000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            0
    Received prefixes:          0
    Suppressed due to damping:  0
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            1
    Received prefixes:          1
    Suppressed due to damping:  0
  Table BGP-INET.inet.0 Bit: 30000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            0
    Received prefixes:          0
    Suppressed due to damping:  0
  Table OSPF.inet.0 Bit: 60000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            0
    Received prefixes:          0
    Suppressed due to damping:  0
  Table RIP.inet.0 Bit: 70000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            0
    Received prefixes:          0
    Suppressed due to damping:  0
  Table STATIC.inet.0 Bit: 80000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
```

```
    Send state: in sync
    Active prefixes:            0
    Received prefixes:          0
    Suppressed due to damping:  0
  Table L2VPN.l2vpn.0 Bit: 90000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            1
    Received prefixes:          1
    Suppressed due to damping:  0
  Last traffic (seconds): Received 28   Sent 28   Checked 28
  Input messages:  Total 2      Updates 0       Refreshes 0     Octets 86
  Output messages: Total 13     Updates 10      Refreshes 0     Octets 1073
  Output Queue[0]: 0
  Output Queue[1]: 0
  Output Queue[2]: 0
  Output Queue[3]: 0
  Output Queue[4]: 0
  Output Queue[5]: 0
  Output Queue[6]: 0
  Output Queue[7]: 0
  Output Queue[8]: 0

user@PE1> show route instance detail
master:
  Router ID: 10.245.14.176
  Type: forwarding        State: Active
  Restart State: Complete Path selection timeout: 300
  Tables:
    inet.0                : 17 routes (15 active, 0 holddown, 1 hidden)
    Restart Complete
    inet.3                : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    iso.0                 : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
    mpls.0                : 19 routes (19 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l3vpn.0           : 10 routes (10 active, 0 holddown, 0 hidden)
    Restart Complete
    inet6.0               : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l2vpn.0           : 1 routes (1 active, 0 holddown, 0 hidden)
```

```
    Restart Complete
BGP-INET:
  Router ID: 10.96.103.1
  Type: vrf               State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.103
  Route-distinguisher: 10.245.14.176:103
  Vrf-import: [ BGP-INET-import ]
  Vrf-export: [ BGP-INET-export ]
  Tables:
    BGP-INET.inet.0        : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Complete
L2VPN:
  Router ID: 0.0.0.0
  Type: l2vpn             State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.512
  Route-distinguisher: 10.245.14.176:512
  Vrf-import: [ L2VPN-import ]
  Vrf-export: [ L2VPN-export ]
  Tables:
    L2VPN.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
OSPF:
  Router ID: 10.96.101.1
  Type: vrf               State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.101
  Route-distinguisher: 10.245.14.176:101
  Vrf-import: [ OSPF-import ]
  Vrf-export: [ OSPF-export ]
  Tables:
    OSPF.inet.0            : 8 routes (7 active, 0 holddown, 0 hidden)
    Restart Complete
RIP:
  Router ID: 10.96.102.1
  Type: vrf               State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.102
```

```
  Route-distinguisher: 10.245.14.176:102
  Vrf-import: [ RIP-import ]
  Vrf-export: [ RIP-export ]
  Tables:
    RIP.inet.0             : 6 routes (6 active, 0 holddown, 0 hidden)
    Restart Complete
STATIC:
  Router ID: 10.96.100.1
  Type: vrf               State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.100
  Route-distinguisher: 10.245.14.176:100
  Vrf-import: [ STATIC-import ]
  Vrf-export: [ STATIC-export ]
  Tables:
    STATIC.inet.0          : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Complete
__juniper_private1__:
  Router ID: 0.0.0.0
  Type: forwarding        State: Active

user@PE1> show route protocol l2vpn
inet.0: 16 destinations, 17 routes (15 active, 0 holddown, 1 hidden)
Restart Complete
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
BGP-INET.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
Restart Complete
OSPF.inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
Restart Complete
RIP.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
Restart Complete
STATIC.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
mpls.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
800003             *[L2VPN/7] 00:06:00
                    > via t3-0/0/0.512, Pop       Offset: 4
t3-0/0/0.512       *[L2VPN/7] 00:06:00
```

```
                    > via t1-0/1/0.0, Push 800003, Push 100004(top) Offset: -4
bgp.l3vpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Restart Complete
inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
L2VPN.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
10.245.14.176:512:512:611/96
                   *[L2VPN/7] 00:06:01
                      Discard

bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
```

Router PE1 Status During a Restart

Before you can verify that graceful restart is working, you must simulate a router restart. To cause the routing process to refresh and simulate a restart, use the **restart routing** operational mode command:

```
user@PE1> restart routing
Routing protocol daemon started, pid 3558
```

The following sample output is captured during the router restart:

```
user@PE1> show bgp neighbor
Peer: 10.96.103.2     AS 65100 Local: 10.96.103.1     AS 65103
  Type: External    State: Active         Flags: <ImportEval>
  Last State: Idle          Last Event: Start
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS Refresh>
  Address families configured: inet-unicast
  Local Address: 10.96.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
Peer: 10.245.14.182+179 AS 69    Local: 10.245.14.176+2131 AS 69
  Type: Internal    State: Established    Flags: <ImportEval>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.245.14.176 Holdtime: 90 Preference: 170
```

```
  Number of flaps: 0
  Peer ID: 10.245.14.182    Local ID: 10.245.14.176    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
  NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
  NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
  NLRI of received end-of-rib markers: inet-vpn-unicast l2vpn
  Table bgp.l3vpn.0 Bit: 10000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            10
    Received prefixes:          10
    Suppressed due to damping:  0
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            1
    Received prefixes:          1
    Suppressed due to damping:  0
  Table BGP-INET.inet.0 Bit: 30000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            2
    Received prefixes:          2
    Suppressed due to damping:  0
Table OSPF.inet.0 Bit: 60000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            2
    Received prefixes:          2
    Suppressed due to damping:  0
```

```
  Table RIP.inet.0 Bit: 70000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            2
    Received prefixes:          2
    Suppressed due to damping:  0
  Table STATIC.inet.0 Bit: 80000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            1
    Received prefixes:          1
    Suppressed due to damping:  0
  Table L2VPN.l2vpn.0 Bit: 90000
    RIB State: BGP restart is complete
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:            1
    Received prefixes:          1
    Suppressed due to damping:  0
  Last traffic (seconds): Received 0    Sent 0    Checked 0
  Input messages:  Total 14     Updates 13      Refreshes 0     Octets 1053
  Output messages: Total 3      Updates 0       Refreshes 0     Octets 105
  Output Queue[0]: 0
  Output Queue[1]: 0
  Output Queue[2]: 0
  Output Queue[3]: 0
  Output Queue[4]: 0
  Output Queue[5]: 0
  Output Queue[6]: 0
  Output Queue[7]: 0
  Output Queue[8]: 0

user@PE1> show route instance detail
master:
  Router ID: 10.245.14.176
  Type: forwarding        State: Active
  Restart State: Pending  Path selection timeout: 300
  Tables:
    inet.0                 : 17 routes (15 active, 1 holddown, 1 hidden)
    Restart Pending: OSPF LDP
    inet.3                 : 2 routes (2 active, 0 holddown, 0 hidden)
```

```
    Restart Pending: OSPF LDP
    iso.0                 : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
    mpls.0                : 23 routes (23 active, 0 holddown, 0 hidden)
    Restart Pending: LDP VPN
    bgp.l3vpn.0           : 10 routes (10 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
    inet6.0               : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l2vpn.0           : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
BGP-INET:
  Router ID: 10.96.103.1
  Type: vrf               State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.103
  Route-distinguisher: 10.245.14.176:103
  Vrf-import: [ BGP-INET-import ]
  Vrf-export: [ BGP-INET-export ]
  Tables:
    BGP-INET.inet.0       : 6 routes (5 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
L2VPN:
  Router ID: 0.0.0.0
  Type: l2vpn             State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.512
  Route-distinguisher: 10.245.14.176:512
  Vrf-import: [ L2VPN-import ]
  Vrf-export: [ L2VPN-export ]
  Tables:
    L2VPN.l2vpn.0         : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: VPN L2VPN
OSPF:
  Router ID: 10.96.101.1
  Type: vrf               State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.101
  Route-distinguisher: 10.245.14.176:101
  Vrf-import: [ OSPF-import ]
```

```
  Vrf-export: [ OSPF-export ]
  Tables:
    OSPF.inet.0            : 8 routes (7 active, 1 holddown, 0 hidden)
    Restart Pending: OSPF VPN
RIP:
  Router ID: 10.96.102.1
  Type: vrf               State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.102
  Route-distinguisher: 10.245.14.176:102
  Vrf-import: [ RIP-import ]
  Vrf-export: [ RIP-export ]
  Tables:
    RIP.inet.0             : 8 routes (6 active, 2 holddown, 0 hidden)
    Restart Pending: RIP VPN
STATIC:
  Router ID: 10.96.100.1
  Type: vrf               State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.100
  Route-distinguisher: 10.245.14.176:100
  Vrf-import: [ STATIC-import ]
  Vrf-export: [ STATIC-export ]
  Tables:
    STATIC.inet.0          : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
__juniper_private1__:
  Router ID: 0.0.0.0
  Type: forwarding        State: Active
```

```
user@PE1> show route instance summary
Instance           Type           Primary rib            Active/holddown/hidden
master             forwarding
                                  inet.0                 15/0/1
                                  iso.0                  1/0/0
                                  mpls.0                 35/0/0
                                  l3vpn.0                0/0/0
                                  inet6.0                2/0/0
                                  l2vpn.0                0/0/0
```

```
                                l2circuit.0             0/0/0
BGP-INET            vrf
                                BGP-INET.inet.0         5/0/0
                                BGP-INET.iso.0          0/0/0
                                BGP-INET.inet6.0        0/0/0
L2VPN               l2vpn
                                L2VPN.inet.0            0/0/0
                                L2VPN.iso.0             0/0/0
                                L2VPN.inet6.0           0/0/0
                                L2VPN.l2vpn.0           2/0/0
OSPF                vrf
                                OSPF.inet.0             7/0/0
                                OSPF.iso.0              0/0/0
                                OSPF.inet6.0            0/0/0
RIP                 vrf
                                RIP.inet.0              6/0/0
                                RIP.iso.0               0/0/0
                                RIP.inet6.0             0/0/0
STATIC              vrf
                                STATIC.inet.0           4/0/0
                                STATIC.iso.0            0/0/0
                                STATIC.inet6.0          0/0/0
__juniper_private1__ forwarding
                                __juniper_priva.inet.0 0/0/0
                                __juniper_privat.iso.0 0/0/0
                                __juniper_priv.inet6.0 0/0/0

user@PE1> show route protocol l2vpn

inet.0: 16 destinations, 17 routes (15 active, 1 holddown, 1 hidden)
Restart Pending: OSPF LDP

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Pending: OSPF LDP

BGP-INET.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
Restart Pending: VPN

OSPF.inet.0: 7 destinations, 8 routes (7 active, 1 holddown, 0 hidden)
Restart Pending: OSPF VPN

RIP.inet.0: 6 destinations, 8 routes (6 active, 2 holddown, 0 hidden)
Restart Pending: RIP VPN
```

```
STATIC.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Pending: VPN

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
Restart Pending: LDP VPN
+ = Active Route, - = Last Active, * = Both

800001             *[L2VPN/7] 00:00:13
                    > via t3-0/0/0.512, Pop       Offset: 4
t3-0/0/0.512       *[L2VPN/7] 00:00:13
                    > via t1-0/1/0.0, Push 800003, Push 100004(top) Offset: -4

bgp.l3vpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Restart Pending: BGP VPN

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

L2VPN.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Pending: VPN L2VPN
+ = Active Route, - = Last Active, * = Both

10.245.14.176:512:512:611/96
                   *[L2VPN/7] 00:00:13
                      Discard
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Pending: BGP VPN
```

Configuring VPN Graceful Restart

IN THIS SECTION

Graceful restart allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. Without graceful restart, a control plane restart disrupts any VPN services provided by the router. Graceful restart is supported on Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, and VPLS.

To implement graceful restart for a Layer 2 VPN or Layer 3 VPN, perform the configuration tasks described in the following sections:

Configuring Graceful Restart Globally

To enable graceful restart, include the `graceful-restart` statement at the `[edit routing-options]` hierarchy level. To configure a global duration for the graceful restart period, include the `restart-duration` statement at the `[edit routing-options graceful-restart]` hierarchy level.

```
[edit]
routing-options {
    graceful-restart {
        disable;
        restart-duration seconds;
    }
}
```

To disable graceful restart globally, include the `disable` statement at the `[edit routing-options graceful-restart]` hierarchy level.

Configuring Graceful Restart for the Routing Instance

For Layer 3 VPNs only, you must also configure graceful restart for all routing and MPLS-related protocols within a routing instance by including the `graceful-restart` statement at the `[edit routing-instances` *`instance-name`* `routing-options]` hierarchy level. Because you can configure multi-instance BGP and multi-instance LDP, graceful restart for a carrier-of-carriers scenario is supported. To configure the duration of the graceful restart period for the routing instance, include the `restart-duration` statement at the `[edit routing-instances` *`instance-name`* `routing-options]`.

```
[edit]
routing-instances {
    instance-name {
```

```
        routing-options {
            graceful-restart {
                disable;
                restart-duration seconds;
            }
        }
    }
}
```

You can disable graceful restart for individual protocols with the `disable` statement at the `[edit routing-instances instance-name protocols protocol-name graceful-restart]` hierarchy level.

Configuring Logical System Graceful Restart

IN THIS SECTION

Graceful restart for a logical system functions much as graceful restart does in the main router. The only difference is the location of the `graceful-restart` statement.

The following topics describe what to configure to implement graceful restart in a logical system:

Enabling Graceful Restart Globally

To enable graceful restart in a logical system, include the `graceful-restart` statement at the `[edit logical-systems logical-system-name routing-options]` hierarchy level. To configure a global duration of the graceful restart period, include the `restart-duration` statement at the `[edit logical-systems logical-system-name routing-options graceful-restart]` hierarchy level.

```
[edit]
logical-systems {
    logical-system-name {
        routing-options {
            graceful-restart {
                disable;
```

```
                restart-duration seconds;
            }
        }
    }
}
```

To disable graceful restart globally, include the `disable` statement at the `[edit logical-systems logical-system-name routing-options graceful-restart]` hierarchy level.

Configuring Graceful Restart for a Routing Instance

For Layer 3 VPNs only, you must also configure graceful restart globally for a routing instance inside a logical system. To configure, include the `graceful-restart` statement at the `[edit logical-systems logical-system-name routing-instances instance-name routing-options]` hierarchy level. Because you can configure multi-instance BGP and multi-instance LDP, graceful restart for a carrier-of-carriers scenario is supported. To configure the duration of the graceful restart period for the routing instance, include the `restart-duration` statement at the `[edit logical-systems logical-system-name routing-instances instance-name routing-options]`.

```
[edit]
logical-systems {
    logical-system-name {
        routing-instances {
            instance-name {
                routing-options {
                    graceful-restart {
                        disable;
                        restart-duration seconds;
                    }
                }
            }
        }
    }
}
```

To disable graceful restart for individual protocols with the `disable` statement at the `[edit logical-systems logical-system-name routing-instances instance-name protocols protocol-name graceful-restart]` hierarchy level.

Configuring Graceful Restart for QFabric Systems

IN THIS SECTION

When you configure graceful restart in the QFabric CLI, the QFabric system applies the configuration to the network Node group to participate in graceful restart operations with devices external to the QFabric system. Such configuration preserves routing table state and helps neighboring routing devices to resume routing operations more quickly after a system restart. This also enables the network Node group to resume routing operations rapidly if there is a restart in the QFabric system (such as a software upgrade). As a result, we recommend enabling graceful restart for routing protocols in the QFabric CLI.

NOTE: The QFabric system also uses graceful restart internally within the fabric to facilitate interfabric resiliency and recovery. This internal feature is enabled by default with no configuration required.

Enabling Graceful Restart

By default, graceful restart is disabled. To enable graceful restart, include the **graceful-restart** statement at the **[edit routing-instance *instance-name* routing-options]** or **[edit routing-options]** hierarchy level.

For example:

```
routing-options {
    graceful-restart;
}
```

To configure the duration of the graceful restart period, include the **restart-duration** at the **[edit routing-options graceful-restart]** hierarchy level.

NOTE: Helper mode (the ability to assist a neighboring router attempting a graceful restart) is enabled by default when you start the routing platform, even if graceful restart is not enabled. You can disable helper mode on a per-protocol basis.

```
[edit]
routing-options {
    graceful-restart {
        disable;
        restart-duration seconds;
    }
}
```

To disable graceful restart globally, include the **disable** statement at the **[edit routing-options graceful-restart]** hierarchy level.

When graceful restart is enabled for all routing protocols at the **[edit routing-options graceful-restart]** hierarchy level, you can disable graceful restart on a per-protocol basis.

NOTE: If you configure graceful restart after a BGP or LDP session has been established, the BGP or LDP session restarts and the peers negotiate graceful restart capabilities. Also, the BGP peer routing statistics are reset to zero.

Configuring Graceful Restart Options for BGP

To configure the duration of the BGP graceful restart period, include the **restart-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level. To set the length of time the router waits to receive messages from restarting neighbors before declaring them down, include the **stale-routes-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level.

```
[edit]
protocols {
    bgp {
        graceful-restart {
            disable;
            restart-time seconds;
            stale-routes-time seconds;
        }
    }
}
```

```
routing-options {
    graceful-restart;
}
```

To disable BGP graceful restart capability for all BGP sessions, include the **disable** statement at the **[edit protocols bgp graceful-restart]** hierarchy level.

NOTE: To set BGP graceful restart properties or disable them for a group, include the desired statements at the **[edit protocols bgp group *group-name* graceful-restart]** hierarchy level.

To set BGP graceful restart properties or disable them for a specific neighbor in a group, include the desired statements at the **[edit protocols bgp group *group-name* neighbor *ip-address* graceful-restart]** hierarchy level.

NOTE: Configuring graceful restart for BGP resets the BGP peer routing statistics to zero. Also, existing BGP sessions restart, and the peers negotiate graceful restart capabilities.

Configuring Graceful Restart Options for OSPF and OSPFv3

To configure the duration of the OSPF/OSPFv3 graceful restart period, include the **restart-duration** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level. To specify the length of time for which the router notifies helper routers that it has completed graceful restart, include the **notify-duration** at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level. Strict OSPF link-state advertisement (LSA) checking results in the termination of graceful restart by a helping router. To disable strict LSA checking, include the **no-strict-lsa-checking** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level.

```
[edit]
protocols {
    ospf | ospfv3{
        graceful-restart {
            disable;
            helper-disable
            no-strict-lsa-checking;
            notify-duration seconds;
            restart-duration seconds;
        }
    }
}
```

```
routing-options {
    graceful-restart;
}
```

To disable OSPF/OSPFv3 graceful restart, include the **disable** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level.

Starting with Release 11.3, the Junos OS supports both the standard (based on RFC 3623, *Graceful OSPF Restart*) and the restart signaling-based (as specified in RFC 4811, RFC 4812, and RFC 4813) helper modes for OSPF version 2 graceful restart configurations. Both the standard and restart signaling-based helper modes are enabled by default. To disable the helper mode for OSPF version 2 graceful restart configurations, include the **helper-disable <both | restart-signaling | standard>** statement at the **[edit protocols ospf graceful-restart]** hierarchy level. Note that the last committed statement always takes precedence over the previous one.

```
[edit protocols ospf]
    graceful-restart {
        helper-disable <both | restart-signaling | standard>
    }
```

To reenable the helper mode, delete the **helper-disable** statement from the configuration by using the **delete protocols ospf graceful-restarthelper-disable <restart-signaling | standard | both>** command. In this case also, the last executed command takes precedence over the previous ones.

NOTE:
Restart signaling-based helper mode is not supported for OSPFv3 configurations. To disable helper mode for OSPFv3 configurations, include the **helper-disable** statement at the **[edit protocols ospfv3 graceful-restart]** hierarchy level.

TIP: You can also track graceful restart events with the **traceoptions** statement at the **[edit protocols (ospf | ospf3)]** hierarchy level. For more information, see "Tracking Graceful Restart Events" on page 345.

NOTE: If you configure BFD and graceful restart for OSPF, graceful restart might not work as expected.

Tracking Graceful Restart Events

To track the progress of a graceful restart event, you can configure graceful restart trace options flags for IS-IS and OSPF/OSPFv3. To configure graceful restart trace options, include the **graceful-restart** statement at the **[edit protocols *protocol* traceoptions flag]** hierarchy level:

```
[edit protocols]
isis {
    traceoptions {
        flag graceful-restart;
    }
}
(ospf | ospf3) {
    traceoptions {
        flag graceful-restart;
    }
}
```

Example: Managing Helper Modes for OSPF Graceful Restart

IN THIS SECTION

Configuration

Step-by-Step Procedure

Both standard and restart signaling-based helper modes are enabled by default, irrespective of the graceful-restart configuration status on the routing device. Junos OS allows you to disable or enable the helper modes based on your requirements.

To configure the helper mode options for graceful restart:

1. To enable graceful restart, add the `graceful-restart` statement at the `[edit routing-options]` hierarchy level.

```
[edit routing-options]
user@host# set graceful-restart
```

   The helper modes, both standard and restart signaling-based, are enabled by default.

2. To disable one or both of the helper modes, add the `helper-disable <both | restart-signaling | standard>` statement at the `[edit protocols ospf graceful-restart]` hierarchy level.

   - To disable both standard and restart signaling-based helper modes:

```
[edit protocols ospf graceful-restart]
user@host# set helper-disable both
```

   - To disable only the restart signaling-based helper mode:

```
[edit protocols ospf graceful-restart]
user@host# set helper-disable restart-signaling
```

   - To disable only the standard helper mode:

```
[edit protocols ospf graceful-restart]
user@host# set helper-disable standard
```

NOTE: You must commit the configuration before the change takes effect.
The last committed statement always takes precedence over the previous one.

3. To enable one or both of the helper modes when the helper modes are disabled, delete the `helper-disable <both | restart-signaling | standard>` statement from the `[edit protocols ospf graceful-restart]` hierarchy level.

   - To enable both standard and restart signaling-based helper modes:

```
[edit protocols ospf graceful-restart]
user@host# delete helper-disable
```

- To enable the restart signaling-based helper mode:

```
[edit protocols ospf graceful-restart]
user@host# delete helper-disable restart-signaling
```

- To enable the standard helper mode:

```
[edit protocols ospf graceful-restart]
user@host# delete helper-disable standard
```

NOTE: You must commit the configuration before the change takes effect.
The last committed statement always takes precedence over the previous one.

Requirements

M Series or T Series routers running Junos OS Release 11.4 or later and EX Series switches.

Overview

Junos OS Release 11.4 extends OSPF graceful restart support to include restart signaling-based helper mode. Both standard (RFC 3623-based) and restart signaling-based helper modes are enabled by default, irrespective of the graceful-restart configuration status on the routing device.

Junos OS, however, enables you to choose between the helper modes with the `helper-disable <standard | restart-signaling | both>` statement.

Verification

IN THIS SECTION

- Verifying OSPF Graceful Restart and Helper Mode Configuration | 348

Confirm that the configuration is working properly.

Verifying OSPF Graceful Restart and Helper Mode Configuration

Purpose

Verify the OSPF graceful restart and helper mode configuration on a router.

Action

- Enter the `run show ospf overview` command from configuration mode.

```
user@host# run show ospf overview

  ~
  ~
  ~
  Restart: Enabled
    Restart duration: 180 sec
    Restart grace period: 210 sec
    Graceful restart helper mode: Enabled
    Restart-signaling helper mode: Enabled
  ~
  ~
  ~
```

Meaning

The output shows that graceful restart and both of the helper modes are enabled.

Tracing Restart Signaling-Based Helper Mode Events for OSPF Graceful Restart

Junos OS provides a tracing option to log restart signaling-based helper mode events for OSPF graceful restart. To enable tracing for restart signaling-based helper mode events, include the `traceoptions flag restart-signaling` statement at the `[edit protocols ospf]` hierarchy level.

To enable tracing for restart signaling-based events:

1. Create a log file for saving the log.

```
[edit protocols ospf]
user@host# set traceoptions file ospf-log
```

   where *ospf-log* is the name of the log file.

2. Enable tracing for restart signaling-based helper mode events.

```
[edit protocols ospf]
user@host# set traceoptions flag restart-signaling
```

3. Commit the configuration.

```
[edit protocols ospf]
user@host# commit
```

The logs are saved to the *ospf-log* file in the **/var/log** folder.

Viewing the Log File

To view the restart signaling-based events from the log file, type:

```
user@host> file show /var/log/ospf-log | match “restart signaling”
Jun 25 14:44:08.890216 OSPF Restart Signaling: Start helper mode for nbr ip 14.19.3.2 id
10.10.10.1
Jun 25 14:44:11.358636 OSPF restart signaling: Received DBD with R bit set from nbr ip=14.19.3.2
id=10.10.10.1. Start oob-resync.
Jun 25 14:44:11.380198 OSPF restart signaling: Received DBD with LR bit on from nbr ip=14.19.3.2
id=10.10.10.1. Save its oob-resync capability 1
Jun 25 14:44:11.467200 OSPF restart signaling: nbr fsm for nbr ip=14.19.3.2 id=10.10.10.1 moving
to state Full. Reset oob-resync parameters.
```

Verifying Graceful Restart Operation

IN THIS SECTION

This topic contains the following sections:

Graceful Restart Operational Mode Commands

To verify proper operation of graceful restart, use the following commands:

- `show bgp neighbor` (for BGP graceful restart)
- `show log` (for IS-IS and OSPF/OSPFv3 graceful restart)
- `show (ospf | ospfv3) overview` (for OSPF/OSPFv3 graceful restart)
- `show rsvp neighbor detail` (for RSVP graceful restart—helper router)
- `show rsvp version` (for RSVP graceful restart—restarting router)
- `show ldp session detail` (for LDP graceful restart)
- `show connections` (for CCC and TCC graceful restart)
- `show route instance detail` (for Layer 3 VPN graceful restart and for any protocols using graceful restart in a routing instance)
- `show route protocol l2vpn` (for Layer 2 VPN graceful restart)

For more information about these commands and a description of their output fields, see the CLI Explorer.

Verifying BGP Graceful Restart

To view graceful restart information for BGP sessions, use the `show bgp neighbor` command:

```
user@PE1> show bgp neighbor 192.0.2.10
Peer: 192.0.2.10+179 AS 64496 Local: 192.0.2.5+1106 AS 64496
  Type: Internal    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ static ]
  Options:<Preference LocalAddress HoldTime GracefulRestart Damping PeerAS Refresh>
  Local Address: 192.0.2.5 Holdtime: 90 Preference: 170
  IPSec SA Name: hope
  Number of flaps: 0
  Peer ID: 192.0.2.10     Local ID: 192.0.2.5      Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 180
  Stale routes from peer are kept for: 180
  Restart time requested by this peer: 300
  NLRI that peer supports restart for: inet-unicast
  NLRI that peer saved forwarding for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Table inet.0 Bit: 10000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 0
    Suppressed due to damping: 0
  Last traffic (seconds): Received 19   Sent 19   Checked 19
  Input messages:  Total 2      Updates 1       Refreshes 0     Octets 42
  Output messages: Total 3      Updates 0       Refreshes 0     Octets 116
  Output Queue[0]: 0
```

Verifying IS-IS and OSPF Graceful Restart

To view graceful restart information for IS-IS and OSPF, configure traceoptions (see "Tracking Graceful Restart Events" on page 362).

Here is the output of a traceoptions log from an OSPF restarting router:

```
Oct  8 05:20:12 Restart mode - sending grace lsas
Oct  8 05:20:12 Restart mode - estimated restart duration timer triggered
Oct  8 05:20:13 Restart mode - Sending more grace lsas
```

Here is the output of a traceoptions log from an OSPF helper router:

```
Oct  8 05:20:14 Helper mode for neighbor 192.0.2.5
Oct  8 05:20:14  Received multiple grace lsa from 192.0.2.5
```

Verifying CCC and TCC Graceful Restart

To view graceful restart information for CCC and TCC connections, use the `show connections` command. The following example assumes four remote interface CCC connections between CE1 and CE2:

```
user@PE1> show connections
CCC and TCC connections [Link Monitoring On]
  Legend for status (St)             Legend for connection types
  UN -- uninitialized                if-sw:  interface switching
  NP -- not present                  rmt-if: remote interface switching
  WE -- wrong encapsulation          lsp-sw: LSP switching
  DS -- disabled
  Dn -- down                         Legend for circuit types
  -> -- only outbound conn is up     intf -- interface
  <- -- only inbound  conn is up     tlsp -- transmit LSP
  Up -- operational                  rlsp -- receive LSP
  RmtDn -- remote CCC down
  Restart -- restarting

CCC Graceful restart : Restarting

Connection/Circuit               Type    St      Time last up     # Up trans
CE1-CE2-0                        rmt-if  Restart -----                    0
  fe-1/1/0.0                       intf  Up
  PE1-PE2-0                        tlsp  Up
```

```
  PE2-PE1-0                      rlsp  Up
CE1-CE2-1                      rmt-if  Restart -----                  0
  fe-1/1/0.1                     intf  Up
  PE1-PE2-1                      tlsp  Up
  PE2-PE1-1                      rlsp  Up
CE1-CE2-2                      rmt-if  Restart -----                  0
  fe-1/1/0.2                     intf  Up
  PE1-PE2-2                      tlsp  Up
  PE2-PE1-2                      rlsp  Up
CE1-CE2-3                      rmt-if  Restart -----                  0
  fe-1/1/0.3                     intf  Up
  PE1-PE2-3                      tlsp  Up
  PE2-PE1-3                      rlsp  Up
```

Change History Table

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

Release	Description
15.1	You can, optionally, modify the global settings at the individual protocol level or, as of Junos OS 15.1, at the individual routing instance level.

Configuring Graceful Restart for Routing Protocols

SUMMARY

You can configure graceful restart for routing protocols with the steps below.

IN THIS SECTION

Enabling Graceful Restart

By default, graceful restart is disabled. To enable graceful restart, include the **graceful-restart** statement at the **[edit routing-instance *instance-name* routing-options]** or **[edit routing-options]** hierarchy level.

For example:

```
routing-options {
    graceful-restart;
}
```

To configure the duration of the graceful restart period, include the **restart-duration** at the **[edit routing-options graceful-restart]** hierarchy level.

NOTE: Helper mode (the ability to assist a neighboring router attempting a graceful restart) is enabled by default when you start the routing platform, even if graceful restart is not enabled. You can disable helper mode on a per-protocol basis.

```
[edit]
routing-options {
    graceful-restart {
        disable;
        restart-duration seconds;
```

```
    }
}
```

To disable graceful restart globally, include the **disable** statement at the **[edit routing-options graceful-restart]** hierarchy level.

When graceful restart is enabled for all routing protocols at the **[edit routing-options graceful-restart]** hierarchy level, you can disable graceful restart on a per-protocol basis.

NOTE: If you configure graceful restart after a BGP or LDP session has been established, the BGP or LDP session restarts and the peers negotiate graceful restart capabilities. Also, the BGP peer routing statistics are reset to zero.

Configuring Graceful Restart Options for BGP

To configure the duration of the BGP graceful restart period, include the **restart-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level. To set the length of time the router waits to receive messages from restarting neighbors before declaring them down, include the **stale-routes-time** statement at the **[edit protocols bgp graceful-restart]** hierarchy level.

```
[edit]
protocols {
    bgp {
        graceful-restart {
            disable;
            restart-time seconds;
            stale-routes-time seconds;
        }
    }
}
routing-options {
    graceful-restart;
}
```

To disable BGP graceful restart capability for all BGP sessions, include the **disable** statement at the **[edit protocols bgp graceful-restart]** hierarchy level.

NOTE: To set BGP graceful restart properties or disable them for a group, include the desired statements at the **[edit protocols bgp group *group-name* graceful-restart]** hierarchy level.

To set BGP graceful restart properties or disable them for a specific neighbor in a group, include the desired statements at the **[edit protocols bgp group *group-name* neighbor *ip-address* graceful-restart]** hierarchy level.

NOTE: Configuring graceful restart for BGP resets the BGP peer routing statistics to zero. Also, existing BGP sessions restart, and the peers negotiate graceful restart capabilities.

NOTE: Do not configure both Bidirectional Forwarding Detection (BFD) for BGP and graceful restart for BGP. Routing performance may be sub-optimal if you do this.

Using Control Plane Dependent BFD along with Graceful Restart Helper Mode

When BFD is control plane dependent and the device detects a BFD down event and is not already entering the graceful restart helper mode, this is treated as a regular BFD down event and the device enters the graceful restart helper mode. This behavior makes the control plane dependent BFD unusable in conjunction with graceful restart.

Include the `dont-help-shared-fate-bfd-down` statement at the `[edit protocols bgp graceful-restart]` hierarchy to ensures that the device does not enter the graceful restart helper mode and data traffic continues to be fowarded to an alternate path even if there is an interface failure (without a control plane restart on the BGP neighbor).

```
[edit]
protocols {
    bgp {
        graceful-restart {
            disable;
            dont-help-shared-fate-bfd-down;
            restart-time seconds;
            stale-routes-time seconds;
        }
```

```
    }
}
routing-options {
    graceful-restart;
}
```

You can prevent SRX Series Firewalls from entering the graceful restart helper mode when the device is configured with BFD with a single-hop external BGP (EBGP), by including the `dont-help-shared-fate-bfd-down` statement at the `[edit protocols bgp graceful-restart]` hierarchy.

SEE ALSO

> *dont-help-shared-fate-bfd-down*

Configuring Graceful Restart Options for ES-IS

To configure the duration of the ES-IS graceful restart period, include the `restart-duration` statement at the `[edit protocols esis graceful-restart]` hierarchy level.

```
[edit]
protocols {
    esis {
        graceful-restart {
            disable;
            restart-duration seconds;
        }
    }
}
routing-options {
    graceful-restart;
}
```

To disable ES-IS graceful restart capability, include the `disable` statement at the `[edit protocols esis graceful-restart]` hierarchy level.

Configuring Graceful Restart Options for IS-IS

To configure the duration of the IS-IS graceful restart period, include the `restart-duration` statement at the `[edit protocols isis graceful-restart]` hierarchy level.

```
[edit]
protocols {
    isis {
        graceful-restart {
            disable;
            helper-disable;
            restart-duration seconds;
        }
    }
}
routing-options {
    graceful-restart;
}
```

To disable IS-IS graceful restart helper capability, include the `helper-disable` statement at the `[edit protocols isis graceful-restart]` hierarchy level. To disable IS-IS graceful restart capability, include the `disable` statement at the `[edit protocols isis graceful-restart]` hierarchy level.

NOTE: If adjacencies between the Routing Engine and the neighboring peer 'helper' routers time out, graceful restart protocol extensions are unable to notify the peer 'helper' routers about the impending restart.Graceful restart can then stop and cause interruptions in traffic.

To ensure that these adjacencies are kept, change the hold-time for IS-IS protocols from the default of 27 seconds to a value higher than 40 seconds.

NOTE: You can also track graceful restart events with the `traceoptions` statement at the `[edit protocols isis]` hierarchy level. For more information, see "Tracking Graceful Restart Events" on page 362.

Configuring Graceful Restart Options for OSPF and OSPFv3

To configure the duration of the OSPF/OSPFv3 graceful restart period, include the **restart-duration** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level. To specify the length of time for which the router notifies helper routers that it has completed graceful restart, include the **notify-duration** at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level. Strict OSPF link-state advertisement (LSA) checking results in the termination of graceful restart by a helping router. To disable strict LSA checking, include the **no-strict-lsa-checking** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level.

```
[edit]
protocols {
    ospf | ospfv3{
        graceful-restart {
            disable;
            helper-disable
            no-strict-lsa-checking;
            notify-duration seconds;
            restart-duration seconds;
        }
    }
}
routing-options {
    graceful-restart;
}
```

To disable OSPF/OSPFv3 graceful restart, include the **disable** statement at the **[edit protocols (ospf | ospf3) graceful-restart]** hierarchy level.

Junos OS supports both the standard (based on RFC 3623, *Graceful OSPF Restart*) and the restart signaling-based (as specified in RFC 4811, RFC 4812, and RFC 4813) helper modes for OSPF version 2 graceful restart configurations. Both the standard and restart signaling-based helper modes are enabled by default. To disable the helper mode for OSPF version 2 graceful restart configurations, include the **helper-disable <both | restart-signaling | standard>** statement at the **[edit protocols ospf graceful-restart]** hierarchy level. Note that the last committed statement always takes precedence over the previous one.

```
[edit protocols ospf]
    graceful-restart {
        helper-disable <both | restart-signaling | standard>
    }
```

To reenable the helper mode, delete the **helper-disable** statement from the configuration by using the **delete protocols ospf graceful-restarthelper-disable <restart-signaling | standard | both>** command. In this case also, the last executed command takes precedence over the previous ones.

NOTE:
Restart signaling-based helper mode is not supported for OSPFv3 configurations. To disable helper mode for OSPFv3 configurations, include the `helper-disable` statement at the **[edit protocols ospfv3 graceful-restart]** hierarchy level.

TIP: You can also track graceful restart events with the **traceoptions** statement at the **[edit protocols (ospf | ospf3)]** hierarchy level. For more information, see "Tracking Graceful Restart Events" on page 362.

Configuring Graceful Restart Options for RIP and RIPng

To configure the duration of the RIP or RIPng graceful restart period, include the `restart-time` statement at the `[edit protocols (rip | ripng) graceful-restart]` hierarchy level.

```
[edit]
protocols {
    (rip | ripng) {
        graceful-restart {
            disable;
            restart-time seconds;
        }
    }
}
routing-options {
    graceful-restart;
}
```

To disable RIP or RIPng graceful restart capability, include the `disable` statement at the `[edit protocols (rip | ripng) graceful-restart]` hierarchy level.

Configuring Graceful Restart Options for PIM Sparse Mode

PIM sparse mode continues to forward existing multicast packet streams during a graceful restart, but does not forward new streams until after the restart is complete. After a restart, the routing platform updates the forwarding state with any updates that were received from neighbors and occurred during the restart period. For example, the routing platform relearns the join and prune states of neighbors during the restart, but does not apply the changes to the forwarding table until after the restart.

PIM sparse mode-enabled routing platforms generate a unique 32-bit random number called a generation identifier. Generation identifiers are included by default in PIM hello messages, as specified in the IETF Internet draft *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*. When a routing platform receives PIM hellos containing generation identifiers on a point-to-point interface, Junos OS activates an algorithm that optimizes graceful restart.

Before PIM sparse mode graceful restart occurs, each routing platform creates a generation identifier and sends it to its multicast neighbors. If a PIM sparse mode-enabled routing platform restarts, it creates a new generation identifier and sends it to its neighbors. When a neighbor receives the new identifier, it resends multicast updates to the restarting router to allow it to exit graceful restart efficiently. The restart phase completes when either the PIM state becomes stable or when the restart interval timer expires.

If a routing platform does not support generation identifiers or if PIM is enabled on multipoint interfaces, the PIM sparse mode graceful restart algorithm does not activate, and a default restart timer is used as the restart mechanism.

To configure the duration of the PIM graceful restart period, include the `restart-duration` statement at the `[edit protocols pim graceful-restart]` hierarchy level:

```
[edit]
protocols {
    pim {
        graceful-restart {
            disable;
            restart-duration seconds;
        }
    }
}
routing-options {
    graceful-restart;
}
```

To disable PIM sparse mode graceful restart capability, include the `disable` statement at the `[edit protocols pim graceful-restart]` hierarchy level.

NOTE: Multicast forwarding can be interrupted in two ways. First, if the underlying routing protocol is unstable, multicast reverse-path-forwarding (RPF) checks can fail and cause an interruption. Second, because the forwarding table is not updated during the graceful restart period, new multicast streams are not forwarded until graceful restart is complete.

Tracking Graceful Restart Events

To track the progress of a graceful restart event, you can configure graceful restart trace options flags for IS-IS and OSPF/OSPFv3. To configure graceful restart trace options, include the **graceful-restart** statement at the **[edit protocols *protocol* traceoptions flag]** hierarchy level:

```
[edit protocols]
isis {
    traceoptions {
        flag graceful-restart;
    }
}
(ospf | ospf3) {
    traceoptions {
        flag graceful-restart;
    }
}
```

Configuring Graceful Restart for MPLS-Related Protocols

IN THIS SECTION

This section contains the following topics:

Configuring Graceful Restart Globally

To configure graceful restart globally for all MPLS-related protocols, include the `graceful-restart` statement at the `[edit routing-options]` hierarchy level. To configure the duration of the graceful restart period, include the **restart-duration** at the `[edit routing-options graceful-restart]` hierarchy level:

```
[edit]
routing-options {
    graceful-restart {
        disable;
        restart-duration seconds;
    }
}
```

To disable graceful restart globally, include the `disable` statement at the `[edit routing-options graceful-restart]` hierarchy level.

Configuring Graceful Restart Options for RSVP, CCC, and TCC

Because CCC and TCC rely on RSVP, you must modify these three protocols as a single group.

To configure how long the router retains the state of its RSVP neighbors while they undergo a graceful restart, include the `maximum-helper-recovery-time` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to recover.

To configure the delay between when the router discovers that a neighboring router has gone down and when it declares the neighbor down, include the `maximum-helper-restart-time` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to restart.

```
[edit]
protocols {
    rsvp {
        graceful-restart {
            disable;
            helper-disable;
            maximum-helper-recovery-time;
            maximum-helper-restart-time;
        }
    }
```

```
}
routing-options {
    graceful-restart;
}
```

To disable RSVP, CCC, and TCC graceful restart, include the `disable` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level. To disable RSVP, CCC, and TCC helper capability, include the `helper-disable` statement at the `[edit protocols rsvp graceful-restart]` hierarchy level.

Configuring Graceful Restart Options for LDP

When configuring graceful restart for LDP, you can include the following optional statements at the `[edit protocols ldp graceful-restart]` hierarchy level:

```
[edit protocols ldp graceful-restart]
disable;
helper-disable;
maximum-neighbor-reconnect-time seconds;
maximum-neighbor-recovery-time seconds;
reconnect-time seconds;
recovery-time seconds;

[edit routing-options]
graceful-restart;
```

The statements have the following effects on the graceful restart process:

- To configure the length of time required to reestablish a session after a graceful restart, include the `reconnect-time` statement; the range is 30 through 300 seconds. To limit the maximum reconnect time allowed from a restarting neighbor router, include the `maximum-neighbor-reconnect-time` statement; the range is 30 through 300 seconds.
- To configure the length of time that helper routers are required to maintain the old forwarding state during a graceful restart, include the `recovery-time` statement; the range is 120 through 1800 seconds. On the helper router, you can configure a statement that overrides the request from the restarting router and sets the maximum length of time the helper router will maintain the old forwarding state. To configure this feature, include the `maximum-neighbor-recovery-time` statement; the range is 140 through 1900 seconds.

NOTE: The value for the **recovery-time** and `maximum-neighbor-recovery-time` statements at the `[edit protocols ldp graceful-restart]` hierarchy level should be approximately

80 seconds longer than the value for the `restart-duration` statement at the `[edit routing-options graceful-restart]` hierarchy level. Otherwise, a warning message appears when you try to commit the configuration.

- To disable LDP graceful restart capability, include the `disable` statement. To disable LDP graceful restart helper capability, include the `helper-disable` statement.

RELATED DOCUMENTATION

Configuring Graceful Restart

PART 11

Power Management Overview

Understanding Power Management on EX Series Switches

SUMMARY

Power management on EX series switches helps prevent your switch from being disrupted if there's not enough power for all the switch components.

IN THIS SECTION

The power management feature for Juniper Networks Ethernet Switches helps ensure that normal operation of the system is not disrupted because of insufficient power to the switch. For example:

- Power management ensures that operating line cards continue to receive power if a user installs a new line card in an operating switch when power is insufficient for both the new and existing line cards.
- Power management reserves a certain amount of power to power supply redundancy, so that if a power supply fails, the switch can continue to operate normally. If power management must use some of this reserved power to provide power to switch components, it raises an alarm to indicate that power supply redundancy no longer exists and that normal operations might be disrupted if a power supply fails.
- If power supply failure requires power management to power down some components, it does so gracefully by powering down line cards and PoE ports in the order specified by the user.

Power management manages power to switch components by employing a power budget policy. In its power budget policy, power management:

- Budgets power for each installed switch component that requires power. With the exception of *PoE* power for line cards that support PoE, the amount that power management budgets for each component is the maximum power that component might consume under worst case operating conditions. For example, for the fan tray, power management budgets the amount of power required to run the fans at their maximum speed setting, even if the current fan speed is much lower.
- Reserves a set amount of power for power supply redundancy. In its default configuration, power management manages the switch for *N*+1 power redundancy, which ensures uninterrupted system operation if one power supply fails. For example, if a switch has four online 3000 W power supplies, power management reserves 3000 W in its power budget policy for redundancy. It allocates the remaining 9000 W to normal operating power.

- Specifies the rules under which components receive power. These rules are designed to ensure the least disruption to switch operation under conditions of insufficient power. For example, power management provides power to core system components, such as the Routing Engines, before it provides power to line cards.

You can configure certain aspects of power management's budget policy, specifically:

- The power priority of individual line cards. By assigning different power priorities to the line cards, you can determine which line cards are more likely to receive power in the event of insufficient power.
- The power redundancy configuration. The default power redundancy configuration is *N*+1; you can optionally configure *N*+*N*. For example, if you have deployed two independent AC power feeds to the switch, configure *N*+*N* redundancy. When you configure power management for *N*+*N* redundancy, it reserves the appropriate amount of power in its power budget and reports insufficient power conditions accordingly.

These configurable items are discussed further in:

Power Priority of Line Cards

The power priority of line cards determines:

- The order in which line cards are allocated power
- The order in which line cards that support PoE are allocated power for PoE
- How power is reallocated in cases of changes in power availability or demand in an operating switch

This section covers:

How a Line Card's Power Priority Is Determined

Using the CLI, you can assign a explicit power priority to a line-card slot. If more than one slot has the same assigned priority, the power priority is determined by slot number, with the lowest-numbered slots receiving power first.

Line Card Priority and Line Card Power

Power management allocates power to components according to its power budget policy. After power management has allocated power to the base chassis components, it allocates the remaining available power to the line cards. It powers on the line cards in priority order until all line cards are powered on or the available power (including reserved power, if necessary) is exhausted. Thus if available power is

exhausted before all line cards receive power, higher-priority cards are powered on while lower-priority cards remain powered off.

A lower-priority card might receive power while a higher-priority card does not if the remaining available power is sufficient to power on the lower-priority card but not the higher-priority card. For example, if a line card requiring 450 W is in a higher-priority slot than line card requiring 330 W, the line card requiring 330 W receives the power if there is less than 450 W but more than 330 W remaining in the power budget.

Line cards that have been administratively taken offline are not allocated power.

NOTE: Because power management does not allocate power to a line card that has been administratively taken offline, a line card that has been taken offline is not automatically brought online when you commit a configuration. You must explicitly use the `request chassis fpc slot` *`slot-number`* `online` command to bring a line card online that was taken offline previously. This behavior differs from other platforms running Juniper Networks Junos operating system (Junos OS), which automatically bring an offline FPC online when you commit a configuration.

If power management cannot power on a line card because of insufficient power, it raises a major (red) alarm.

Line Card Priority and PoE Power

After all line cards have been powered on, power management allocates any remaining available power, including reserved power, to the PoE power budgets of line cards that have PoE ports. Power management allocates PoE power to line cards in the order of power priority. If enough power is available, a line card receives its full PoE power budget before power management allocates PoE power to the next highest-priority line card. If not enough power is available, a line card receives partial PoE power and lower-priority line cards receive no PoE power.

If power management is unable to allocate enough power to meet the PoE power budget for a line card, it logs a message to the system log.

The default PoE power budget for a line card is the amount of power needed to supply the maximum supported power to all PoE ports. In cases where powered devices do not require the maximum power or in which some PoE ports are not used for powered devices, you can configure a smaller PoE power budget for a line card. By configuring a smaller PoE power budget, you make more power available for the PoE power budgets of lower-priority line cards.

You can also configure the power priority of the PoE ports on a line card. If power management is unable to allocate enough power to a line card to meet its PoE power budget, the line card PoE controller will turn off power to PoE ports in reverse priority order as required to meet the reduced power allocation.

See *Configuring PoE Interfaces on EX Series Switches* for more information on how to configure the PoE power budget for a line card and how to configure PoE port priorities.

Line Card Priority and Changes in the Power Budget

In an operating switch, power management dynamically reallocates power in response to changes in power availability or demand or changes in line card priority. Power management uses line card priority to determine how to reallocate power in response to the following events:

- A power supply fails, is removed, or is taken offline:
  - If power is insufficient to meet the PoE power allocations of all PoE line cards, power management deallocates PoE power from the line cards in reverse priority order until power is sufficient to meet the remaining PoE power allocations.
  - If power is insufficient to meet the base (non-PoE) power requirements of all the line cards, all PoE power is deallocated. If, after the deallocation of PoE power, power is still not sufficient, power management turns off line cards in reverse priority order until power is sufficient for the remaining line cards.
- A new line card is inserted or a line card is brought online:
  - If the line card supports PoE and there is insufficient power to meet its PoE power budget, PoE power is reallocated from lower-priority line cards. If not enough PoE power can be reallocated from lower-priority line cards, the new line card receives a partial PoE power allocation.
  - If there is insufficient power to power on the new line card, PoE power is removed from PoE line cards in reverse priority order until the new line card can be powered on.
  - If the removal of all PoE power is insufficient to free up enough power to power on the line card, the line card remains powered off and the PoE line cards continue to receive their PoE power allocations. To minimize disruption on an operating switch, lower-priority line cards are not turned off to provide power to the new line card. However, if you restart the switch, power management reruns the current power budget policy and powers line cards on or off based on their priority. As a result, line cards receive power strictly by priority order and previously operating line cards might no longer receive power.
- A new power supply is brought online:
  - Any line cards that were powered off because of insufficient power are powered on in priority order.
  - After all line cards are powered on, remaining power is allocated to the PoE power budgets of line cards in priority order.
- A line card is removed or taken offline, freeing up power:

- Any line cards that were powered down because of insufficient power are powered on in priority order.
- After all line cards are powered on, any remaining power is allocated to the PoE power budgets of line cards in priority order.

- A user changes the assigned power priority of one or more line cards when power is insufficient to meet the power budget:
  - PoE power to the line cards is reallocated based on the new power priorities.
  - Base power allocation to the line cards is not changed—in other words, power management does not power down line cards that had been receiving power because they are now a lower priority. However, if you restart the switch, power management reruns the current power budget policy and powers line cards on or off based on their priority. As a result, line cards receive power strictly by priority order and previously operating line cards might no longer receive power.

If, because of insufficient power, power management reduces or eliminates the PoE power budget for a line card, it logs a message to the system log. If power management must power down a line card because of insufficient power, it raises a major (red) alarm.

Power Supply Redundancy

By default, power management in EX Series switches is configured to manage the power supplies for *N* +1 redundancy, in which one power supply is held in reserve for backup if one of the other power supplies is removed or fails.

You can configure power management to manage the power supplies for *N*+*N* redundancy. In *N*+*N* redundancy, power management holds *N* power supplies in reserve for backup. For example, if your switch has six power supplies and you configure *N*+*N* redundancy, power management makes three power supplies available for normal operating power and reserves three power supplies for redundancy (3+3). If you have an odd number of power supplies, power management allocates one more power supply to normal operating power than to redundant power. For example, if you have five power supplies, the *N*+*N* configuration is 3+2.

Given the same number of power supplies, an *N*+*N* configuration usually provides less normal operating power than an *N*+1 configuration because the *N*+*N* configuration holds more power in reserve for backup. Table 9 on page 372 shows the effect on normal operating power in *N*+1 and *N*+*N* configurations.

Table 9: Available Operating Power in N+1 and N+N Redundancy Configurations

Number of Power Supplies at *n* W Each	Normal Operating Power in *N*+1 Configuration	Normal Operating Power in *N*+*N* Configuration
2	1 x (*n* W)	1 x (*n* W)
3	2 x (*n* W)	2 x (*n* W)
4	3 x (*n* W)	2 x (*n* W)
5	4 x (*n* W)	3 x (*n* W)
6	5 x (*n* W)	3 x (*n* W)

For switches with reduced normal operating power, power management allocates less power to the chassis in an *N*+*N* configuration than in an *N*+1 configuration. This reduction in allocated chassis power allows a switch in an *N*+*N* configuration to power more line cards than it could without the reduction.

NOTE: To achieve the reduction in allocated chassis power, power management reduces the maximum fan speed to 60 percent in an *N*+*N* configuration from 80 percent in an *N* +1 configuration. Because the maximum fan speed is reduced, it is possible that a line card that overheats would be shut down sooner in an *N*+*N* configuration than in an *N*+1 configuration.

Power management automatically recalculates the reserved power and normal operating power as power supplies go online or offline. For example, if you have an *N*+*N* configuration with three online 2000 W power supplies, power management allocates 2000 W to reserved power. If you bring a fourth 2000 W power supply online, power management then allocates 4000 W to reserved power. If a power supply goes offline again, power management once again allocates 2000 W to reserved power.

When power is insufficient to meet the budgeted power requirements, power management raises alarms as follows:

- A minor (yellow) alarm is raised when insufficient power exists to maintain the configured *N*+1 or *N* +*N* power reserves, but all line cards are still receiving their base and PoE power allocations. If this condition persists for 5 minutes, the alarm becomes a major (red) alarm. Even though operation of the switch is unaffected in this condition, you should remedy it as quickly as possible because a power supply failure might cause a disruption in switch operation.

- A major (red) alarm is raised when insufficient power exists to provide all the line cards with their base and PoE power allocations. One or more PoE ports might be down or one or more line cards might be down.

Power management clears all alarms when sufficient power is available to meet normal operating and reserved power requirements.

RELATED DOCUMENTATION

Understand Alarm Types and Severity Levels on EX Series Switches

Understanding Power Management on EX Series Switches | 367

Configuring Power Management

SUMMARY

Follow the steps below to configure power management on your switch.

IN THIS SECTION

Configuring the Power Priority of Line Cards (CLI Procedure)

The power management facility on EX6200 and EX8200 switches allows you to assign power priorities to the slots occupied by line cards. Power management provides power to the slots in priority order, which means that line cards in higher priority slots are more likely to receive power than line cards in lower priority slots if power to the switch is insufficient to power all the line cards.

The power priority you assign to a PoE line card affects both the order in which it receives base power and the order in which it receives PoE power. Base power is allocated first to all line cards in priority order. PoE power is then allocated to the PoE line cards in priority order.

When assigning power priority to slots, keep these points in mind:

- 0 is the highest priority. The number of priority levels depends on the number of slots in a switch—for example, for an EX8208 switch, which has eight slots, you can assign a priority of 0 through 7 to a slot.
- All slots are assigned the lowest priority by default.
- If a group of slots shares the same assigned priority, each slot's power priority within the group is based on its slot number, with the lowest-numbered slots receiving power first. For example, if slot 3 and slot 7 each have an assigned power priority of 2, slot 3 has the higher power priority.
- On EX6200 switches, slots containing a Switch Fabric and Routing Engine (SRE) module are automatically assigned the highest priority. If you assign a priority of 0 to a slot that has a lower number than a slot an SRE module is in, the slot with an SRE module still receives power first. You cannot change the power priority of slot containing an SRE module.

To assign or change the power priority for a slot:

```
[edit chassis]
user@switch# set fpc slot power-budget-priority priority
```

For example, to set slot 6 to priority 0, enter:

```
[edit chassis]
user@switch# set fpc 6 power-budget-priority 0
```

Configuring Power Supply Redundancy (CLI Procedure)

By default, the power management feature in EX Series switches is configured to manage the power supplies for *N*+1 redundancy, in which one power supply is held in reserve for backup if any one of the other power supplies is removed or fails.

You can configure power management to manage the power supplies for *N*+*N* redundancy. For example, to set up your AC power supplies for dual power feed, *N*+*N* redundancy is required. In *N*+*N* redundancy, power management allocates half of the online power supplies to normal operating power and half to redundant power. If you have an odd number of online power supplies, power management allocates one more power supply to normal operating power than to redundant power.

This topic describes how to configure power management for *N*+*N* redundancy and how to revert back to *N*+1 redundancy if your deployment needs change.

Before you configure power management for *N*+*N* redundancy, ensure that you have sufficient power supplies to meet the power requirements of an *N*+*N* configuration. Use the `show chassis power-budget-statistics` command to display your current power budget.

NOTE: To allow more power to be available to line cards in an EX8200 switch, power management compensates for the reduced normal operating power in an *N*+*N* configuration by allocating less power to the chassis than it does in an *N*+1 configuration. For the EX8208 switch, the power allocated to the chassis is reduced to 1200 W from 1600 W. For the EX8216 switch, it is reduced to 1800 W from 2400 W. In determining whether you have enough power for an *N*+*N* configuration, take this reduction of allocated chassis power into account.

The reduction in allocated chassis power is achieved by reducing the maximum fan speed to 60 percent in an *N*+*N* configuration from 80 percent in an *N*+1 configuration. Because the maximum fan speed is reduced, it is possible that a line card that overheats would be shut down sooner in an *N*+*N* configuration than in an *N*+1 configuration.

On EX6200 switches, the same amount of power is allocated for the chassis in *N*+*N* configurations as in *N*+1 configurations.

To configure *N*+*N* redundancy:

```
[edit chassis]user@switch# set psu redundancy n-plus-n
```

To revert back to *N*+1 redundancy:

```
[edit chassis]user@switch# delete chassis psu redundancy n-plus-n
```

RELATED DOCUMENTATION

Understanding Power Management on EX Series Switches | 367

Understanding the EX Series Redundant Power System

SUMMARY

The Redundant Power System (RPS) provides backup power to a switch if the primary power source fails.

IN THIS SECTION

EX Series Redundant Power System Hardware Overview

IN THIS SECTION

You can use the EX Series Redundant Power System (RPS) to provide backup power for Juniper Networks EX2200 Ethernet Switches, (except Juniper Networks EX2200-C Ethernet Switches) and Juniper Networks EX3300 Ethernet Switches that are standalone switches or are members of a *Virtual Chassis*.

Most EX Series switches have a built-in capability for redundant power supplies—therefore, if one power supply fails on those switches, the other power supply takes over. However, EX2200 switches and EX3300 switches have only one internal fixed power supply. If an EX2200 switch or EX3300 switch is

deployed in a critical situation, we recommend that you connect a an RPS to that switch to supply backup power during a loss of power.

RPS is not a primary power supply—it only provides backup power to switches when the single dedicated power supply fails. An RPS operates in parallel with the single dedicated power supplies of the switches connected to it and provides all connected switches enough power to support either Power over Ethernet (PoE) or non-PoE devices when the power supplies on the switches fail.

An RPS can hold up to three power supplies connected to as many as six switches—how that power is allocated is up to you. You determine whether or not to connect switches that provide PoE and you determine which switches have priority. Priority becomes an issue when you connect more than three switches that provide PoE to a fully loaded RPS because a switch providing PoE requires more power than a switch that does not provide PoE. Because a power supply can support only one switch providing PoE, the RPS can become oversubscribed when too many switches that must have enough power for PoE have a power failure.

Benefits of the EX Series Redundant Power System

Provides power backup—You connect up to six EX2200, EX3300, or a combination of these switches and supply power to any three of them.

Protection from high-voltage input and short circuits—RPS provides protection from high-voltage input and short circuits.

Switch Models and Configurations Supported by the RPS

The RPS supports all EX3300 switches and EX2200 switches except EX2200-C switches. You can simultaneously connect any supported switches to the same RPS, whether the switches are standalone switches or are configured in a Virtual Chassis.

All power provided by RPS is either PoE or non-PoE. By default, RPS supports switches that provide PoE. If even one switch provides PoE, then the RPS must be configured to provide enough power for PoE. When enough power for PoE is supplied, one switch can be powered by each power supply. If the switches are not providing PoE power, two switches can be powered by one RPS power supply—you can reconfigure an RPS to provide non-PoE power using a feature called multi-backup.

Table 10 on page 378 lists some possible scenarios and RPS solutions. These examples assume that each RPS is fully loaded with three power supplies.

Table 10: Sample Requirements and RPS Solutions

Switches Requiring Backup	You need this RPS configuration:
Six switches that do not provide PoE to attached devices	One RPS can simultaneously provide power to all six switches if you change the power default to multi-backup—this indicates that no attached switch provides PoE to any devices.
One switch that provides PoE to other devices or two switches that do not provide PoE to any devices	One RPS will always back up all three switches, whether or not they provide PoE to connected devices. Leave the power at the default setting (no multi-backup) and let RPS determine that two switches need only minimum power and one switch provides PoE and therefore needs extra power. RPS automatically supplies the correct level of power.
One EX Series Virtual Chassis member that supplies PoE, one switch that supplies PoE, and one switch that does not supply PoE to any connected devices	One RPS will always back up all three switches. Leave the power default setting (no multi-backup) and let RPS determine that one switch needs only minimum power, one switch needs extra power because it supplies PoE, and the Virtual Chassis member also provides PoE to connected devices.
One switch that supplies PoE and five switches that do not supply PoE	You have two options. Option 1—Use one RPS: Up to three switches that do or do not supply PoE can be backed up simultaneously. You can prioritize the six switches to determine which three are most important if all six fail at once. You must leave the power default setting (no multi-backup) because you have one switch that supplies PoE to attached devices and therefore requires more power. Option 2—Use Two RPSs: In this case, you can connect three switches to each RPS and all switches will be backed up if they all fail at once. Alternatively, you can change the power default to multi-backup on one RPS and connect all five switches that do not supply PoE to that RPS, leaving the other RPS to back up the switch that supplies PoE.
EX Series Virtual Chassis	Use as many RPSs as needed to back up all members of the Virtual Chassis.

When a Switch's Power Supply Fails

Because the power supplies for both EX3300 switches and EX2200 switches are internal, if the switch's power supply fails, you must replace the switch. You should remove or replace a switch with a failed power supply as soon as possible.

Do not try to use an RPS as a primary power supply because an RPS cannot boot or reboot a switch. Each switch connected to the RPS must have its own dedicated power supply and must have booted up using the internal power supply.

If a switch is deployed in a large network center where RPS has a separate source of electricity than the switches it supports, the RPS supplies power when only the switch's electricity fails. In this case, you would not have to replace the switch because the power supply is still functional. The switch will resume using its own internal power supply when electricity to the switch is restored.

Components of the RPS

Table 11 on page 379 lists and describes the components of an RPS:

Table 11: Redundant Power System Components

Component	Value
Power supplies that can be installed	Up to three EX-PWR3-930-AC power supplies. One is included and additional power supplies must be ordered separately.
Switch connector ports on RPS	6 (2 per power supply)
Power cords (for connecting power supplies to the AC power source outlet)	Up to three power cords, one per power supply.
RPS cables (for connecting a switch to a power supply installed in the RPS)	6 (1 for each RPS-to-switch connection). One cable is supplied with the RPS. Additional cables must be ordered separately.

Understanding How Power Priority Is Determined and Set for Switches Connected to the EX Series Redundant Power System

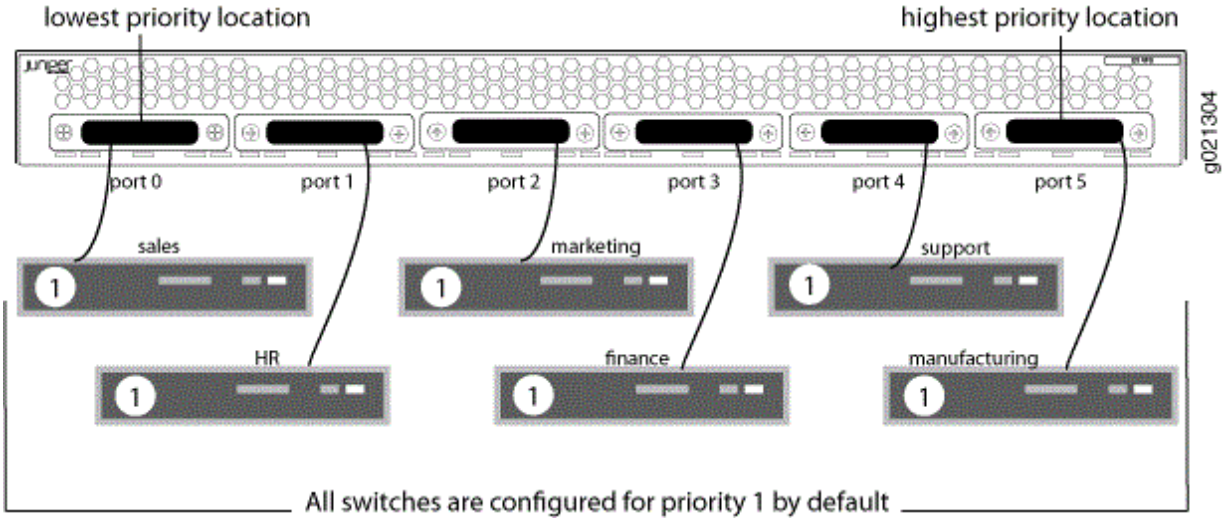
IN THIS SECTION

The Redundant Power System (RPS) is designed to provide backup power to switches that lack built-in redundant power supplies. The RPS provides backup power to switches that either supply power over Ethernet (PoE), which require more power, or switches that do not supply PoE, which require less power. A power supply can either power one PoE device or two non-PoE devices. That means if an RPS is fully loaded with three power supplies, supports PoE switches, and more than three PoE switches have a power failure, some switches will not be powered. You can, however, determine which switches will be powered when an RPS is oversubscribed. When too many connected switches fail, the switches are given power based on their priority. Priority is also reconfigured when any power change takes place. For example, if three switches are already being backed up and another switch has a power failure, the RPS detects this, reconfigures the current top priorities, and allots power accordingly.

Default RPS Priority

While six non-PoE switches can all simultaneously be backed up with three power supplies, only three PoE switches can be backed up (because PoE uses more power). This means that an RPS with four or more PoE switches connected will have to select three of them for backup. You can determine priority by the connector positions you use to connect the switches. By default, an RPS assigns priority to switches based on their switch connector port location, with the leftmost port having the lowest priority and the rightmost port having the highest priority. If the PoE switches shown in Figure 23 on page 381 all fail, the manufacturing, support, and finance switches will be backed up because they are connected to the rightmost connectors.

Figure 23: Default PoE Switch Priority Is Determined by Connector Port Location

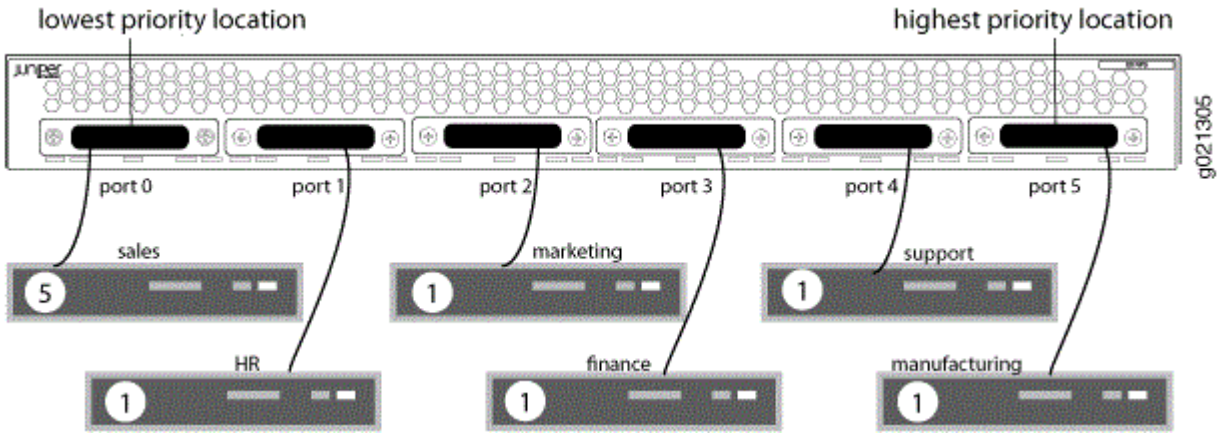


Changing the Priority of Switches on an EX Series RPS

There is a way to alter the priority of PoE switches on an RPS without disconnecting the cables. You can optionally reconfigure any of the attached switches from their CLIs to establish a switch's RPS priority—this CLI configuration overcomes the priority determined by the switch connector port location. Priority ranges from zero (off) to 1 (lowest) through 6 (highest). By default, all switches are configured to 1, the lowest priority. Let's say that the sales switch is reconfigured from the switch's CLI for priority 5 (second highest).

Now in Figure 24 on page 381, with the sales switch configured for RPS 5 from the CLI, the highest priority changes to sales (because 5 is higher than 1), then manufacturing, and then support.

Figure 24: Switch Priority After CLI Configuration



When assigning power priority to switches by using the CLI on the switch, keep these points in mind:

- By default, all switches are assigned priority 1 (lowest) and derive precedence from the location of their connector port on the RPS, with the rightmost port having highest priority.
- Priority 0 assigned from a switch CLI means that the RPS does not provide any backup power to the switch. Essentially, this turns off RPS support.
- Priority 6 assigned from a switch CLI is the highest priority and priority 1 is the lowest priority.
- The CLI command that assigns priority to EX2200 switches is slightly different from the CLI command that assigns priority to EX3300 switches because EX3300 switches can be configured as a *Virtual Chassis*.
- If two or more switches are assigned the same priority value from the switches' CLIs , then the power priority for those switches is determined by the RPS switch connector port location, with the ports to the right receiving priority.
- If a single power supply is installed, the RPS can provide backup power to one switch out of all the switches connected to the RPS. If you do not need any PoE power backup on any switch, you can increase the number of supported switches to two per power supply. Switches connected to an RPS must be either all PoE or all non-PoE.
- The RPS discontinues supplying backup power to a lower-priority switch if it detects a backup power need for a higher-priority switch at the same time.

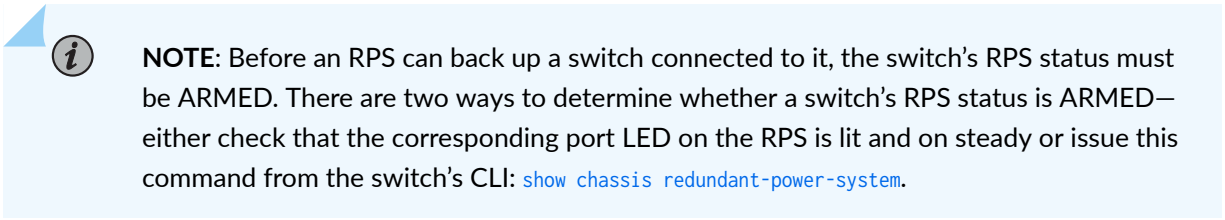
Determining and Setting Priority for Switches Connected to an EX Series RPS

IN THIS SECTION

A Redundant Power System (RPS) provides backup power according to the RPS priority configured on the standalone EX Series switches or Virtual Chassis member switches connected to it. If all switches connected to the RPS are set to the default priority of 1, the priority is determined on the basis of the RPS port to which they are connected, with higher port numbers having the higher priorities.

The number of switches for which an RPS can provide backup power depends on whether the switches provide power over Ethernet (PoE).

- PoE: A fully loaded RPS provides backup power to a maximum of three switches that are enabled for PoE—the result in this case is one switch powered per power supply. If more than three PoE-enabled switches are connected to the RPS and the RPS is already providing backup power to three switches when another switch's power supply fails, the RPS detects this and re-allots backup power as required. It would then stop providing backup power to a low-priority switch to provide backup power to a higher-priority switch.
- Non-PoE: If you changed the RPS power setting to non-PoE with the command `request redundant-power-system multi-backup`, your RPS is configured to provide back up power to as many as six non-PoE switches on a fully loaded RPS. Each power supply can support two switches when the switches do not need enough power for PoE.



NOTE: Before an RPS can back up a switch connected to it, the switch's RPS status must be ARMED. There are two ways to determine whether a switch's RPS status is ARMED—either check that the corresponding port LED on the RPS is lit and on steady or issue this command from the switch's CLI: `show chassis redundant-power-system`.

This topic describes how to determine and set the power priority for a switch connected to an RPS.

Using RPS Default Configuration

No configuration is required on an RPS if you:

- Plan to back up as many as six non-PoE switches
- Back up three PoE switches with three RPS power supplies
- Back up four or more PoE switches with RPS three power supplies and let the RPS port to which the switch is connected determine the priority

By default, an RPS assigns priority to switches on the basis of their switch connector port location, with the with higher port numbers having the higher priorities. By default, all switches are themselves configured with the same RPS priority (priority 1, the lowest), which is why priority is derived from the RPS connector port numbers.

Setting the EX Series RPS Priority for a Switch (CLI)

Each switch connected to RPS has an RPS priority value—that priority value determines which PoE switches receive power first from the RPS. By default, all switches are configured for priority 1 so priority is then determined by switch connector port location, left (lowest) to right (highest).

You can change the priority of a switch to 0 (off), or 1 (lowest) through 6 (highest) from the switch itself —this configuration takes precedence over switch connector port location.

To set or change the priority for a switch that does not support Virtual Chassis:

```
[edit]
user@switch# set redundant-power-system priority
```

To set or change the priority for a switch that supports Virtual Chassis:

```
[edit]
user@switch# set redundant-power-system member vc-member-id priority priority-number
```

Where member is 0 for a switch that has never been configured in a Virtual Chassis.

12

PART

Configuring Virtual Router Redundancy Protocol (VRRP)

Understanding VRRP

SUMMARY

Virtual Router Redundancy Protocol (VRRP) can be used to create virtual redundant routing platforms on a LAN, enabling traffic on the LAN to be routed without relying on a single routing platform.

IN THIS SECTION

Understanding VRRP

For Ethernet, Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, and logical interfaces, you can configure the Virtual Router Redundancy Protocol (VRRP) or VRRP for IPv6. VRRP enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routing platforms share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routing platforms is the primary (active) and the others are backups. If the primary routing platform fails, one of the backup routing platforms becomes the new primary routing platform, providing a virtual default routing platform and enabling traffic on the LAN to be routed without relying on a single routing platform. Using VRRP, a backup device can take over a failed default device within a few seconds. This is done with minimum VRRP traffic and without any interaction with the hosts. Virtual Router Redundancy Protocol is not supported on management interfaces.

Devices running VRRP dynamically elect primary and backup devices. You can also force assignment of primary and backup devices using priorities from 1 through 255, with 255 being the highest priority. In VRRP operation, the default primary device sends advertisements to backup devices at regular intervals. The default interval is 1 second. If a backup device does not receive an advertisement for a set period, the backup device with the next highest priority takes over as primary and begins forwarding packets.

NOTE: Priority 255 cannot be set for routed VLAN interfaces (RVIs).

NOTE: To minimize network traffic, VRRP is designed in such a way that only the device that is acting as the primary sends out VRRP advertisements at any given point in time. The backup devices do not send any advertisement until and unless they take over primary role.

VRRP for IPv6 provides a much faster switchover to an alternate default router than IPv6 neighbor discovery procedures. Typical deployments use only one backup router.

NOTE: Do not confuse the VRRP primary and backup routing platforms with the primary and backup member switches of a *Virtual Chassis* configuration. The primary and backup members of a Virtual Chassis configuration compose a single host. In a VRRP topology, one host operates as the primary routing platform and another operates as the backup routing platform, as shown in Figure 27 on page 390.

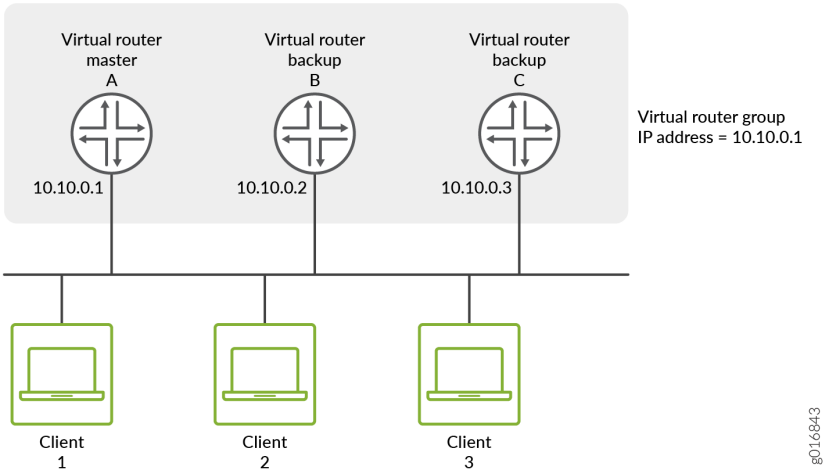
VRRP is defined in RFC 3768, *Virtual Router Redundancy Protocol*. VRRP for IPv6 is defined in draft-ietf-vrrp-ipv6-spec-08.txt, *Virtual Router Redundancy Protocol for IPv6*. See also draft-ietf-vrrp-unified-mib-06.txt, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6*.

NOTE: Even though VRRP, as defined in RFC 3768, does not support authentication, the Junos OS implementation of VRRP supports authentication as defined in RFC 2338. This support is achieved through the backward compatibility options in RFC 3768.

NOTE: On EX2300 and EX3400 switches, the VRRP protocol must be configured with a Hello interval of 2 seconds or more with dead interval not less than 6 seconds to prevent flaps during CPU intensive operations events such as routing engine switchover, interface flaps, and exhaustive data collection from the packet forwarding engine.

Figure 25 on page 388 illustrates a basic VRRP topology. In this example, Routers A, B, and C are running VRRP and together make up a virtual router. The IP address of this virtual router is 10.10.0.1 (the same address as the physical interface of Router A).

Figure 25: Basic VRRP



Because the virtual router uses the IP address of the physical interface of Router A, Router A is the primary VRRP router, while routers B and C function as backup VRRP routers. Clients 1 through 3 are configured with the default gateway IP address of 10.10.0.1. As the primary router, Router A forwards packets sent to its IP address. If the primary virtual router fails, the router configured with the higher priority becomes the primary virtual router and provides uninterrupted service for the LAN hosts. When Router A recovers, it becomes the primary virtual router again.

NOTE: In some cases, during an inherit session, there is a small time frame during which two routers are in Primary-Primary state. In such cases, the VRRP groups that inherit the state do send out VRRP advertisements every 120 seconds. So, it takes the routers up to 120 seconds to recover after moving to Primary-Backup state from Primary-Primary state.

ACX series routers can support up to 64 VRRP group entries. These can be a combination of IPv4 or IPv6 families. If either of the family (IPv4 or IPv6) is solely configured for VRRP, then 64 unique VRRP group identifiers are supported. If both IPv4 and IPv6 families share the same VRRP group, then only 32 unique VRRP identifiers are supported.

NOTE: ACX Series routers support VRRP version 3 for IPv6 addresses.

ACX5448 router supports RFC 3768 VRRP version 2 and RFC 5798 VRRP version 3. ACX5448 router also supports configuring VRRP over aggregated Ethernet and integrated routing and bridging (IRB) interfaces.

The following limitations apply while configuring VRRP on ACX5448 router:

- Configure a maximum of 16 VRRP groups.
- Interworking of VRRP version 2 and VRRP version 3 is not supported.
- VRRP delegate processing is not supported.
- VRRP version 2 authentication is not supported.

Figure 25 on page 388 illustrates a basic VRRP topology with EX Series switches. In this example, Switches A, B, and C are running VRRP and together they make up a virtual routing platform. The IP address of this virtual routing platform is `10.10.0.1` (the same address as the physical interface of Switch A).

Figure 26: Basic VRRP on EX Series Switches

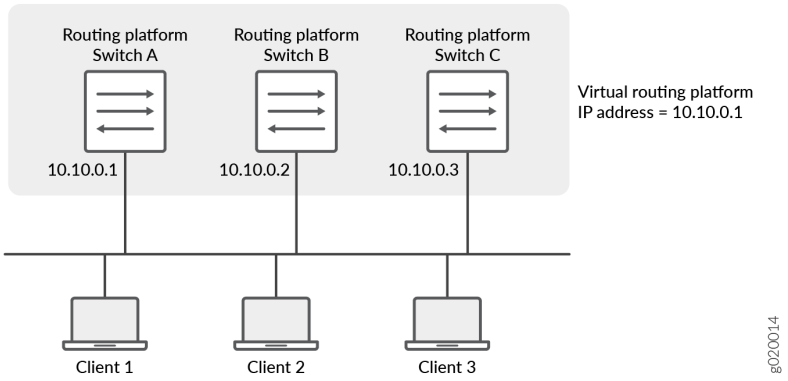
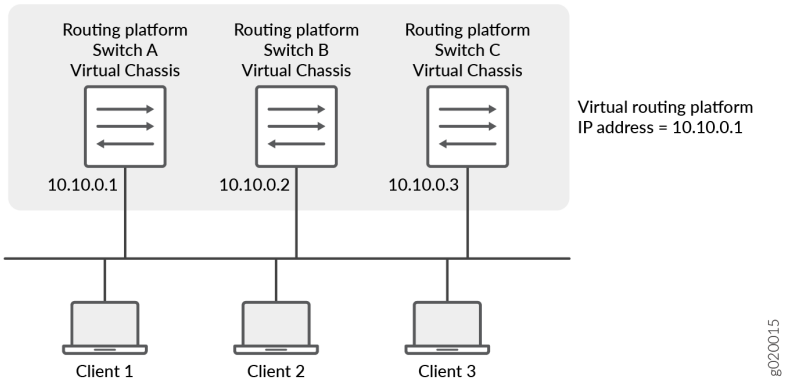


Figure 27 on page 390 illustrates a basic VRRP topology using Virtual Chassis configurations. Switch A, Switch B, and Switch C are each composed of multiple interconnected Juniper Networks EX4200 Ethernet Switches. Each Virtual Chassis configuration operates as a single switch, which is running VRRP, and together they make up a virtual routing platform. The IP address of this virtual routing platform is `10.10.0.1` (the same address as the physical interface of Switch A).

Figure 27: VRRP on Virtual Chassis Switches



Because the virtual routing platform uses the IP address of the physical interface of Switch A, Switch A is the primary VRRP routing platform, while Switch B and Switch C function as backup VRRP routing platforms. Clients 1 through 3 are configured with the default gateway IP address of `10.10.0.1` as the primary router, Switch A, forwards packets sent to its IP address. If the primary routing platform fails, the switch configured with the higher priority becomes the primary virtual routing platform and provides uninterrupted service for the LAN hosts. When Switch A recovers, it becomes the primary virtual routing platform again.

VRRP and VRRP for IPv6 Overview

You can configure the Virtual Router Redundancy Protocol (VRRP) and VRRP for IPv6 for the following interfaces:

- Ethernet
- Fast Ethernet
- Tri-Rate Ethernet copper
- Gigabit Ethernet
- 10-Gigabit Ethernet LAN/WAN PIC
- Ethernet logical interfaces

VRRP and VRRP for IPv6 allow hosts on a LAN to make use of redundant routers on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routers share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routers is the primary (active) and the others are backups. If the primary fails, one of the backup

routers becomes the new primary router, thus always providing a virtual default router and allowing traffic on the LAN to be routed without relying on a single router.

VRRP is defined in RFC 3768, *Virtual Router Redundancy Protocol*.

For VRRP and VRRP for IPv6 overview information, configuration guidelines, and statement summaries, see the Junos OS High Availability User Guide.

Understanding VRRP Between QFabric Systems

IN THIS SECTION

Juniper Networks QFabric systems support the Virtual Router Redundancy Protocol (VRRP). This topic covers:

VRRP Differences on QFabric Systems

Configuring servers on your network with static routes to a default gateway minimizes configuration effort and complexity and reduces processing overhead. However, a failure of the default gateway normally results in a catastrophic event, isolating the servers. Using Virtual Router Redundancy Protocol (VRRP) enables you to dynamically provide alternative gateways for servers if the primary gateway fails.

Switches configured with VRRP share a virtual IP (VIP) address, which is the address you configure as the default route on the servers. In normal VRRP operation, one of the switches is the VRRP primary, meaning that it owns the VIP and is the active default gateway. The other devices are backups. The switches dynamically assign primary and backup roles based on priorities that you configure. If the primary fails, the backup switch with the highest priority becomes the primary and takes ownership of the VIP within a few seconds. This is done without any interaction with the servers.

You can configure two QFabric systems to participate in a VRRP configuration as if they were two standalone switches. However, in normal VRRP operation, only one system can be the primary for a given VRRP group at any one time, which means that only one system can act as a default gateway using the VIP configured for the group. When running VRRP over two QFabric systems, you might want both systems to simultaneously use the VIP to act as a gateway and forward traffic. To achieve this, you can configure a firewall filter to block the VRRP advertisement packets between the QFabric systems on

the link between the two network Node groups. When you do this, both QFabric systems act as primary and forward traffic received by the VIP (which is the default gateway address that you configure on servers connected to both QFabric systems). If you use VMware's vMotion, this configuration allows virtual machines to transition between servers connected to the QFabric systems without updating their default gateway information. For example, a virtual machine running on a server connected to a QFabric system in data center A can transition to a server connected to a QFabric system in data center B without needing to resolve a new gateway IP address and MAC address because both QFabric systems use the same VIP.

NOTE: To use a firewall filter to block VRRP traffic, create a firewall term that matches traffic for `protocol vrrp` and discards that traffic.

Configuration Details

Configuring a VRRP group across two QFabric systems is similar to configuring VRRP on two switches. The main differences are listed here:

- All the interfaces in both QFabric systems that participate in VRRP must be members of the same VLAN.
- You must create routed VLAN interfaces (RVIs) in that VLAN on both QFabric systems.
- The IP addresses that you assign to both RVIs must be in the same subnet.
- You must configure VRRP on the RVIs.
- Both RVIs must be members of the same VRRP group. This is what allows the two QFabric systems to share a virtual IP address.

The following tables list the elements of an example VRRP configuration running on two QFabric systems–QFabric system A and QFabric system B. This example is configured so that both QFabric systems act as the VRRP primary for VIP 10.1.1.50/24 and assumes that a firewall filter blocks the VRRP advertisements between the systems. Table 12 on page 392 lists the required characteristics of the RVIs in the example configuration.

NOTE: Most of the configuration settings in the following tables would also apply in a traditional VRRP configuration. However, the advertisement interval and priority settings would need to be different (as noted).

Table 12: RVIs on QFabric systems in example VRRP configuration

RVI on QFabric System A	RVI on QFabric System B

vlan.100	vlan.200
Member of VLAN 100. (Note that the VLAN is the same on both QFabric systems.)	Member of VLAN 100
IP address 10.1.1.100/24	IP address 10.1.1.200/24
Member of VRRP group 500	Member of VRRP group 500
Virtual IP address 10.1.1.50/24	Virtual IP address 10.1.1.50/24

You must configure VRRP on the RVIs on both QFabric systems.Table 13 on page 393 lists the elements of a sample VRRP configuration on each RVI. Note that with the exception of the priority, the parameters *must* be the same on both systems.

Table 13: Sample VRRP configuration each RVI

VRRP on RVI on QFabric System A	VRRP on RVI on QFabric System B
VRRP group 500	VRRP group 500
Virtual IP address 10.1.1.50/24	Virtual IP address 10.1.1.50/24
Advertisement interval 60 seconds. (In a normal VRRP configuration, you would set this interval to be much smaller, such as 1 second. However, in this configuration these packets are blocked by the firewall filter on the interface that connects to QFabric system B, so there is no need to send them frequently.)	Advertisement interval 60 seconds
Authentication type md5	Authentication type md5
Authentication key $9$1.4ElMVb2aGi4aZjkqzFRhSeWx7-wY2aM8	Authentication key $9$1.4ElMVb2aGi4aZjkqzFRhSeWx7-wY2aM8

Priority 254. (In a normal VRRP configuration, this value would be different on the two systems and the system with the higher value would be the primary. However, in this configuration both systems are acting as primary, so you do not have to configure different values.)	Priority 254

NOTE: Priority 255 is not supported for RVIs.

Table 14 on page 394 lists the all the interfaces on QFabric system A in the example configuration and identifies what they connect to.

Table 14: Interfaces on QFabric system A. All interfaces are members of VLAN 100.

VLAN 100 Interfaces on QFabric System A	Connects To
vlan.100	vlan.200
Network Node group interface QFA-NNG:xe-0/0/0	QFB-NNG:xe-0/0/0 on QFabric system B
Network Node group interface QFA-NNG:xe-0/0/1	Redundant server Node group interface QFA-RSNG:xe-0/0/0
Redundant server Node group interface QFA-RSNG:xe-0/0/0	Connects to a network Node group interface QFA-NNG:xe-0/0/1
Redundant server Node group interface QFA-RSNG:xe-0/0/1	LAN with servers running virtual machines

Table 15 on page 394 lists the all the interfaces on QFabric system B in the example configuration and identifies what they connect to.

Table 15: Interfaces on QFabric system B. All interfaces are members of VLAN 100 (same as on QFabric system A).

VLAN 100 Interfaces on QFabric System B	Connects To
vlan.200	vlan.100

Network Node group interface QFB-NNG:xe-0/0/0	QFA-NNG:xe-0/0/0 on QFabric system A
Network Node group interface QFB-NNG:xe-0/0/1	Redundant server Node group interface QFB-RSNG:xe-0/0/0
Redundant server Node group interface QFB-RSNG:xe-0/0/0	Connects to a network Node group interface QFB-NNG:xe-0/0/1
Redundant server Node group interface QFB-RSNG:xe-0/0/1	LAN with servers running virtual machines

Junos OS Support for VRRPv3

IN THIS SECTION

The advantage of using VRRPv3 is that VRRPv3 supports both IPv4 and IPv6 address families, whereas VRRPv2 supports only IPv4 addresses.

The following topics describe the Junos OS support for and interoperability of VRRPv3, as well as some differences between VRRPv3 and its precursors:

Junos OS VRRP Support

In releases earlier than Release 12.2, Junos OS supported RFC 3768, *Virtual Router Redundancy Protocol (VRRP)* (for IPv4) and Internet draft draft-ietf-vrrp-ipv6-spec-08, *Virtual Router Redundancy Protocol for IPv6*.

VRRPv3 is not supported on routers that use releases earlier than Junos OS Release 12.2 and is also not supported for IPv6 on QFX10000 switches.

NOTE: VRRPv3 for IPv6 is supported on QFX10002-60C.

Starting with Release 12.2, Junos OS supports:

- RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*
- RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6*
- RFC 6527, *Definitions of Managed Objects for Virtual Router Redundancy Protocol Version 3 (VRRPv3)*

NOTE: VRRP (for IPv6) on routers that use Junos OS Release 12.2 and later releases does not interoperate with VRRP (for IPv6) on routers with earlier Junos OS releases because of the differences in VRRP checksum calculations. See "IPv6 VRRP Checksum Behavioral Differences" on page 396.

IPv6 VRRP Checksum Behavioral Differences

You must consider the following checksum differences when enabling IPv6 VRRP networks:

- In releases earlier than Junos OS Release 12.2, when VRRP for IPv6 is configured, the VRRP checksum is calculated according to section 5.3.8 of RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*.
- Starting with Junos OS Release 12.2, when VRRP for IPv6 is configured, irrespective of VRRPv3 being enabled or not, the VRRP checksum is calculated according to section 5.2.8 of RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6*.

  Moreover, the pseudoheader is included only when calculating the IPv6 VRRP checksum. The pseudoheader is not included when calculating the IPv4 VRRP checksum.

  To make the router with Junos OS Release 12.2 (or later Junos OS releases) IPv6 VRRP interoperate with the router running a Junos OS release earlier than Release 12.2, include the `checksum-without-pseudoheader` configuration statement at the `[edit protocols vrrp]` hierarchy level in the router running Junos OS Release 12.2 or later.

- The `tcpdump` utility in Junos OS Release 12.2 and later calculates the VRRP checksum according to RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6*. Therefore, when

tcpdump parses IPv6 VRRP packets that are received from older Junos OS releases (earlier than Junos OS Release 12.2), the bad vrrp cksum message is displayed:

```
23:20:32.657328 Out
...
        -----original packet-----
        00:00:5e:00:02:03 > 33:33:00:00:00:12, ethertype IPv6 (0x86dd), length 94: (class
0xc0, hlim 255, next-header: VRRP (112), length: 40) fe80::224:dcff:fe47:57f > ff02::12:
VRRPv3-advertisement 40: vrid=3 prio=100 intvl=100(centisec)  (bad vrrp cksum b4e2!)
addrs(2): fe80::200:5eff:fe00:3,2001:4818:f000:14::1
                        3333 0000 0012 0000 5e00 0203 86dd 6c00
                        0000 0028 70ff fe80 0000 0000 0000 0224
                        dcff fe47 057f ff02 0000 0000 0000 0000
                        0000 0000 0012 3103 6402 0064 b4e2 fe80
                        0000 0000 0000 0200 5eff fe00 0003 2001
                        4818 f000 0014 0000 0000 0000 0001
```

You can ignore this message because it does not indicate VRRP failure.

VRRP Interoperability

In releases earlier than Junos OS Release 12.2, VRRP (IPv6) followed Internet draft draft-ietf-vrrp-ipv6-spec-08, but checksum was calculated based on RFC 3768 section 5.3.8. Starting with Release 12.2, VRRP (IPv6) follows RFC 5798 and checksum is calculated based on RFC 5798 section 5.2.8. Because of the differences in VRRP checksum calculations, IPv6 VRRP configured on routers that use Junos OS Release 12.2 and later releases does not interoperate with IPv6 VRRP configured in releases before Junos OS Release 12.2.

To make the router with Junos OS Release 12.2 (or later Junos OS releases) IPv6 VRRP interoperate with the router running Junos OS releases earlier than Release 12.2, include the checksum-without-pseudoheader configuration statement at the [edit protocols vrrp] hierarchy level in the router with Junos OS Release 12.2 or later.

Here are some general points to know about VRRP interoperability:

- If you have configured VRRPv3 (IPv4 or IPv6) on routers that use Junos OS Release 12.2 or later releases, it will not operate with routers that use Junos OS Release 12.1 or earlier releases. This is because only Junos OS Release 12.2 and later releases support VRRPv3.
- VRRP (IPv4 or IPv6) configured on routers that use Junos OS Release 12.2 and later releases interoperate with VRRP (IPv4 or IPv6) configured on routers that use releases earlier than Junos OS Release 12.2.

- VRRPv3 for IPv4 does not interoperate with the previous versions of VRRP. If VRRPv2 IPv4 advertisement packets are received by a router on which VRRPv3 is enabled, the router transitions itself to the backup state to avoid creating multiple primaries in the network. Due to this behavior, you must be cautious when enabling VRRPv3 on your existing VRRPv2 networks. See "Upgrading from VRRPv2 to VRRPv3" on page 398 for more information.

> **NOTE**: VRRPv3 advertisement packets are ignored by the routers on which previous versions of VRRP are configured.

Upgrading from VRRPv2 to VRRPv3

Enable VRRPv3 in your network only if VRRPv3 can be enabled on all the VRRP routers in your network.

Enable VRRPv3 on your VRRPv2 network only when upgrading from VRRPv2 to VRRPv3. Mixing the two versions of VRRP is not a permanent solution.

> **CAUTION**: VRRP version change is considered catastrophic and disruptive and may not be hitless. The packet loss duration depends on many factors, such as number of VRRP groups, the interfaces and FPCs involved, and the load of other services and protocols running on the router.

Upgrading from VRRPv2 to VRRPv3 must be done very carefully to avoid traffic loss, due to these considerations:

- It is not possible to configure VRRPv3 on all routers simultaneously.
- During the transition period, both VRRPv2 and VRRPv3 operate in the network.
- Changing VRRP versions restarts the state machine for all VRRP groups.
- VRRPv3 (for IPv4) routers default to the backup state when they get VRRPv2 (for IPv4) advertisement packets.
- VRRPv2 (for IPv4) packets are always given the highest priority.
- Checksum differences between VRRPv2 and VRRPv3 (for IPv6) can create multiple primary routers.

  Disable VRRPv3 (for IPv6) on the backup routers while upgrading to avoid creating multiple primary routers.

Table 16 on page 399 illustrates the steps and events that take place during a VRRPv2 to VRRPv3 transition. In Table 16 on page 399, two VRRPv2 routers, R1 and R2, are configured in two groups, G1 and G2. Router R1 acts as the primary for G1, and Router R2 acts as the primary for G2.

Table 16: VRRPv2 to VRRPv3 Transition Steps and Events

1. Upgrade Router R1 with Junos OS Release 12.2 or later.
   - Router R2 becomes the primary for both G1 and G2.
   - After the upgrade of Router R1 is completed, Router R1 becomes the primary for G1.
   - Router R2 remains the primary for G2.
2. Upgrade Router R2 with Junos OS Release 12.2 or later.
   - Router R1 becomes the primary for both G1 and G2.
   - After the upgrade of Router R2 is completed, Router R2 becomes the primary for G2.
   - Router R1 remains the primary for G1.

For IPv4	For IPv6
1. Enable VRRPv3 on Router R1. • Router R1 becomes the backup for both G1 and G2 because VRRPv2 IPv4 advertisement packets are given higher priority. 2. Enable VRRPv3 on Router R2. • Router R1 becomes the primary for G1. • Router R2 becomes the primary for G2.	1. Deactivate G1 and G2 on Router R2. • G1 and G2 on Router R1 become primary. 2. Enable VRRPv3 on Router R1. • Router R1 becomes the primary for both G1 and G2. 3. Enable VRRPv3 on Router R2. 4. Activate G1 and G2 on Router R2. • Router R2 becomes the primary for G2. • Router R1 remains the primary for G1.

When enabling VRRPv3, make sure that VRRPv3 is enabled on all the VRRP routers in the network because VRRPv3 (IPv4) does not interoperate with the previous versions of VRRP. For example, if VRRPv2 IPv4 advertisement packets are received by a router on which VRRPv3 is enabled, the router transitions itself to the backup state to avoid creating multiple primaries in the network.

You can enable VRRPv3 by configuring the **version-3** statement at the `[edit protocols vrrp]` hierarchy level (for IPv4 or IPv6 networks). Configure the same protocol version on all VRRP routers on the LAN.

Functionality of VRRPv3 Features

Some Junos OS features differ in VRRPv3 from previous VRRP versions.

VRRPv3 Authentication

When VRRPv3 (for IPv4) is enabled, it does not allow authentication.

- The `authentication-type` and `authentication-key` statements cannot be configured for any VRRP groups.
- You must use non-VRRP authentication.

VRRPv3 Advertisement Intervals

VRRPv3 (for IPv4 and IPv6) advertisement intervals must be set with the `fast-interval` statement at the `[edit interfaces` *`interface-name`* `unit 0 family inet address` *`ip-address`* `vrrp-group` *`group-name`*`]` hierarchy level.

- Do not use the `advertise-interval` statement (for IPv4).
- Do not use the `inet6-advertise-interval` statement (for IPv6).

Unified ISSU for VRRPv3

Design changes for VRRP unified in-service software upgrade (ISSU) are made in Junos OS Release 15.1 to achieve the following functionality:

- Maintain protocol adjacency with peer routers during unified ISSU. Protocol adjacency created on peer routers for the router undergoing unified ISSU should not flap, which means that VRRP on the remote peer router should not flap.
- Maintain interoperability with competitive or complementary equipment.
- Maintain interoperability with other Junos OS releases and other Juniper Network products.

The values of the following configurations (found at the `[edit interfaces` *`interface-name`* `unit 0 family inet address` *`ip-address`* `vrrp-group` *`group-name`*`]` hierarchy level) need to be kept at maximum values to support unified ISSU:

- On the primary router, the advertisement interval (the `fast-interval` statement) needs to be kept at 40950 milliseconds.
- On the backup router, the primary-down interval (the `advertisements-threshold` statement) needs to be kept at the largest threshold value.

This VRRP unified ISSU design only works for VRRPv3. It is not supported on VRRPv1 or VRRPv2. Other limitations include the following:

- The VRRP unified ISSU takes care of VRRP only. Packet forwarding is the responsibility of the Packet Forwarding Engine. The Packet Forwarding Engine unified ISSU should ensure uninterrupted traffic flow.
- VRRP is not affected by any change event during unified ISSU, for example, the switchover of the primary Routing Engine to backup or the backup Routing Engine to primary.
- VRRP might stop and discard any running timer before entering into unified ISSU. This means the expected action upon the expiry of the timer never takes place. However, you can defer unified ISSU until the expiration of all running timers.
- Unified ISSU at both local and remote routers cannot be done simultaneously.

VRRP failover-delay Overview

IN THIS SECTION

Failover is a backup operational mode in which the functions of a network device are assumed by a secondary device when the primary device becomes unavailable because of a failure or a scheduled down time. Failover is typically an integral part of mission-critical systems that must be constantly available on the network.

VRRP does not support session synchronization between members. If the primary device fails, the backup device with the highest priority takes over as primary and will begin forwarding packets. Any existing sessions will be dropped on the backup device as out-of-state.

A fast failover requires a short delay. Thus, failover-delay configures the failover delay time, in milliseconds, for VRRP and VRRP for IPv6 operations. Junos OS supports a range of 50 through 100000 milliseconds for delay in failover time.

The VRRP process (vrrpd) running on the Routing Engine communicates a VRRP primary role change to the Packet Forwarding Engine for every VRRP session. Each VRRP group can trigger such communication to update the Packet Forwarding Engine with its own state or the state inherited form an active VRRP group. To avoid overloading the Packet Forwarding Engine with such messages, you can configure a failover-delay to specify the delay between subsequent Routing Engine to Packet Forwarding Engine communications.

The Routing Engine communicates a VRRP primary role change to the Packet Forwarding Engine to facilitate necessary state change on the Packet Forwarding Engine, such as reprogramming of Packet Forwarding Engine hardware filters, VRRP sessions and so on. The following sections elaborate the Routing Engine to Packet Forwarding Engine communication in two scenarios:

When failover-delay Is Not Configured

Without failover-delay configured, the sequence of events for VRRP sessions operated from the Routing Engine is as follows:

1. When the first VRRP group detected by the Routing Engine changes state, and the new state is primary, the Routing Engine generates appropriate VRRP announcement messages. The Packet Forwarding Engine is informed about the state change, so that hardware filters for that group are reprogrammed without delay. The new primary then sends gratuitous ARP message to the VRRP groups.
2. The delay in failover timer starts. By default, failover-delay timer is:
   - 500 miliseconds—when the configured VRRP announcement interval is less than 1 second.
   - 2 seconds—when the configured VRRP announcement interval is 1 second or more, and the total number of VRRP groups on the router is 255.
   - 10 seconds—when the configured VRRP announcement interval is 1 second or more, and the number of VRRP groups on the router is more than 255.
3. The Routing Engine performs one-by-one state change for subsequent VRRP groups. Every time there is a state change, and the new state for a particular VRRP group is primary, the Routing Engine generates appropriate VRRP announcement messages. However, communication toward the Packet Forwarding Engine is suppressed until the failover-delay timer expires.
4. After failover-delay timer expires, the Routing Engine sends message to the Packet Forwarding Engine about all VRRP groups that managed to change the state. As a consequence, hardware filters for those groups are reprogrammed, and for those groups whose new state is primary, gratuitous ARP messages are sent.

This process repeats until state transition for all VRRP groups is complete.

Thus, without configuring failover-delay, the full state transition (including states on the Routing Engine and the Packet Forwarding Engine) for the first VRRP group is performed immediately, while state transition on the Packet Forwarding Engine for remaining VRRP groups is delayed by at least 0.5-10 seconds, depending on the configured VRRP announcement timers and the number of VRRP groups. During this intermediate state, receiving traffic for VRRP groups for state changes that were not yet completed on the Packet Forwarding Engine might be dropped at the Packet Forwarding Engine level due to deferred reconfiguration of hardware filters.

When failover-delay Is Configured

When failover-delay is configured, the sequence of events for VRRP sessions operated from the Routing Engine is modified as follows:

1. The Routing Engine detects that some VRRP groups require a state change.
2. The failover-delay starts for the period configured. The allowed failover-delay timer range is 50 through 100000 miliseconds.
3. The Routing Engine performs one-by-one state change for the VRRP groups. Every time there is a state change, and the new state for a particular VRRP group is primary, the Routing Engine generates appropriate VRRP announcement messages. However, communication toward the Packet Forwarding Engine is suppressed until the failover-delay timer expires.
4. After failover-delay timer expires, the Routing Engine sends message to the Packet Forwarding Engine about all VRRP groups that managed to change the state. As a consequence, hardware filters for those groups are reprogrammed, and for those groups whose new state is primary, gratuitous ARP messages are sent.

This process repeats until state transition for all VRRP groups is complete.

Thus, when failover-delay is configured even the Packet Forwarding Engine state for the first VRRP group is deferred. However, the network operator has the advantage of configuring a failover-delay value that best suits the need of the network deployment to ensure minimal outage during VRRP state change.

failover-delay influences only VRRP sessions operated by the VRRP process (vrrpd) running on the Routing Engine. For VRRP sessions distributed to the Packet Forwarding Engine, failover-delay configuration has no effect.

Change History Table

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

Release	Description
12.2	Junos OS Release 12.2 and later releases support VRRPv3.

RELATED DOCUMENTATION

Understanding High Availability Features on Juniper Networks Routers | 2

Configuring VRRP

Configuring VRRP

SUMMARY

Configure virtual router redundancy protocol (VRRP)_on your device with the steps and examples below.

IN THIS SECTION

Configuring Basic VRRP Support

NOTE: VRRP nonstop active routing (NSR) is enabled only when you configure the `nonstop-routing` statement at the `[edit routing-options]` or `[edit logical system` *`logical-system-name`* `routing-options]` hierarchy level.

The Virtual Router Redundancy Protocol (VRRP) groups multiple routing devices into a virtual router. At any time, one of the VRRP routing platforms is the primary (active) and the others are backups. If the primary fails, one of the backup routing platforms becomes the new primary router.

To configure basic VRRP support, configure VRRP groups on interfaces by including the `vrrp-group` statement:

```
vrrp-group group-id {
    priority number;
```

```
    virtual-address [ addresses ];
}
```

An interface can be a member of multiple VRRP groups. Within a VRRP group, the primary virtual router and the backup virtual router must be configured on different routing platforms.

You can include this statement at the following hierarchy level:

- `[edit interfaces interface-name unit logical-unit-number family inet address address]`

NOTE: We recommend only assigning one interface per VRRP group. All configured interfaces in a VRRP group will fail if the primary device goes down, which can cause issues with the failover process if multiple interfaces are configured.

Mandatory parameters to configure a VRRP group are as follows (examples will follow):

1. Configure the group identifier (mandatory).
2. Configure the group:
   - Configure the virtual IP address of one or more virtual routers that are members of the VRRP group (mandatory).
   - Configure the virtual link-local address (VRRP for IPv6 only). The virtual link-local address is autogenerated when you enable VRRPv3 on the interface. You may explicitly define a virtual link-local address for each VRRP for the IPv6 group. The virtual link-local address must be on the same subnet as the physical interface address.

When choosing a VRRP group identifier, consider the following:

- If `network-services` is configured in IP mode, don't configure the same VRRP group ID for multiple VRRP sessions on the same physical interface unless VRRP delegation is disabled. If multiple VRRP sessions are configured on the same physical interface with the same VRRP group ID while VRRP delegation is enabled, the other VRRP virtual IP addresses become unreachable when one of the logical interfaces is deleted.
- If `network-services` is configured in enhanced-ip mode, you can use the same VRRP group ID for multiple VRRP sessions.

When configuring a virtual IP address, consider the following:

- The virtual IP address must be the same for all routing platforms in the VRRP group.
- If you configure a virtual IP address to be the same as the physical interface's address, the interface becomes the primary virtual router for the group. In this case, you must configure the priority to be 255, and you must configure preemption by including the `preempt` statement.

- If the virtual IP address you choose is not the same as the physical interface's address, you must ensure that the virtual IP address does not appear anywhere else in the routing platform's configuration. Verify that you do not use this address for other interfaces, for the IP address of a tunnel, or for the IP address of static ARP entries.
- You cannot configure a virtual IP address to be the same as the interface's address for an aggregated Ethernet interface. This configuration is not supported.
- For VRRP for IPv6, the `EUI-64` option cannot be used. In addition, the Duplicate Address Detection (DAD) process will not run for virtual IPv6 addresses.
- You cannot configure the same virtual IP address on interfaces that belong to the same logical system and routing instance combination. However, you can configure the same virtual IP address on interfaces that belong to different logical systems and routing instance combinations.

In determining what priority will make a given routing platform in a VRRP group a primary or backup, consider the following:

- You can force assignment of primary and backup routers using priorities from 1 through 255, where 255 is the highest priority.
- The priority value for the VRRP router that owns the IP address(es) associated with the virtual router must be 255.
- VRRP routers backing up a virtual router must use priority values from 1 through 254.
- The default priority value for VRRP routers backing up a virtual router is 100.
- Are there tracked interfaces or routes with priority costs?

  The priority cost is the value associated with a tracked logical interface or route that is to be subtracted from the configured VRRP priority when the tracked logical interface or route goes down, forcing a new primary router election. The value of a priority cost can be from 1 through 254. The sum of the priority costs for all tracked logical interfaces or routes must be less than or equal to the configured priority of the VRRP group.

NOTE: Mixed tagging (configuring two logical interfaces on the same Ethernet port, one with single-tag framing and one with dual-tag framing) is supported only for interfaces on Gigabit Ethernet IQ2 and IQ PICs. If you include the `flexible-vlan-tagging` statement at the `[edit interfaces` *`interface-name`*`]` hierarchy level for a VRRP-enabled interface on a PIC that does not support mixed tagging, VRRP on that interface is disabled. In the output of the `show vrrp summary` operational command, the interface status is listed as `Down`.

NOTE: If you enable MAC source address filtering on an interface, you must include the virtual MAC address in the list of source MAC addresses that you specify in the `source-address-filter` statement at the `[edit interfaces` *`interface-name`*`]` hierarchy level. (For more information, see the Junos OS Network Interfaces Library for Routing Devices.) MAC addresses ranging from 00:00:5e:00:01:00 through 00:00:5e:00:01:ff are reserved for VRRP, as defined in RFC 2378. The VRRP group number must be the decimal equivalent of the last hexadecimal byte of the virtual MAC address.

Here are specific examples of configuring a VRRP group.

Configuring for VRRP IPv4 Groups

To configure basic VRRP (IPv4) groups on interfaces:

NOTE: You can also configure a VRRP IPv4 group at the `[edit logical-systems` *`logical-system-name`*`]` hierarchy level.

1. Configure the group identifier.

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
user@device# set vrrp-group group-id
```

   Assign a value from 0 through 255.

2. Configure the VRRP for IPv4 group:

   - Configure the virtual IP address of one or more virtual routers that are members of the VRRP group.

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
user@device# set vrrp-group group-id virtual-address [ addresses ]
```

     Normally, you configure only one virtual IP address per group. However, you can configure up to eight addresses. Do not include a prefix length in a virtual IP address.

   - Configure the priority for this routing platform to become the primary virtual router.

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
user@device# set vrrp-group group-id priority number
```

Configure the value used to elect the primary virtual router in the VRRP group. It can be a number from 1 through 255. The default value for backup routers is 100. A larger value indicates a higher priority. The routing platform with the highest priority within the group becomes the primary router. Primary router sends periodic VRRP advertisement messages to each virtual routers. The backup routers do not attempt to preempt the primary router unless it has higher priority. This eliminates service disruption unless a more preferred path becomes available. It is possible to administratively prohibit all preemption attempts, with the exception of a VRRP router becoming primary router of any virtual router associated with addresses it owns.

Configuring VRRP for IPv6 Groups

To configure basic VRRP for IPv6 groups on interfaces:

NOTE: You can also configure a VRRP IPv6 group at the `[edit logical-systems logical-system-name]` hierarchy level.

1. Configure the group identifier.

```
[edit interfaces interface-name unit logical-unit-number family inet6 address ipv6-address]
user@device# set vrrp-inet6-group group-id
```

   Assign a value from 0 through 255.

2. Configure the VRRP for IPv6 group:

   - Configure the virtual IP address of one or more virtual routers that are members of the VRRP group.

```
[edit interfaces interface-name unit logical-unit-number family inet6 address ipv6-address]
user@device# set vrrp-inet6-group group-id virtual-inet6-address [ ipv6-addresses ]
```

     Normally, you configure only one virtual IP address per group. However, you can configure up to eight addresses. Do not include a prefix length in a virtual IP address.

   - Configure the virtual link-local address.

```
[edit interfaces interface-name unit logical-unit-number family inet6 address ipv6-address]
user@device# set vrrp-inet6-group group-id virtual-link-local-address ipv6-address
```

You must explicitly define a virtual link-local address for each VRRP for IPv6 group. Otherwise, when you attempt to commit the configuration, the commit request fails. The virtual link-local address must be on the same subnet as the physical interface address.

- Configure the priority for this routing platform to become the primary virtual router.

```
[edit interfaces interface-name unit logical-unit-number family inet6 address ipv6-address]
user@device# set vrrp-inet6-group group-id priority number
```

Configure the value used to elect the primary virtual router in the VRRP group. It can be a number from 1 through 255. The default value for backup routers is 100. A larger value indicates a higher priority. The routing platform with the highest priority within the group becomes the primary router. If there are two or more backup routers with the same priority, the router that has the highest primary address becomes the primary.

Example: Configuring VRRP for IPv4

IN THIS SECTION

This example shows how to configure VRRP properties for IPv4.

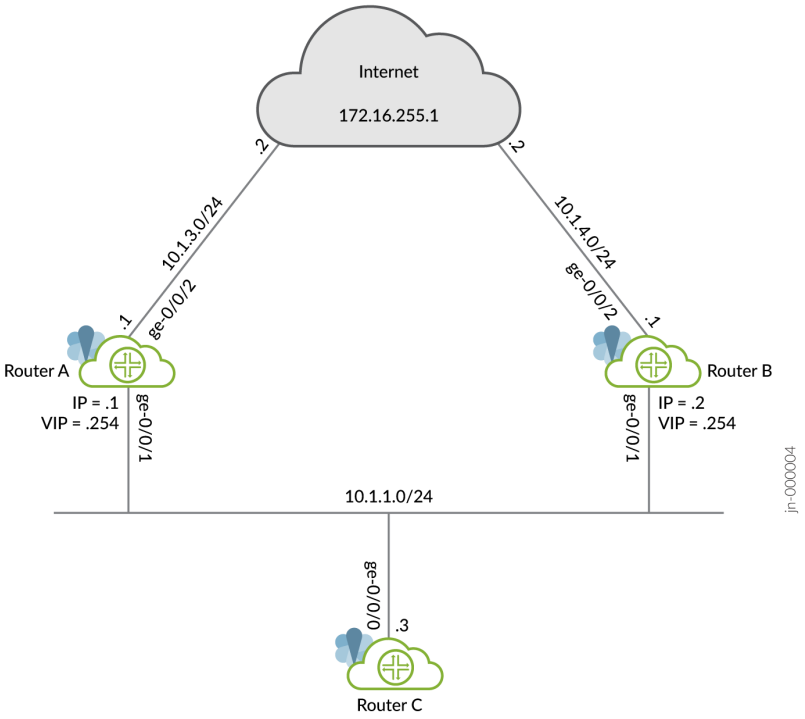
Requirements

This example uses the following hardware and software components:

- Three routers
- Junos OS Release 11.3 or later
  - This example has been recently updated and revalidated on Junos OS Release 21.1R1.
  - For details on VRRP support for specific platform and Junos OS release combinations, see Feature Explorer.

Overview

This example uses a VRRP group, which has a virtual address for IPv4. Devices on the LAN use this virtual address as their default gateway. If the primary router fails, the backup router takes over for it.



Configuring VRRP

IN THIS SECTION

Configuring Router A

CLI Quick Configuration

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

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24 vrrp-group 1 virtual-address
10.1.1.254
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24 vrrp-group 1 priority 110
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24 vrrp-group 1 accept-data
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24 vrrp-group 1 track interface
ge-0/0/2 priority-cost 20
set interfaces ge-0/0/2 unit 0 family inet address 10.1.3.1/24
set routing-options static route 0.0.0.0/0 next-hop 10.1.3.2
```

Step-by-Step Procedure

To configure this example:

1. Configure the interfaces.

```
[edit]
user@routerA# set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24
user@routerA# set interfaces ge-0/0/2 unit 0 family inet address 10.1.3.1/24
```

2. Configure the IPv4 VRRP group identifier and the virtual IP address.

```
[edit interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24]
user@routerA# set vrrp-group 1 virtual-address 10.1.1.254
```

3. Configure the priority for RouterA higher than RouterB to become the primary virtual router. RouterB is using the default priority of 100.

```
[edit interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24]
user@routerA# set vrrp-group 1 priority 110
```

4. Configure `track interface` to track whether the interface connected to the Internet is up, down, or not present to change the priority of the VRRP group.

```
[edit interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24]
user@routerA# set vrrp-group 1 track interface ge-0/0/2 priority-cost 20
```

5. Configure `accept-data` to enable the primary router to accept all packets destined for the virtual IP address.

```
[edit interfaces ge-0/0/1 unit 0 family inet address 10.1.1.1/24]
user@routerA# set vrrp-group 1 accept-data
```

6. Configure a static route for traffic to the Internet.

```
[edit]
user@routerA# set routing-options static route 0.0.0.0/0 next-hop 10.1.3.2
```

Results

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

```
[edit]
user@routerA# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.1/24 {
                vrrp-group 1 {
                    virtual-address 10.1.1.254;
                    priority 110;
                    accept-data;
                    track {
                        interface ge-0/0/2 {
                            priority-cost 20;
                        }
                    }
```

```
                }
            }
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.1.3.1/24;
        }
    }
}
```

```
[edit]
user@routerA# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.1.3.2;
}
```

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

Configuring Router B

CLI Quick Configuration

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

```
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24 vrrp-group 1 virtual-address
10.1.1.254
set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24 vrrp-group 1 accept-data
set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.1/24
set routing-options static route 0.0.0.0/0 next-hop 10.1.4.2
```

Step-by-Step Procedure

To configure this example:

1. Configure the interfaces.

```
[edit]
user@routerB# set interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24
user@routerB# set interfaces ge-0/0/2 unit 0 family inet address 10.1.4.1/24
```

2. Configure the IPv4 VRRP group identifier and the virtual IP address.

```
[edit interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24]
user@routerB# set vrrp-group 1 virtual-address 10.1.1.254
```

3. Configure `accept-data` to enable the backup router to accept all packets destined for the virtual IP address in the event the backup router becomes primary.

```
[edit interfaces ge-0/0/1 unit 0 family inet address 10.1.1.2/24]
user@routerB# set vrrp-group 1 accept-data
```

4. Configure a static route for traffic to the Internet.

```
[edit]
user@routerB# set routing-options static route 0.0.0.0/0 next-hop 10.1.4.2
```

Results

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

```
[edit]
user@routerB# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.1.2/24 {
                vrrp-group 1 {
                    virtual-address 10.1.1.254;
                    accept-data;
```

```
                }
            }
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.1.4.1/24;
        }
    }
}
```

```
[edit]
user@routerB# show routing-options
static {
    route 0.0.0.0/0 next-hop 10.1.4.2;
}
```

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

Configuring Router C

CLI Quick Configuration

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

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.3/24
set routing-options static route 0.0.0.0/0 next-hop 10.1.1.254
```

Verification

IN THIS SECTION

Verifying That VRRP Is Working on Router A

Purpose

Verify that VRRP is active on Router A and that its role in the VRRP group is correct.

Action

Use the following commands to verify that VRRP is active on Router A, that the router is primary for group 1 and the interface connected to the Internet is being tracked.

```
user@routerA> show vrrp
Interface     State       Group   VR state VR Mode   Timer    Type   Address
ge-0/0/1.0    up              1   master   Active      A  0.779 lcl    10.1.1.1
                                                              vip    10.1.1.254
```

```
user@routerA> show vrrp track
Track Int   State         Speed   VRRP Int   Group   VR State      Current prio
ge-0/0/2.0  up               1g   ge-0/0/1.0     1   master                 110
```

Meaning

The `show vrrp` command displays fundamental information about the VRRP configuration. This output shows that the VRRP group is active and that this router has assumed the primary role. The `lcl` address is the physical address of the interface and the `vip` address is the virtual address shared by both routers. The `Timer` value (`A 0.779`) indicates the remaining time (in seconds) in which this router expects to receive a VRRP advertisement from the other router.

Verifying That VRRP Is Working on Router B

Purpose

Verify that VRRP is active on Router B and that its role in the VRRP group is correct.

Action

Use the following command to verify that VRRP is active on Router B and that the router is backup for group 1.

```
user@routerB> show vrrp
Interface     State       Group   VR state VR Mode   Timer    Type   Address
ge-0/0/1.0    up              1   backup   Active      D  2.854 lcl    10.1.1.2
                                                               vip    10.1.1.254
                                                               mas    10.1.1.1
```

Meaning

The `show vrrp` command displays fundamental information about the VRRP configuration. This output shows that the VRRP group is active and that this router has assumed the backup role. The `lcl` address is the physical address of the interface and the `vip` address is the virtual address shared by both routers. The `Timer` value (`D 2.854`) indicates the remaining time (in seconds) in which this router expects to receive a VRRP advertisement from the other router.

Verifying Router C Reaches the Internet Transiting Router A

Purpose

Verify connectivity to the Internet from Router C.

Action

Use the following commands to verify that Router C can reach the Internet.

```
user@routerC> ping 172.16.255.1 count 2
PING 172.16.255.1 (172.16.255.1): 56 data bytes
64 bytes from 172.16.255.1: icmp_seq=0 ttl=63 time=9.394 ms
64 bytes from 172.16.255.1: icmp_seq=1 ttl=63 time=30.536 ms
```

```
--- 172.16.255.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 9.394/19.965/30.536/10.571 ms
```

```
user@routerC> traceroute 172.16.255.1
traceroute to 172.16.255.1 (172.16.255.1), 30 hops max, 52 byte packets
 1  10.1.1.1  3.781 ms  37.650 ms  3.877 ms
 2  172.16.255.1  31.581 ms  31.337 ms  27.170 ms
```

Meaning

The `ping` command shows reachability to the Internet and the `traceroute` command shows that Router A is being transited.

Verifying Router B Becomes Primary for VRRP

Purpose

Verify that Router B becomes primary for VRRP when the interface between Router A and the Internet goes down.

Action

Use the following commands to verify that Router B is primary and that Router C can reach the Internet transiting Router B.

```
user@routerA> show vrrp track detail
Tracked interface: ge-0/0/2.0
  State: down, Speed: 1g
  Incurred priority cost: 20
  Tracking VRRP interface: ge-0/0/1.0, Group: 1
    VR State: backup
    Current priority: 90, Configured priority: 110
    Priority hold-time: disabled
```

```
user@routerB> show vrrp
Interface     State       Group   VR state VR Mode   Timer    Type   Address
```

```
ge-0/0/1.0    up             1   master   Active      A  0.079 lcl    10.1.1.2
                                                              vip    10.1.1.254
```

```
user@routerC> traceroute 172.16.255.1
traceroute to 172.16.255.1 (172.16.255.1), 30 hops max, 52 byte packets
 1  10.1.1.2  6.532 ms  3.800 ms  2.958 ms
 2  172.16.255.1  44.359 ms  16.268 ms  22.823 ms
```

Meaning

The `show vrrp track detail` command shows the tracked interface is down on Router A, that the priority has dropped to 90, and that Router A is now the backup. The `show vrrp` command shows that Router B is now the primary for VRRP and the `traceroute` command shows that Router B is now being transited.

Configuring VRRP and VRRP for IPv6

To configure VRRP or VRRP for IPv6, include the `vrrp-group` or `vrrp-inet6-group` statement, respectively. These statements are available at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address*]

The VRRP and VRRP IPv6 configuration statements are as follows:

```
(vrrp-group | vrrp-inet6-group) group-number {
    (accept-data | no-accept-data);
    advertise-interval seconds;
    authentication-key key;
    authentication-type authentication;
    fast-interval milliseconds;
    (preempt | no-preempt) {
        hold-time seconds;
    }
    priority-number number;
    track {
        priority-hold-time;
        interface interface-name {
```

```
            priority-cost priority;
            bandwidth-threshold bits-per-second {
                priority-cost;
            }
        }
    }
    virtual-address [ addresses ];
}
```

You can configure VRRP IPv6 with a global unicast address.

To trace VRRP and VRRP for IPv6 operations, include the `traceoptions` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
traceoptions {
    file <filename> <files number <match regular-expression <microsecond-stamp> <size size>
<world-readable | no-world-readable>;
    flag flag;
    no-remote-trace;
}
```

When there are multiple VRRP groups, there is a few seconds delay between the time the first gratuitous ARP is sent out and the rest of the gratuitous ARP are sent. Configuring failover-delay compensates for this delay. To configure the failover delay from 500 to 2000 milliseconds for VRRP and VRRP for IPv6 operations, include the `failover-delay milliseconds` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
failover-delay milliseconds;
```

To configure the startup period for VRRP and VRRP for IPv6 operations, include the `startup-silent-period` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
startup-silent-period seconds;
```

To enable VRRPv3, set the `version-3` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
version-3;
```

Configuring VRRP for IPv6 (CLI Procedure)

By configuring the Virtual Router Redundancy Protocol (VRRP) on EX Series switches, you can enable hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts. You can configure VRRP for IPv6 on Gigabit Ethernet, 10-Gigabit Ethernet, and logical interfaces.

To configure VRRP for IPv6:

1. Configure VRRP group support on interfaces:

   ```
   [edit interfaces interface-name unit logical-unit-number family inet6 address address]

   user@switch# set vrrp-inet6-group group-id priority number virtual-inet6-address address
   virtual-link-local-address ipv6-address
   ```

   You must explicitly define a virtual link local address for each VRRP for IPv6 group. Otherwise, when you attempt to commit the configuration, the commit request fails. The virtual link local address must be on the same subnet as the physical interface address.
2. If you want to configure the priority order in which this switch functioning as a backup router becomes the primary router if the primary router becomes nonoperational, configure a priority for this switch:

   ```
   [edit interfaces interface-name unit logical-unit-number family inet6 address address vrrp-
   inet6-group group-id]

   user@switch# set priority number
   ```

3. Specify the interval in milliseconds in which the primary router sends advertisement packets to the members of the VRRP group:

```
[edit interfaces interface-name unit logical-unit-number family inet6 address address vrrp-inet6-group group-id]
user@switch# set inet6-advertise-interval milliseconds
```

4. By default, a higher-priority backup router preempts a lower-priority primary router.
   - To explicitly enable the primary router to be preempted:

```
[edit interfaces interface-name unit logical-unit-number family inet6 address address vrrp-inet6-group group-id]
user@switch# set preempt
```

   - To prohibit a higher-priority backup router from preempting a lower priority primary router:

```
[edit interfaces interface-name unit logical-unit-number family inet6 address address vrrp-inet6-group group-id]
user@switch# set no-preempt
```

Example: Configuring VRRP for IPv6

IN THIS SECTION

This example shows how to configure VRRP properties for IPv6.

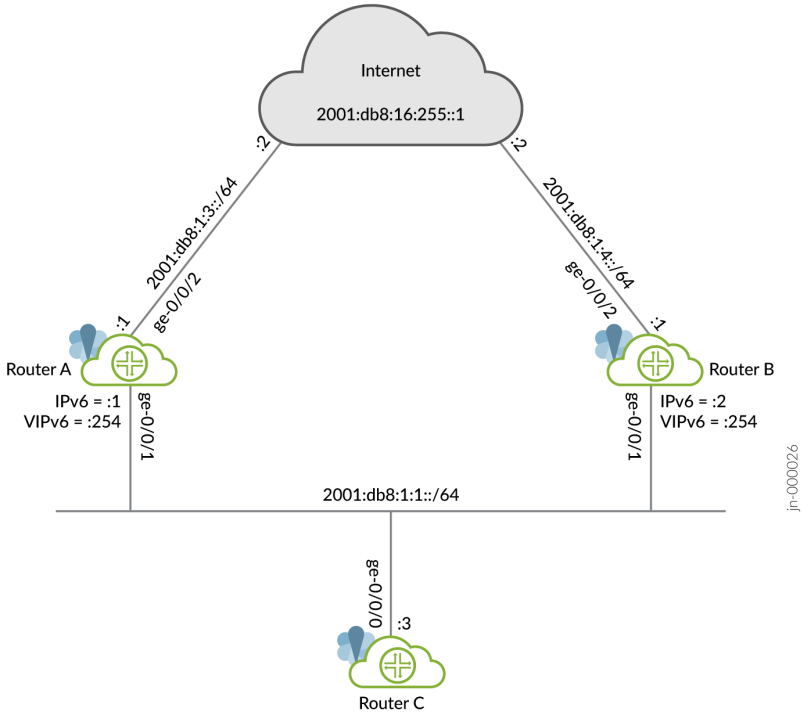
Requirements

This example uses the following hardware and software components:

- Three routers
- Junos OS Release 11.3 or later
  - This example has been recently updated and revalidated on Junos OS Release 21.1R1.
  - For details on VRRP support for specific platform and Junos OS release combinations, see Feature Explorer.

Overview

This example uses a VRRP group, which has a virtual address for IPv6. Devices on the LAN use this virtual address as their default gateway. If the primary router fails, the backup router takes over for it.



Configuring VRRP

IN THIS SECTION

Configuring Router A

CLI Quick Configuration

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

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64 vrrp-inet6-group 1
virtual-inet6-address 2001:db8:1:1::254
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64 vrrp-inet6-group 1
priority 110
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64 vrrp-inet6-group 1 accept-
data
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64 vrrp-inet6-group 1 track
interface ge-0/0/2 priority-cost 20
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1:3::1/64
set protocols router-advertisement interface ge-0/0/1.0 virtual-router-only
set protocols router-advertisement interface ge-0/0/1.0 prefix 2001:db8:1:1::/64
set routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:1:3::2
```

Step-by-Step Procedure

To configure this example:

1. Configure the interfaces.

```
[edit]
user@routerA# set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64
user@routerA# set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1:3::1/64
```

2. Configure the IPv6 VRRP group identifier and the virtual IP address.

```
[edit interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64]
user@routerA# set vrrp-inet6-group 1 virtual-inet6-address 2001:db8:1:1::254
```

3. Configure the priority for RouterA higher than RouterB to become the primary virtual router. RouterB is using the default priority of 100.

```
[edit interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64]
user@routerA# set vrrp-inet6-group 1 priority 110
```

4. Configure `track interface` to track whether the interface connected to the Internet is up, down, or not present to change the priority of the VRRP group.

```
[edit interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64]
user@routerA# set vrrp-inet6-group 1 track interface ge-0/0/2 priority-cost 20
```

5. Configure `accept-data` to enable the primary router to accept all packets destined for the virtual IP address.

```
[edit interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::1/64]
user@routerA# set vrrp-inet6-group 1 accept-data
```

6. Configure a static route for traffic to the Internet.

```
[edit]
user@routerA# set routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:1:3::2
```

7. For VRRP for iPv6, you must configure the interface on which VRRP is configured to send IPv6 router advertisements for the VRRP group. When an interface receives an IPv6 router solicitation message, it sends an IPv6 router advertisement to all VRRP groups configured on it.

```
[edit protocols router-advertisement interface ge-0/0/1.0]
user@routerA# set prefix 2001:db8:1:1::/64
```

8. Configure router advertisements to be sent only for VRRP IPv6 groups configured on the interface if the groups are in the primary state.

```
[edit protocols router-advertisement interface ge-0/0/1.0]
user@routerA# set virtual-router-only
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols router-advertisement` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@routerA# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:1:1::1/64 {
                vrrp-inet6-group 1 {
                    virtual-inet6-address 2001:db8:1:1::254;
                    priority 110;
                    accept-data;
                    track {
                        interface ge-0/0/2 {
                            priority-cost 20;
                        }
                    }
                }
            }
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet6 {
            address 2001:db8:1:3::1/64;
        }
    }
}
```

```
[edit]
user@routerA# show protocols router-advertisement
interface ge-0/0/1.0 {
    virtual-router-only;
```

```
    prefix 2001:db8:1:1::/64;
}
```

```
[edit]
user@routerA# show routing-options
rib inet6.0 {
    static {
        route 0::0/0 next-hop 2001:db8:1:3::2;
    }
}
```

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

Configuring Router B

CLI Quick Configuration

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

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::2/64 vrrp-inet6-group 1
virtual-inet6-address 2001:db8:1:1::254
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::2/64 vrrp-inet6-group 1
priority 110
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::2/64 vrrp-inet6-group 1 accept-
data
set protocols router-advertisement interface ge-0/0/1.0 virtual-router-only
set protocols router-advertisement interface ge-0/0/1.0 prefix 2001:db8:1:1::/64
```

Step-by-Step Procedure

To configure this example:

1. Configure the interfaces.

```
[edit]
user@routerB# set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::2/64
user@routerB# set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:1:4::1/64
```

2. Configure the IPv6 VRRP group identifier and the virtual IP address.

```
[edit interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::2/64]
user@routerB# set vrrp-inet6-group 1 virtual-inet6-address 2001:db8:1:1::254
```

3. Configure `accept-data` to enable the backup router to accept all packets destined for the virtual IP address in the event the backup router becomes primary.

```
[edit interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:1:1::2/64]
user@routerB# set vrrp-inet6-group 1 accept-data
```

4. Configure a static route for traffic to the Internet.

```
[edit]
user@routerB# set routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:1:4::2
```

5. Configure the interface on which VRRP is configured to send IPv6 router advertisements for the VRRP group. When an interface receives an IPv6 router solicitation message, it sends an IPv6 router advertisement to all VRRP groups configured on it.

```
[edit protocols router-advertisement interface ge-0/0/1.0]
user@routerB# set prefix 2001:db8:1:1::/64
```

6. Configure router advertisements to be sent only for VRRP IPv6 groups configured on the interface if the groups are in the primary state.

```
[edit protocols router-advertisement interface ge-0/0/1.0]
user@routerB# set virtual-router-only
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show protocols router-advertisement` and `show routing-options` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@routerB# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:db8:1:1::2/64 {
                vrrp-inet6-group 1 {
                    virtual-inet6-address 2001:db8:1:1::254;
                    accept-data;
                }
            }
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet6 {
            address 2001:db8:1:4::1/64;
        }
    }
}
```

```
[edit]
user@routerB# show protocols router-advertisement
interface ge-0/0/1.0 {
    virtual-router-only;
    prefix 2001:db8:1:1::/64;
}
```

```
[edit]
user@routerB# show routing-options
rib inet6.0 {
    static {
```

```
        route 0::0/0 next-hop 2001:db8:1:4::2;
    }
}
```

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

Configuring Router C

CLI Quick Configuration

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

```
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:1:1::3/64
set routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:1:1::254
```

Verification

IN THIS SECTION

Verifying That VRRP Is Working on Router A

Purpose

Verify that VRRP is active on Router A and that its role in the VRRP group is correct.

Action

Use the following commands to verify that VRRP is active on Router A, that the router is primary for group 1 and the interface connected to the Internet is being tracked.

```
user@routerA> show vrrp
Interface     State       Group   VR state VR Mode   Timer    Type   Address
ge-0/0/1.0    up              1   master   Active      A  0.690 lcl    2001:db8:1:1::1
                                                               vip    fe80::200:5eff:fe00:201
                                                               vip    2001:db8:1:1::254
```

```
user@routerA> show vrrp track
Track Int   State         Speed   VRRP Int   Group   VR State      Current prio
ge-0/0/2.0  up               1g   ge-0/0/1.0     1   master                 110
```

Meaning

The `show vrrp` command displays fundamental information about the VRRP configuration. This output shows that the VRRP group is active and that this router has assumed the primary role. The `lcl` address is the physical address of the interface and the `vip` address is the virtual address shared by both routers. The `Timer` value (`A 0.690`) indicates the remaining time (in seconds) in which this router expects to receive a VRRP advertisement from the other router.

Verifying That VRRP Is Working on Router B

Purpose

Verify that VRRP is active on Router B and that its role in the VRRP group is correct.

Action

Use the following command to verify that VRRP is active on Router B and that the router is backup for group 1.

```
user@routerB> show vrrp
Interface     State       Group   VR state VR Mode   Timer    Type   Address
ge-0/0/1.0    up              1   backup   Active      D  2.947 lcl    2001:db8:1:1::2
                                                               vip    fe80::200:5eff:fe00:201
```

```
                                                    vip    2001:db8:1:1::254
                                                    mas    fe80::5668:a0ff:fe99:2d7d
```

Meaning

The `show vrrp` command displays fundamental information about the VRRP configuration. This output shows that the VRRP group is active and that this router has assumed the backup role. The `lcl` address is the physical address of the interface and the `vip` address is the virtual address shared by both routers. The `Timer` value (`D 2.947`) indicates the remaining time (in seconds) in which this router expects to receive a VRRP advertisement from the other router.

Verifying Router C Reaches the Internet Transiting Router A

Purpose

Verify connectivity to the Internet from Router C.

Action

Use the following commands to verify that Router C can reach the Internet.

```
user@routerC> ping 2001:db8:16:255::1 count 2
PING6(56=40+8+8 bytes) 2001:db8:1:1::3 --> 2001:db8:16:255::1
16 bytes from 2001:db8:16:255::1, icmp_seq=0 hlim=63 time=12.810 ms
16 bytes from 2001:db8:16:255::1, icmp_seq=1 hlim=63 time=30.139 ms

--- 2001:db8:16:255::1 ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 12.810/21.474/30.139/8.664 ms
```

```
user@routerC> traceroute 2001:db8:16:255::1
traceroute6 to 2001:db8:16:255::1 (2001:db8:16:255::1) from 2001:db8:1:1::3, 64 hops max, 12
byte packets
 1  2001:db8:1:1::1 (2001:db8:1:1::1)  9.891 ms  32.353 ms  7.859 ms
 2  2001:db8:16:255::1 (2001:db8:16:255::1)  257.483 ms  19.877 ms  7.451 ms
```

Meaning

The `ping` command shows reachabilty to the Internet and the `traceroute` command shows that Router A is being transited.

Verifying Router B Becomes Primary for VRRP

Purpose

Verify that Router B becomes primary for VRRP when the interface between Router A and the Internet goes down.

Action

Use the following commands to verify that Router B is primary and that Router C can reach the Internet transiting Router B.

```
user@routerA> show vrrp track detail
Tracked interface: ge-0/0/2.0
  State: down, Speed: 1g
  Incurred priority cost: 20
  Tracking VRRP interface: ge-0/0/1.0, Group: 1
    VR State: backup
    Current priority: 90, Configured priority: 110
    Priority hold-time: disabled
```

```
user@routerB> show vrrp
Interface     State       Group   VR state VR Mode   Timer    Type   Address
ge-0/0/1.0    up              1   master   Active      A  0.119 lcl    2001:db8:1:1::2
                                                               vip    fe80::200:5eff:fe00:201
                                                               vip    2001:db8:1:1::254
```

```
user@routerC> traceroute 2001:db8:16:255::1
traceroute6 to 2001:db8:16:255::1 (2001:db8:16:255::1) from 2001:db8:1:1::3, 64 hops max, 12
byte packets
 1  2001:db8:1:1::2 (2001:db8:1:1::2)  52.945 ms  344.383 ms  29.540 ms
 2  2001:db8:16:255::1 (2001:db8:16:255::1)  46.168 ms  24.744 ms  23.867 ms
```

Meaning

The `show vrrp track detail` command shows the tracked interface is down on Router A, that the priority has dropped to 90, and that Router A is now the backup. The `show vrrp` command shows that Router B is now the primary for VRRP and the `traceroute` command shows that Router B is now being transited.

Configuring VRRP Authentication (IPv4 Only)

VRRP (IPv4 only) protocol exchanges can be authenticated to guarantee that only trusted routing platforms participate in routing in an autonomous system (AS). By default, VRRP authentication is disabled. You can configure one of the following authentication methods. Each VRRP group must use the same method.

- Simple authentication—Uses a text password included in the transmitted packet. The receiving routing platform uses an authentication key (password) to verify the packet.
- Message Digest 5 (MD5) algorithm—Creates the authentication data field in the IP authentication header. This header is used to encapsulate the VRRP PDU. The receiving routing platform uses an authentication key (password) to verify the authenticity of the IP authentication header and VRRP PDU.

To enable authentication and specify an authentication method, include the `authentication-type` statement:

```
authentication-type authentication;
```

authentication can be **simple** or **md5**. The authentication type must be the same for all routing platforms in the VRRP group.

You can include this statement at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number family inet address address vrrp-group group-id]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family inet address address vrrp-group group-id]`

If you include the `authentication-type` statement, you can configure a key (password) on each interface by including the `authentication-key` statement:

```
authentication-key key;
```

key (the password) is an ASCII string. For simple authentication, it can be from 1 through 8 characters long. For MD5 authentication, it can be from 1 through 16 characters long. If you include spaces, enclose all characters in quotation marks (“ ”). The key must be the same for all routing platforms in the VRRP group.

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

NOTE: When VRRPv3 is enabled, the authentication-type and authentication-key statements cannot be configured for any VRRP groups. Therefore, if authentication is required, you need to configure alternative non-VRRP authentication mechanisms.

Configuring VRRP Preemption and Hold Time

IN THIS SECTION

Configuring VRRP Preemption

By default, a higher-priority VRRP backup switch preempts a lower-priority primary switch. To explicitly enable this behavior, include the following statement:

```
preempt;
```

To prohibit a higher-priority VRRP backup switch from preempting a lower-priority primary switch, include the following statement on the lower-priority switch:

```
no-preempt;
```

You can include these statements at the following hierarchy level:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

Configuring the Preemption Hold Time

You can also configure a preemption hold time, which is the number of seconds a higher-priority backup router that has just started up waits before preempting the primary router. You might want to configure a hold time so that routing protocols or other Junos OS components converge before preemption occurs.

The hold time is applied only on startup. By default, the hold-time value is 0 seconds, meaning that preemption can occur immediately after the backup router starts up.

To modify the preemption hold-time value, configure the following statement:

```
hold-time seconds;
```

The hold time can be from 0 through 3600 seconds.

You can include this statement at the following hierarchy level:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet *address* vrrp-group *group-id*] preempt

Configuring the Advertisement Interval for the VRRP Primary Router

IN THIS SECTION

By default, the primary router sends VRRP advertisement packets every second to all members of the VRRP group. These packets indicate that the primary router is still operational. If the primary router fails or becomes unreachable, the backup router with the highest priority value becomes the new primary router.

You can modify the advertisement interval in seconds or in milliseconds. The interval must be the same for all routing platforms in the VRRP group.

For VRRP for IPv6, you must configure IPv6 router advertisements for the interface on which VRRP is configured to send IPv6 router advertisements for the VRRP group. To do so, include the `interface`

interface-name statement at the [edit protocols router-advertisement] hierarchy level. (For information about this statement and guidelines, see the Junos OS Routing Protocols Library for Routing Devices.) When an interface receives an IPv6 router solicitation message, it sends an IPv6 router advertisement to all VRRP groups configured on it. In the case of logical systems, IPv6 router advertisements are not sent to VRRP groups.

NOTE: The primary VRRP for an IPv6 router must respond to a router solicitation message with the virtual IP address of the router. However, when the interface *interface-name* statement is included at the [edit protocols router-advertisement] hierarchy level, the backup VRRP for an IPv6 router might send a response before the VRRP primary responds, so that the default route of the client is not set to the primary VRRP router's virtual IP address. To avoid this situation, include the virtual-router-only statement at the [edit protocols router-advertisement interface *interface-name*] hierarchy level. When this statement is included, router advertisements are sent only for VRRP IPv6 groups configured on the interface (if the groups are in the primary state). You must include this statement on both the primary and backup VRRP for IPv6 routers.

NOTE: In an EVPN network, including the virtual-router-only statement at the [edit protocols router-advertisement interface *interface-name*] hierarchy level restricts the router advertisements to be sent only for the link local virtual-gateway-address.

Modifying the Advertisement Interval in Seconds

To modify the time, in seconds, between the sending of VRRP advertisement packets, include the advertise-interval statement:

```
advertise-interval seconds;
```

The interval can be from 1 through 255 seconds.

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

NOTE: When VRRPv3 is enabled, the advertise-interval statement cannot be used to configure advertisement intervals. Instead, use the fast-interval statement to configure advertisement intervals.

Modifying the Advertisement Interval in Milliseconds

To modify the time, in milliseconds, between the sending of VRRP advertisement packets, include the `fast-interval` statement:

```
fast-interval milliseconds;
```

The interval can be from 10 through 40,950 milliseconds.

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]

NOTE: In the VRRP PDU, Junos OS sets the advertisement interval to 0. When you configure VRRP with other vendors' routers, the `fast-interval` statement works correctly only when the other routers also have an advertisement interval set to 0 in the VRRP PDUs. Otherwise, Junos OS interprets other routers' settings as advertisement timer errors.

To modify the time, in milliseconds, between the sending of VRRP for IPv6 advertisement packets, include the `inet6-advertise-interval` statement:

```
inet6-advertise-interval ms;
```

The range of values is from 100 through 40,000 milliseconds (ms).

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

NOTE: When VRRPv3 is enabled, the `inet6-advertise-interval` statement cannot be used to configure advertisement intervals. Instead, use the `fast-interval` statement to configure advertisement intervals.

Configuring the Startup Period for VRRP Operations

To configure the startup period for VRRP operations, include the `startup-silent-period` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
startup-silent-period seconds;
```

NOTE: During the silent startup period, the `show vrrp detail` command output shows a value of 0 for `Master priority`, and your own IP address for `Master router`. These values indicate that the Primary selection is not completed yet, and these values can be ignored.

Configuring a Backup Router to Preempt the VRRP Primary Router

By default, a higher-priority backup router preempts a lower-priority primary router. To explicitly enable the primary router to be preempted, include the `preempt` statement:

```
preempt;
```

You can include this statement at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number family (inet | inet6) address address (vrrp-group | vrrp-inet6-group) group-id]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family (Inet | inet6) address address (vrrp-group | vrrp-inet6-group) group-id]`

To prohibit a higher-priority backup router from preempting a lower-priority primary router, include the `no-preempt` statement:

```
no-preempt;
```

Configuring a Backup to Accept Packets Destined for the Virtual IP Address

By default, a switch configured to be a VRRP backup but acting as the primary does not process packets sent to the virtual IP address—that is, packets in which the destination address is the virtual IP address. To configure a backup switch to process packets sent to the virtual IP address while it is acting as the primary, include the `accept-data` statement on the backup:

```
accept-data;
```

You can include this statement at the following hierarchy level:

- `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet address` *`address`* `vrrp-group]` *`group-id`*

To explicitly prohibit the backup from accepting packets destined for the virtual IP address while acting as primary, include the `no-accept-data` statement:

```
no-accept-data;
```

If you include the `accept-data` statement, configure the connected hosts so that they:

- Process gratuitous ARP requests.
- Do not use packets other than ARP replies to update their ARP cache.

This statement is disabled by default. If you enable it, your configuration does not comply with RFC 3768.

To restrict incoming IP packets to ICMP only, you must configure firewall filters to accept only ICMP packets.

Modifying the Preemption Hold-Time Value for the VRRP Primary Router

The hold time is the maximum number of seconds that can elapse before a higher-priority backup router preempts the primary router. You might want to configure a hold time so that all Junos OS components converge before preemption.

By default, the hold-time value is 0 seconds. A value of 0 means that preemption can occur immediately after the backup router comes online. Note that the hold time is counted from the time the backup router comes online. The hold time is only valid when the VRRP router is just coming online.

To modify the preemption hold-time value, include the `hold-time` statement:

```
hold-time seconds;
```

The hold time can be from 0 through 3600 seconds.

You can include this statement at the following hierarchy levels:

- `[edit interfaces interface-name unit logical-unit-number family (inet | inet6) address address (vrrp-group | vrrp-inet6-group) group-id preempt]`
- `[edit logical-systems logical-system-name interfaces interface-name unit logical-unit-number family (Inet | inet6) address address (vrrp-group | vrrp-inet6-group) group-id preempt]`

Configuring the Asymmetric Hold Time for VRRP Routers

In Junos OS Release 9.5 and later, the `asymmetric-hold-time` statement at the `[edit protocols vrrp]` hierarchy level enables you to configure a VRRP primary router to switch over to the backup router immediately—that is, without waiting for the priority hold time to expire—when a tracked interface or route goes down or when the bandwidth of a tracked interface decreases. Such events can cause an immediate reduction in the priority based on the configured priority cost for the event, and trigger a primary-role election.

However, when the tracked route or interface comes up again, or when the bandwidth for a tracked interface increases, the backup (original primary) router waits for the hold time to expire before it updates the priority and initiates the switchover if the priority is higher than the priority for the VRRP primary (original backup) router.

If the `asymmetric-hold-time` statement is not configured, the VRRP primary waits for the hold time to expire before it initiates a switchover when a tracked route goes down or when the bandwidth of a tracked interface decreases.

Example: Configuring Asymmetric Hold Time

```
[edit]
user@host# set protocols vrrp asymmetric-hold-time
[edit]
user@host# show protocols vrrp
asymmetric-hold-time;
```

Configuring Passive ARP Learning for Backup VRRP Routers

By default, the backup VRRP router drops ARP requests for the VRRP-IP to VRRP-MAC address translation. This means that the backup router does not learn the ARP (IP-to-MAC address) mappings for the hosts sending the requests. When it detects a failure of the primary router and transitions to become the new primary router, the backup router must re-learn all the entries that were present in the ARP cache of the primary router. In environments with many directly attached hosts, such as metro Ethernet environments, the number of ARP entries to learn can be high. This can cause a significant transition delay, during which the traffic transmitted to some of the hosts might be dropped.

Passive ARP learning enables the ARP cache in the backup router to hold approximately the same contents as the ARP cache in the primary router, thus preventing the problem of learning ARP entries in a burst. To enable passive ARP learning, include the `passive-learning` statement at the `[edit system arp]` hierarchy level:

```
[edit system arp]
passive-learning;
```

We recommend setting passive learning on both the backup and primary VRRP routers. Doing so prevents the need to manually intervene when the primary router becomes the backup router. While a router is operating as the primary router, the passive learning configuration has no operational impact. The configuration takes effect only when the router is operating as a backup router.

For information about configuring gratuitous ARP and the ARP aging timer, see the Junos OS Administration Library for Routing Devices.

Configuring VRRP Route Tracking

Configure Routers R1 and R2 to run VRRP. Configure static routes and a policy for exporting the static routes on Router R3. The VRRP routing instances on R2 track the routes that are advertised by R3.

On Router R1

```
[edit interfaces]
ge-1/0/3 {
    unit 0 {
        vlan-id 1;
        family inet {
            address 200.100.50.2/24 {
                vrrp-group 0 {
```

```
                    virtual-address 200.100.50.101;
                    priority 195;
                }
            }
        }
    }
}
```

On Router R2

```
[edit interfaces]
ge-1/0/1 {
    unit 0 {
        vlan-id 1;
        family inet {
            address 200.100.50.1/24 {
                vrrp-group 0 {
                    virtual-address 200.100.50.101;
                    priority 200;
                    track {
                        route 59.0.58.153/32 routing-instance default priority-cost 5;
                        route 59.0.58.154/32 routing-instance default priority-cost 5;
                        route 59.0.58.155/32 routing-instance default priority-cost 5;
                    }
                }
            }
        }
    }
}
```

On Router R3

```
[edit]
policy-options {
    policy-statement static-policy {
        term term1 {
            then accept;
        }
    }
}
protocols {
```

```
    ospf {
        export static-policy;
        reference-bandwidth 4g;
        area 0.0.0.0 {
            interface all;
            interface fxp0.0 {
                disable;
            }
        }
    }
}
routing-options {
    static {
        route 59.0.0.153/32 next-hop 45.45.45.46;
        route 59.0.0.154/32 next-hop 45.45.45.46;
        route 59.0.0.155/32 next-hop 45.45.45.46;
    }
}
```

Configuring a Logical Interface to Be Tracked for a VRRP Group

VRRP can track whether a logical interface is up, down, or not present, and can also dynamically change the priority of the VRRP group based on the state of the tracked logical interface, triggering a new primary router election. VRRP can also track the operational speed of a logical interface and dynamically update the priority of the VRRP group when the speed crosses a configured threshold.

When interface tracking is enabled, you cannot configure a priority of 255 (a priority of 255 designates the primary router). For each VRRP group, you can track up to 10 logical interfaces.

To configure a logical interface to be tracked, include the following statements:

```
track {
    interface interface-name {
        bandwidth-threshold bits-per-second priority-cost priority;
        priority-cost priority;
    }
```

```
    priority-hold-time seconds;
}
```

```
interface et-0/0/0 {
    priority-cost 30;
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]
- [edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

The interface specified is the interface to be tracked for the VRRP group. The priority hold time is the minimum length of time that must elapse between dynamic priority changes. A tracking event, such as an interface state change (up or down) or a change in bandwidth, triggers one of the following responses:

- The first tracking event initiates the priority hold timer, and also initializes the pending priority based on the current priority and the priority cost. However, the current priority remains unchanged.
- A tracking event or a manual configuration change that occurs while the priority hold timer is on triggers a pending priority update. However, the current priority remains unchanged.

This ensures that Junos OS does not initiate primary role elections every time a tracked interface flaps.

When the priority hold time expires, the current priority inherits the value from the pending priority, and the pending priority ceases.

NOTE: If you have configured asymmetric-hold-time, VRRP does not wait for the priority hold time to expire before initiating primary role elections if a tracked interface fails (state changes from up to down), or if the available bandwidth for a tracked interface decreases.

There are two priority-cost statements that show at this hierarchy level. The bandwidth-threshold statement specifies a threshold for the tracked interface. When the bandwidth of the tracked interface drops below the configured bandwidth threshold value, the VRRP group uses the bandwidth threshold priority

cost. You can track up to five bandwidth threshold statements for each tracked interface. Just under the `interface` statement there is a `priority-cost` statement that gives the value to subtract from priority when the interface is down.

The sum of the priority costs for all tracked logical interfaces must be less than or equal to the configured priority of the VRRP group. If you are tracking more than one interface, the router applies the sum of the priority costs for the tracked interfaces (at most, only one priority cost for each tracked interface) to the VRRP group priority.

Prior to Junos OS Release 15.1, an adjusted priority could not be zero. If the difference between the priority costs and the configured priority of the VRRP group was zero, the adjusted priority would become 1.

NOTE: In Junos OS Release 15.1 and later, an adjusted priority can be zero.

The priority value zero (0) indicates that the current primary router has stopped participating in VRRP. Such a priority value is used to trigger one of the backup routers to quickly transition to the primary router without having to wait for the current primary to time out.

If you are tracking more than one interface, the router applies the sum of the priority costs for the tracked interfaces (at most, only one priority cost for each tracked interface) to the VRRP group priority. However, the interface priority cost and bandwidth threshold priority cost values for each VRRP group are not cumulative. The router uses only one priority cost to a tracked interface as indicated in Table 17 on page 447.

Table 17: Interface State and Priority Cost Usage

Tracked Interface State	Priority Cost Usage
Down	`priority-cost` *priority*
Not down; media speed below one or more bandwidth thresholds	Priority cost of the lowest applicable bandwidth threshold

You must configure an interface priority cost only if you have configured no bandwidth thresholds. If you have not configured an interface priority cost value, and the interface is down, the interface uses the bandwidth threshold priority cost value of the lowest bandwidth threshold.

Configuring a Route to Be Tracked for a VRRP Group

VRRP can track whether a route is reachable (that is, the route exists in the routing table of the routing instance included in the configuration) and dynamically change the priority of the VRRP group based on the reachability of the tracked route, triggering a new primary router election.

To configure a route to be tracked, include the following statements:

```
track {
    priority-hold-time seconds;
    route prefix/prefix-length routing-instance instance-name priority-cost priority;
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]
- [edit interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet6 address *address* vrrp-inet6-group *group-id*]

The route prefix specified is the route to be tracked for the VRRP group. The priority hold time is the minimum length of time that must elapse between dynamic priority changes. A route tracking event, such as adding a route to or removing a route from the routing table, might trigger one or more of the following:

- The first tracking event initiates the priority hold timer, and also initializes the pending priority based on the current priority and the priority cost. However, the current priority remains unchanged.
- A tracking event or a manual configuration change that occurs while the priority hold timer is on triggers a pending priority update. However, the current priority remains unchanged.

When the priority hold time expires, the current priority inherits the value from the pending priority, and the pending priority ceases.

This ensures that Junos OS does not initiate primary role elections every time a tracked route flaps.

NOTE: If you have configured `asymmetric-hold-time`, VRRP does not wait for the priority hold time to expire before initiating primary role elections if a tracked route is removed from the routing table.

The routing instance is the routing instance in which the route is to be tracked. If the route is in the default, or global, routing instance, specify the instance name as `default`.

NOTE: Tracking a route that belongs to a routing instance from a different logical system is not supported.

The priority cost is the value to be subtracted from the configured VRRP priority when the tracked route goes down, forcing a new primary router election. The value can be from 1 through 254.

The sum of the priority costs for all tracked routes must be less than or equal to the configured priority of the VRRP group. If you are tracking more than one route, the router applies the sum of the priority costs for the tracked routes (at most, only one priority cost for each tracked route) to the VRRP group priority.

Prior to Junos OS Release 15.1, an adjusted priority could not be zero. If the difference between the priority costs and the configured priority of the VRRP group was zero, the adjusted priority would become 1.

NOTE: In Junos OS Release 15.1 and later, an adjusted priority can be zero.

The priority value zero (0) indicates that the current primary router has stopped participating in VRRP. Such a priority value is used to trigger one of the backup routers to quickly transition to the primary router without having to wait for the current primary to time out.

Example: Configuring Multiple VRRP Owner Groups

IN THIS SECTION

These examples show how to configure multiple virtual router redundancy protocol (VRRP) IPv4 and IPv6 owner groups.

Requirements

This example uses the following hardware and software components:

- A EX-Series, M-Series, MX-Series, or T-Series router.
- Junos OS release 12.3 or later

Overview

Multiple VRRP owner groups allows users to reuse interface address identifiers (IFAs) as virtual IP addresses (VIPs). You can configure multiple IPv4 owner groups, multiple IPv6 owner groups, or a mix of IPv4 and IPv6 owner groups.

Configuration

IN THIS SECTION

CLI Quick Configuration

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

Multiple IPv4 owner groups

```
edit interfaces ge-1/0/0 unit 0 family inet
set address 10.0.0.2/24 vrrp-group 2 virtual-address 10.0.0.4 accept-data
set address 20.0.0.2/24 vrrp-group 3 virtual-address 20.0.0.2 priority 255
set address 30.0.0.2/24 vrrp-group 4 virtual-address 30.0.0.2 priority 255
```

Multiple IPv6 owner groups

```
edit interfaces ge-1/0/0 unit 0 family inet6
set address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-inet6-address 2001:4818:f000:20::1
set address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-link-local-address
fe80:4818:f000:20::1
set address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 priority 255
set address fe80:4818:f000:13::2/64
set address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-inet6-address
2001:1000:f000:20::1
set address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-link-local-address
fe80:1000:f000:20::1
set address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 priority 255
set address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-inet6-address 2001:2000:f000:20::2
set address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-link-local-address
fe80:2000:f000:20::2
set address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 priority 250
```

Multiple IPv4 and IPv6 owner groups

```
edit interfaces ge-1/0/0 unit 0
set family inet address 10.0.0.1/24 vrrp-group 5 virtual-address 10.0.0.1
set family inet address 10.0.0.1/24 vrrp-group 5 priority 255
set family inet6 address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-inet6-address
2001:4818:f000:20::1
set family inet6 address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-link-local-address
fe80:4818:f000:20::1
set family inet6 address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 priority 255
set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-inet6-address
2001:1000:f000:20::1
set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-link-local-address
fe80:1000:f000:20::1
set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 priority 255
set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-inet6-address
```

```
2001:2000:f000:20::2
set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-link-local-address
fe80:2000:f000:20::2
set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 priority 250
```

Configuring multiple IPv4 owner groups

Step-by-Step Procedure

To configure multiple IPv4 owner groups:

1. Create an IPv4 interface on the device

```
[edit]
user@host# edit interfaces ge-1/0/0 unit 0 family inet
```

2. Configure the first IPv4 owner group

```
[edit interfaces ge-1/0/0 unit 0 family inet]
user@host# set address 10.0.0.2/24 vrrp-group 2 virtual-address 10.0.0.4 accept-data
```

3. Configure the second IPv4 owner group

```
[edit interfaces ge-1/0/0 unit 0 family inet]
user@host# set address 20.0.0.2/24 vrrp-group 3 virtual-address 20.0.0.2 priority 255
```

4. Configure the third IPv4 owner group

```
[edit interfaces ge-1/0/0 unit 0 family inet]
user@host# set address 30.0.0.2/24 vrrp-group 4 virtual-address 30.0.0.2 priority 255
```

Configuring multiple IPv6 owner groups

Step-by-Step Procedure

To configure multiple IPv6 owner groups:

1. Create an IPv6 interface on the device

```
[edit]
user@host# edit interfaces ge-1/0/0 unit 0 family inet6
```

2. Configure the inet6 address for the first IPv6 owner group

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-inet6-address
2001:4818:f000:20::1
```

3.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-link-local-
address fe80:4818:f000:20::1
```

4.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 priority 255
```

5.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-
inet6-address 2001:1000:f000:20::1
```

6.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-link-
local-address fe80:1000:f000:20::1
```

7.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 priority 255
```

8.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-
inet6-address 2001:2000:f000:20::2
```

9.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-link-
local-address fe80:2000:f000:20::2
```

10.

```
[edit interfaces ge-1/0/0 unit 0 family inet6]
user@host# set address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 priority 250
```

Configuring multiple IPv4 and IPv6 owner groups

Step-by-Step Procedure

To configure multiple IPv4 and IPv6 owner groups:

1. Create an interface on the device

```
[edit]
user@host# edit interfaces ge-1/0/0 unit 0
```

2. Configure the family inet address and virtual address for the IPv4 owner group

```
[edit interfaces ge-1/0/0 unit 0]
user@host# set family inet address 10.0.0.1/24 vrrp-group 5 virtual-address 10.0.0.1
```

3. Set the priority of the IPv4 owner group to 255

```
[edit interfaces ge-1/0/0 unit 0]
set family inet address 10.0.0.1/24 vrrp-group 5 priority 255
```

4. Configure the inet6 address for the first IPv6 owner group

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-inet6-address
2001:4818:f000:20::1
```

5. Set the virtual link local address for the first IPv6 owner group

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 virtual-link-local-
address fe80:4818:f000:20::1
```

6. Set the first IPv6 owner group's priority to 255

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:4818:f000:20::1/64 vrrp-inet6-group 1 priority 255
```

7. Configure the inet6 address for the second IPv6 owner group

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-inet6-address
2001:1000:f000:20::1
```

8. Set the virtual link local address for the second IPv6 owner group

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 virtual-link-local-
address fe80:1000:f000:20::1
```

9. Set the second IPv6 owner group's priority to 255

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:1000:f000:20::1/64 vrrp-inet6-group 2 priority 255
```

10. Configure the inet6 address for the third IPv6 owner group

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-inet6-address
2001:2000:f000:20::2
```

11. Set the virtual link local address for the third IPv6 owner group

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 virtual-link-local-
address fe80:2000:f000:20::2
```

12. Set the third IPv6 owner group's priority to 250

```
[edit interfaces ge-1/0/0 unit 0]
set family inet6 address 2001:2000:f000:20::1/64 vrrp-inet6-group 3 priority 250
```

Results

Multiple IPv4 owner groups

```
[edit interfaces]
ge-1/0/0
    unit 0 {
        family inet {
            address 10.0.0.2/24 {
                vrrp-group 2 {
                    virtual-address 10.0.0.4;
                    accept-data;
                }
            }
            address 20.0.0.2/24 {
                vrrp-group 3 {
                    virtual-address 20.0.0.2;
                    priority 255;
                }
            }
            address 30.0.0.2/24 {
```

```
                vrrp-group 4 {
                    virtual-address 30.0.0.2;
                    priority 255;
                }
            }
        }
    }
```

Multiple IPv6 owner groups

```
[edit interfaces]
ge-1/0/0
    unit 0 {
        family inet6 {
            address 2001:4818:f000:20::1/64 {
                vrrp-inet6-group 1 {
                    virtual-inet6-address 2001:4818:f000:20::1;
                    virtual-link-local-address fe80:4818:f000:20::1;
                    priority 255;
                }
            }
            address fe80:4818:f000:13::2/64;
            address 2001:1000:f000:20::1/64 {
                vrrp-inet6-group 2 {
                    virtual-inet6-address 2001:1000:f000:20::1;
                    virtual-link-local-address fe80:1000:f000:20::1;
                    priority 255;
                }
            }
            address 2001:2000:f000:20::1/64 {
                vrrp-inet6-group 3 {
                    virtual-inet6-address 2001:2000:f000:20::2;
                    virtual-link-local-address fe80:2000:f000:20::2;
                    priority 250;
                }
            }
        }
    }
```

Multiple IPv4 and IPv6 owner groups

```
[edit interfaces]
ge-1/0/0
    unit 0 {
        family inet {
            address 10.0.0.1/24 {
                vrrp-group 5 {
                    virtual-address 10.0.0.1;
                    priority 255;
                }
            }
        }
        family inet6 {
            address 2001:4818:f000:20::1/64 {
                vrrp-inet6-group 1 {
                    virtual-inet6-address 2001:4818:f000:20::1;
                    virtual-link-local-address fe80:4818:f000:20::1;
                    priority 255;
                }
            }
            address 2001:1000:f000:20::1/64 {
                vrrp-inet6-group 2 {
                    virtual-inet6-address 2001:1000:f000:20::1;
                    virtual-link-local-address fe80:1000:f000:20::1;
                    priority 255;
                }
            }
            address 2001:2000:f000:20::1/64 {
                vrrp-inet6-group 3 {
                    virtual-inet6-address 2001:2000:f000:20::2;
                    virtual-link-local-address fe80:2000:f000:20::2;
                    priority 250;
                }
            }
        }
    }
```

Verification

To verify the configuration, run the `show interfaces ge-1/0/0` command, or use whichever name you assigned to the interface.

Configuring Inheritance for a VRRP Group

Junos OS enables you to configure VRRP groups on the various subnets of a VLAN to inherit the state and configuration of one of the groups, which is known as the *active VRRP group*. When the **vrrp-inherit-from** configuration statement is included in the configuration, only the active VRRP group, from which the other VRRP groups are inheriting the state, sends out frequent VRRP advertisements, and processes incoming VRRP advertisements. The groups that are inheriting the state do not process any incoming VRRP advertisement because the state is always inherited from the active VRRP group. However, the groups that are inheriting the state do send out VRRP advertisements once every 2 to 3 minutes to facilitate MAC address learning on the switches placed between the VRRP routers.

If the `vrrp-inherit-from` statement is not configured, each of the VRRP primary groups in the various subnets on the VLAN sends out separate VRRP advertisements and adds to the traffic on the VLAN.

To configure inheritance for a VRRP group, include the `vrrp-inherit-from` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet address` *`address`* `vrrp-group` *`group-id`*`]` hierarchy level.

```
[edit interfaces interface-name unit logical-unit-number family inet address address vrrp-group
group-id]
vrrp-inherit-from active-group group-index;
vrrp-inherit-from active-interface active-interface-name;
```

When you configure a group to inherit a state from another group, the inheriting groups and the active group must be on the same physical interface and logical system. However, the groups do not need to necessarily be on the same routing instance (as was in Junos OS releases earlier than 9.6), VLAN, or logical interface.

When you include the `vrrp-inherit-from` statement for a VRRP group, the VRRP group inherits the following parameters from the active group:

- **advertise-interval**
- **authentication-key**
- **authentication-type**
- **fast-interval**
- **preempt | no-preempt**
- **priority**
- **track interfaces**
- **track route**

However, you can configure the accept-data | no-accept-data statement for the group to specify whether the interface should accept packets destined for the virtual IP address.

Configuring an Interface to Accept All Packets Destined for the Virtual IP Address of a VRRP Group

In VRRP implementations where the router acting as the primary router is not the IP address owner—the IP address owner is the router that has the interface whose actual IP address is used as the virtual router's IP address (virtual IP address)— the primary router accepts only the ARP packets from the packets that are sent to the virtual IP address. Junos OS enables you to override this limitation with the help of the **accept-data** configuration. When the accept-data statement is included in the configuration, the primary router accepts all packets sent to the virtual IP address even when the primary router is not the IP address owner.

NOTE: If the primary router is the IP address owner or has its priority set to 255, the primary router, by default, accepts all packets addressed to the virtual IP address. In such cases, the **accept-data** configuration is not required.

To configure an interface to accept all packets sent to the virtual IP address, include the accept-data statement:

```
accept-data;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family (Inet | inet6) address *address* (vrrp-group | vrrp-inet6-group) *group-id*]

To prevent a primary router that is the IP address owner or has its priority set to 255 from accepting packets other than the ARP packets addressed to the virtual IP address, include the no-accept-data statement:

```
no-accept-data;
```

NOTE:

- If you want to restrict the incoming IP packets to ICMP packets only, you must configure firewall filters to accept only ICMP packets.
- If you include the `accept-data` statement, your routing platform configuration does not comply with RFC 3768 (see section 6.4.3 of RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*.

Configuring the Silent Period to Avoid Alarms Due to Delay in Receiving VRRP Advertisement Packets

The silent period starts when the interface state is changed from down to up. During this period, the Primary Down Event is ignored. Configure the silent period interval to avoid alarms caused by the delay or interruption of the incoming VRRP advertisement packets during the interface startup phase.

To configure the silent period interval that the Primary Down Event timer ignores, include the `startup-silent-period` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
startup-silent-period seconds;
```

NOTE: During the silent startup period, the `show vrrp detail` command output shows a value of 0 for `Master priority` and your IP address for `Master router`. These values indicate that the Primary selection is not completed yet, and these values can be ignored.

When you have configured **startup-silent-period**, the Primary Down Event is ignored until the **startup-silent-period** expires.

For example, configure a VRRP group, *vrrp-group1*, with an advertise interval of 1 second, startup silent period of 10 seconds, and an interface *interface1* with a priority less than 255.

When *interface1* transitions from down to up:

- The *vrrp-group1* group moves to the backup state, and starts the Primary Down Event timer (3 seconds; three times the value of the advertise interval, which is 1 second in this case).
- If no VRRP PDU is received during the 3-second period, the **startup-silent-period** (10 seconds in this case) is checked, and if the startup silent period has not expired, the Primary Down Event timer is

restarted. This is repeated until the **startup-silent-period** expires. In this example, the Primary Down Event timer runs four times (12 seconds) by the time the 10-second startup silent period expires.

- If no VRRP PDU is received by the end of the fourth 3-second cycle, *vrrp-group1* takes over the primary role.

Enabling the Distributed Periodic Packet Management Process for VRRP

Typically, VRRP advertisements are sent by the VRRP process (vrrpd) on the primary VRRP router at regular intervals to let other members of the group know that the VRRP primary router is operational.

When the vrrpd process is busy and does not send VRRP advertisements, the backup VRRP routers might assume that the primary router is down and take over as the primary router, causing unnecessary flaps. This takeover might occur even though the original primary router is still active and available and might resume sending advertisements after the traffic has decreased. To address this problem and to reduce the load on the vrrpd process, Junos OS uses the periodic packet management process (ppmd) to send VRRP advertisements on behalf of the vrrpd process. However, you can further delegate the job of sending VRRP advertisements to the distributed ppmd process that resides on the Packet Forwarding Engine.

The ability to delegate the sending of VRRP advertisements to the distributed ppmd process ensures that the VRRP advertisements are sent even when the ppmd process—which is now responsible for sending VRRP advertisements—is busy. Such delegation prevents the possibility of false alarms when the ppmd process is busy. The ability to delegate the sending of VRRP advertisements to distributed ppmd also adds to scalability because the load is shared across multiple ppmd instances and is not concentrated on any single unit.

NOTE: CPU-intensive VRRP advertisements, such as advertisements with MD5 authentication, continue to be processed by the VRRP process on the Routing Engine even when distributed ppmd is enabled.

NOTE: VRRP is supported by graceful Routing Engine switchover only in the case that PPM delegation is enabled (the default).

NOTE: Aggregated Ethernet and integrated routing and bridging (IRB) delegation is supported only for MPC line cards. Routing devices with inbuilt MPCs such as the MX104 and below do not support this feature.

To configure the distributed ppmd process to send VRRP advertisements, include the `delegate-processing` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
delegate-processing;
```

To configure the distributed ppmd process to send VRRP advertisements over aggregated Ethernet and IRB interfaces, include the `delegate-processing ae-irb` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
delegate-processing ae-irb;
```

Improving the Convergence Time for VRRP

You can enable faster convergence time for the configured Virtual Router Redundancy Protocol (VRRP), thereby reducing the traffic restoration time to less than 1 second. To improve the convergence time for the VRRP, perform the following tasks:

- **Configure the distributed periodic packet management process**—When the VRRP process is busy and does not send VRRP advertisements, the backup VRRP routers might assume that the primary router is down and take over as the primary router, causing unnecessary flaps. To address this problem and to reduce the load on the VRRP process, Junos OS uses the distributed periodic packet management (PPM) process to send VRRP advertisements on behalf of the VRRP process.

  To configure the distributed PPM process, include the `delegate-processing` statement at the `[edit protocols vrrp]` hierarchy level.

- **Disable the skew timer**—The skew timer in VRRP is used to ensure that two backup routers do not switch to the primary state at the same time in case of a failover situation. When there is only one primary router and one backup router in the network deployment, you can disable the skew timer, thereby reducing the time required to transition to the primary state.

  To disable the skew timer, include the `skew-timer-disable` statement at the `[edit protocols vrrp]` hierarchy level.

- **Configure the number of fast advertisements that can be missed by a backup router before it starts transitioning to the master state**—The backup router waits until a certain number of advertisement packets are lost after which it transitions to the primary state. This waiting time can be fatal in scenarios such as router failure or link failure. To avoid such a situation and to enable faster convergence time, in Junos OS Release 12.2 and later, you can configure a fast advertisement

interval value that specifies the number of fast advertisements that can be missed by a backup router before it starts transitioning to the primary state.

To configure the fast advertisement interval, include the `global-advertisements-threshold` statement at the `[edit protocols vrrp]` hierarchy level.

- **Configure inheritance of VRRP groups**—Junos OS enables you to configure VRRP groups on the various subnets of a virtual LAN (VLAN) to inherit the state and configuration of one of the groups, which is known as the active VRRP group. When the `vrrp-inherit-from` statement is included in the configuration, only the active VRRP group, from which the other VRRP groups inherit the state, sends out frequent VRRP advertisements and processes incoming VRRP advertisements. Use inherit groups for scaled configurations. For example, if you have 1000 VRRP groups with an advertisement interval of 100 ms, then use inherit groups.

  To configure inheritance for a VRRP group, include the `vrrp-inherit-from` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet address` *`address`* `vrrp-group` *`group-id`*`]` hierarchy level.

- **Disable duplicate address detection for IPv6 interfaces**—Starting with Junos OS Release 15.1, duplicate address detection is a feature of the Neighbor Discovery Protocol for IPv6. Duplicate address detection is enabled by default and determines whether an address is already in use by another node. When detection address detection is enabled, convergence time is high after an IPv6 interface that has been configured for VRRP tracking comes up. To disable duplicate address detection, include the `ipv6-duplicate-addr-detection-transmits 0` statement at the `[edit system internet-options]` hierarchy level. To disable duplicate address detection only for a specific interface, include the `dad-disable` statement at the `[edit interfaces` *`interface-name`* `unit` *`logical-unit-number`* `family inet6]` hierarchy level.

NOTE:

- Inheritance of VRRP groups is supported with all types of interfaces. Other measures to reduce convergence time, such as VRRP distribution, disabling skew timer, and reducing advertisement threshold.
- Compared to other routers, the convergence time and the traffic restoration time are less for MX Series routers with MPCs.
- Reduction in convergence time is applicable for all types of configurations at the physical interface but the convergence time might not be less than 1 second for all the configurations. The convergence time depends on the number of groups that are transitioning from the backup to the primary state and the interval at which these groups are transitioning.

Configuring VRRP to Improve Convergence Time

You can enable faster convergence time for the configured Virtual Router Redundancy Protocol (VRRP), thereby reducing the traffic restoration time to less than 1 second. To improve the convergence time for VRRP, perform the following tasks.

Before you begin, configure VRRP. See Configuring VRRP.

1. Configure the distributed periodic packet management (PPM) process to send VRRP advertisements when the VRRP process is busy.

```
[edit]
user@host# set protocols vrrp delegate-processing
```

2. Disable the skew timer to reduce the time required to transition to the primary state.

```
[edit]
user@host# set protocols vrrp skew-timer-disable
```

> **NOTE**: When there is only one primary router and one backup router in the network deployment, you can disable the skew timer, thereby reducing the time required to transition to the primary state.

3. Configure the number of fast advertisements that can be missed by a backup router before it starts transitioning to the primary state.

```
[edit]
user@host# set protocols vrrp global-advertisement-threshold advertisement-value
```

4. Configure VRRP groups on the various subnets of a VLAN to inherit the state and to configure one of the groups.

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-id
```

5. Verify the configuration.

```
[edit]
user@host# show protocols vrrp
```

NOTE:

- Inheritance of VRRP groups is supported with all types of interfaces. Other measures to reduce convergence time, such as VRRP distribution, disabling skew timer, and reducing advertisement threshold, are not applicable when VRRP is configured over integrated routing and bridging (IRB) interfaces, aggregated Ethernet interfaces, and multichassis link aggregation group (MC-LAG) interfaces.
- Compared to other routers, the convergence time and the traffic restoration time are less for MX Series routers with MPCs.
- Reduction in convergence time is applicable for all types of configurations at the physical interface, but the convergence time might not be less than 1 second for all the configurations. The convergence time depends on the number of groups that are transitioning from the backup to the primary state and the interval at which these groups are transitioning.

Tracing VRRP Operations

To trace VRRP operations, include the `traceoptions` statement at the `[edit protocols vrrp]` hierarchy level.

By default, VRRP logs the error, data carrier detect (DCD) configuration, and routing socket events in a file in the **/var/log** directory. By default, this file is named **/var/log/vrrpd**. The default file size is 1 megabyte (MB), and three files are created before the first one gets overwritten.

To change the configuration of the logging file, include the `traceoptions` statement at the `[edit protocols vrrp]` hierarchy level:

```
[edit protocols vrrp]
traceoptions {
    file filename <files number> <match regular-expression> <microsecond-stamp> <size size>
<world-readable | no-world-readable>;
    flag flag;
    no-remote-trace;
```

```
}
flag flag;
```

You can specify the following VRRP tracing flags:

- **all**—Trace all VRRP operations.
- **database**—Trace all database changes.
- **general**—Trace all general events.
- **interfaces**—Trace all interface changes.
- **normal**—Trace all normal events.
- **packets**—Trace all packets sent and received.
- **state**—Trace all state transitions.
- **timer**—Trace all timer events.

Example: Configuring VRRP for Load Sharing

IN THIS SECTION

If you do not want to dedicate a switch to be a VRRP backup (and therefore leave it idle unless the primary fails), you can create a load-sharing configuration in which each participating switch simultaneously acts as a primary and a backup.

One reason to use a load-sharing (active-active) configuration is that you are more likely to actively monitor and maintain both switches and notice if a problem occurs on either of them. If you use a configuration in which one switch is only a backup (an active-backup configuration), you might be less likely to pay attention to the backup switch while it is idle. In the worst case, this could lead to the

backup switch developing an undetected problem and not being able to perform adequately when a failover occurs.

Requirements

This example uses the following hardware and software components:

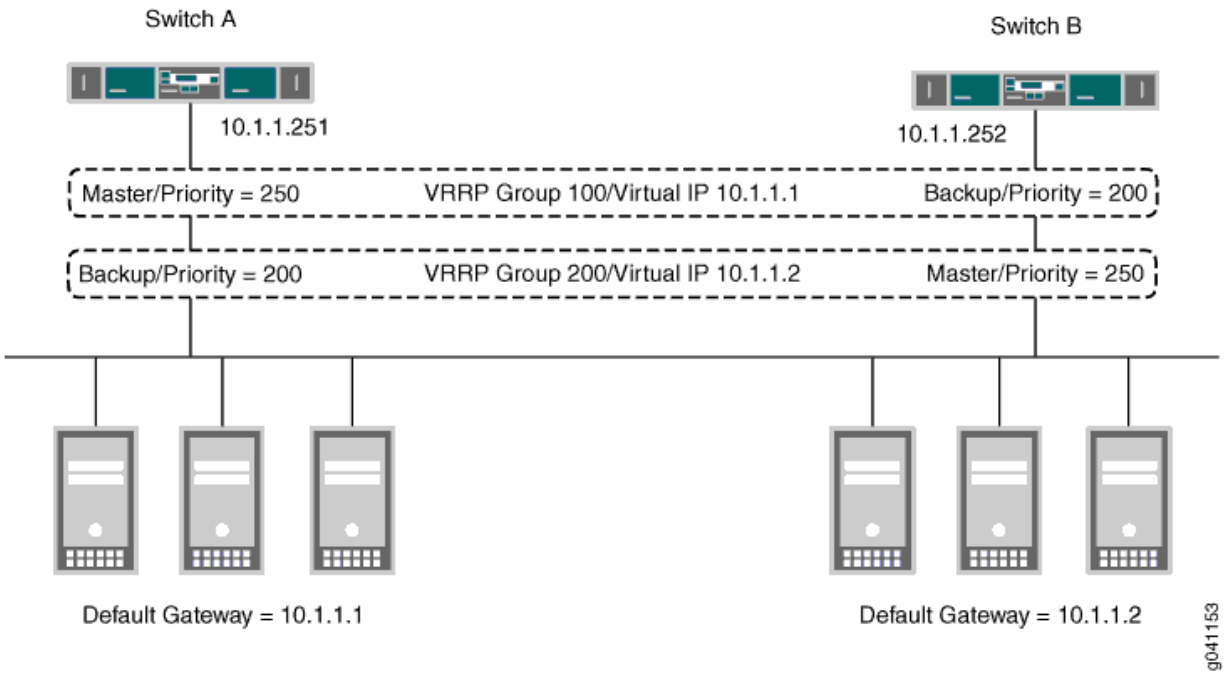
- Two switches
- Junos OS Release 11.3 or later
- Static routing or a dynamic routing protocol enabled on both switches.

Overview and Topology

IN THIS SECTION

This example uses two VRRP groups, each of which has its own virtual IP address. Devices on the LAN use one of these virtual IP addresses as their default gateway. If one of the switches fails, the other switch takes over for it. In the topology shown in Figure 28 on page 469, for example, Switch A is the primary for VRRP group 100. If Switch A fails, Switch B takes over and forwards traffic that the end devices send to the default gateway address 10.1.1.1.

Figure 28: VRRP Load-Sharing Configuration



This example shows a simple configuration to illustrate the basic steps for configuring two switches running VRRP to back each other up.Table 18 on page 469 lists VRRP settings for each switch.

Topology

Table 18: Settings for VRRP Load-Sharing Example

Switch A	Switch B
VRRP Group 100: • Interface address: 10.1.1.251 • VIP: 10.1.1.1 • Priority: 250	VRRP Group 100: • Interface address: 10.1.1.252 • VIP: 10.1.1.1 • Priority: 200

Table 18: Settings for VRRP Load-Sharing Example *(Continued)*

Switch A	Switch B
VRRP Group 200: • Interface address: 10.1.1.251 • VIP: 10.1.1.2 • Priority: 200	VRRP Group 200: • Interface address: 10.1.1.252 • VIP: 10.1.1.2 • Priority: 250

In addition to configuring the two switches as shown, you must configure your end devices so that some of them use one of the virtual IP addresses as their default gateway and the remaining end devices use the other virtual IP address as their default gateway.

Note that if a failover occurs, the remaining switch might be unable to handle all of the traffic, depending on the demand.

Configuring VRRP on Both Switches

IN THIS SECTION

Procedure

CLI Quick Configuration

Enter the following on Switch A:

```
[edit]
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 100 virtual-address
10.1.1.1
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 100 priority 250
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 200 virtual-address
10.1.1.2
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 200 priority 200
```

Enter the following on Switch B:

```
[edit]
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 100 virtual-address
10.1.1.1
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 100 priority 200
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 200 virtual-address
10.1.1.2
set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 200 priority 250
```

Step-by-Step Procedure

Configure the VRRP groups and priorities on Switch A:

1. Create VRRP group 100 on Switch A and configure the virtual IP address for the group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 100
virtual-address 10.1.1.1
```

2. Assign the VRRP priority for this interface in this group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 100
priority 250
```

3. Create VRRP group 200 on Switch A and configure the virtual IP address for the group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 200
virtual-address 10.1.1.2
```

4. Assign the VRRP priority for this interface in this group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 200
priority 200
```

Step-by-Step Procedure

Configure the VRRP groups and priorities on Switch B:

1. Create VRRP group 100 on Switch B and configure the virtual IP address for the group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 100
virtual-address 10.1.1.1
```

2. Assign the VRRP priority for this interface in this group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 100
priority 200
```

   Switch A remains the primary for group 100 because it has the highest priority for this group.

3. Create VRRP group 200 on Switch A and configure the virtual IP address for the group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.252/24 vrrp-group 200
virtual-address 10.1.1.2
```

4. Assign the VRRP priority for this interface in this group:

```
[edit]
user@switch# set interfaces xe-0/0/0 unit 0 family inet address 10.1.1.251/24 vrrp-group 200
priority 250
```

   Switch B becomes the primary for group 200 because it has the highest priority for this group.

Results

Display the results of the configuration on Switch A:

```
user@switch> show configuration
interfaces {
    xe-0/0/0 {
```

```
        unit 0 {
            family inet {
                address 10.1.1.251 {
                    vrrp-group 100 {
                        virtual address 10.1.1.1
                        priority 250
                    }
                    vrrp-group 200 {
                        virtual address 10.1.1.2
                        priority 200
                    }
                }
            }
        }
    }
}
```

Display the results of the configuration on Switch B:

```
user@switch> show configuration
interfaces {
    xe-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.1.252 {
                    vrrp-group 100 {
                        virtual address 10.1.1.1
                        priority 200
                    }
                    vrrp-group 200 {
                        virtual address 10.1.1.2
                        priority 250
                    }
                }
            }
        }
    }
}
```

Verification

IN THIS SECTION

Verifying that VRRP Is Working on Switch A

Purpose

Verify that VRRP is active on Switch A and that the primary and backup roles are correct.

Action

Use the following command to verify that VRRP is active on Switch A and that the switch is primary for group 100 and backup for group 200.

```
user@switch> show vrrp
Interface            State           Group       VR state      Timer               Type
Address
xe-0/0/0.0         up              100         master      A .0327         lcl      10.1.1.251
                                                                       vip      10.1.1.1
xe-0/0/0.0         up              200         backup      A .0327         lcl      10.1.1.251
                                                                       vip      10.1.1.2
```

Meaning

The `show vrrp` command displays fundamental information about the VRRP configuration. This output shows that both VRRP groups are active and that this switch has assumed the correct primary and backup roles. The **lcl** address is the physical address of the interface and the **vip** address is the virtual address shared by both switches. The **Timer** value (**A .0327**) indicates the remaining time (in seconds) in which this switch expects to receive a VRRP advertisement from the other switch. If an advertisement for group 200 does not arrive before the timer expires, Switch A asserts itself as the primary for this group.

Verifying that VRRP Is Working on Switch B

Purpose

Verify that VRRP is active on Switch B and that the primary and backup roles are correct.

Action

Use the following command to verify that VRRP is active on Switch B and that the switch is backup for group 100 and primary for group 200.

```
user@switch> show vrrp
Interface            State           Group       VR state      Timer               Type
Address
xe-0/0/0.0         up              100         backup      A .0327         lcl      10.1.1.252
                                                                        vip      10.1.1.1
xe-0/0/0.0         up              200         master      A .0327         lcl      10.1.1.252
                                                                        vip      10.1.1.2
```

Meaning

The `show vrrp` command displays fundamental information about the VRRP configuration. This output shows that both VRRP groups are active and that this switch has assumed the correct primary and backup roles. The **lcl** address is the physical address of the interface and the **vip** address is the virtual address shared by both switches. The **Timer** value (**A .0327**) indicates the remaining time (in seconds) in which this switch expects to receive a VRRP advertisement from the other switch. If an advertisement for group 100 does not arrive before the timer expires, Switch B asserts itself as the primary for this group.

Troubleshooting VRRP

IN THIS SECTION

Problem

Description

If you configure multiple VRRP groups on an interface (using multiple VLANs), traffic for some of the groups might be briefly dropped if a failover occurs. This can happen because the new primary must send gratuitous ARP replies for each VRRP group to update the ARP tables in the connected devices, and there is a short delay between each gratuitous ARP reply. Traffic sent by devices that have not yet received the gratuitous ARP reply is dropped (until the device receives the reply and learns the MAC address of the new primary).

Solution

Configure a failover delay so that the new primary delays sending gratuitous ARP replies for the period that you set. This allows the new primary to send the ARP replies for all of the VRRP groups simultaneously.

Change History Table

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

Release	Description
18.1R1	Primary router sends periodic VRRP advertisement messages to each virtual routers. The backup routers do not attempt to preempt the primary router unless it has higher priority. This eliminates service disruption unless a more preferred path becomes available. It is possible to administratively prohibit all preemption attempts, with the exception of a VRRP router becoming primary router of any virtual router associated with addresses it owns.
17.3R1	Starting in Junos OS release 17.3R1, if network-services is configured in IP mode, don't configure the same VRRP group ID for multiple VRRP sessions on the same physical interface unless VRRP delegation is disabled.
17.3R1	Starting in Junos OS release 17.3R1, if network-services is configured in enhanced-ip mode, you can use the same VRRP group ID for multiple VRRP sessions.
15.1	In Junos OS Release 15.1 and later, an adjusted priority can be zero.
15.1	Prior to Junos OS Release 15.1, an adjusted priority could not be zero.

13.2	Starting in Junos OS Release 13.2, VRRP nonstop active routing (NSR) is enabled only when you configure the `nonstop-routing` statement at the `[edit routing-options]` or `[edit logical system` *`logical-system-name`* `routing-options]` hierarchy level.

RELATED DOCUMENTATION

Understanding VRRP

PART 13

Performing Unified In-Service Software Upgrade (ISSU)

Understanding Unified ISSU

SUMMARY

Unified in-service software upgrade (ISSU) is a feature that minimizes traffic loss during the software upgrade process.

IN THIS SECTION

Getting Started with Unified In-Service Software Upgrade

The unified in-service software upgrade (ISSU) feature enables you to upgrade between two different Junos OS releases with no disruption on the control plane and with minimal disruption of traffic.

To quickly access the information you need, click on the link in Table 19 on page 479.

Table 19: Locating the Information You Need to Work With ISSU

Task You Need to Perform	Where The Information Is Located
Verify unified ISSU support for your device	"Unified ISSU System Requirements" on page 489
Perform a unified ISSU	Example: Performing a Unified ISSU
Verify that the unified ISSU is successful	Verifying a Unified ISSU

Unified ISSU takes advantage of the redundancy provided by dual Routing Engines and works in conjunction with the graceful Routing Engine switchover feature and the nonstop active routing feature.

Unified ISSU provides the following benefits:

- Eliminates network downtime during software image upgrades

- Reduces operating costs, while delivering higher service levels
- Allows fast implementation of new features

Understanding the Unified ISSU Process

IN THIS SECTION

This topic explains the unified ISSU processes that take place on a router.

Understanding the Unified ISSU Process on a Router

IN THIS SECTION

This topic describes the processes that take place on a router with dual Routing Engines when you initiate a unified in-service software upgrade (ISSU).

Unified ISSU Process on a Router

After you use the `request system software in-service-upgrade` command, the following process occurs.

In *Figure 1* through *Figure 6* below:

- A solid line indicates the high-speed internal link between a Routing Engine and a Packet Forwarding Engine.
- A dotted line indicates the messages exchanged between the Packet Forwarding Engine and the chassis process (chassisd) on the Routing Engine.
- RE0m and RE1b indicate primary and backup Routing Engines, respectively.

- The check mark indicates that the device is running the new version of software.

> **NOTE**: Unified ISSU can only upgrade up to three major releases ahead of the current release on a device. To upgrade to a release more than three releases ahead of the current release on a device, use the unified ISSU process to upgrade the device to one or more intermediate releases until the device is within three major releases of the target release.

1. The primary Routing Engine validates the router configuration to ensure that it can be committed when you use the new software version.

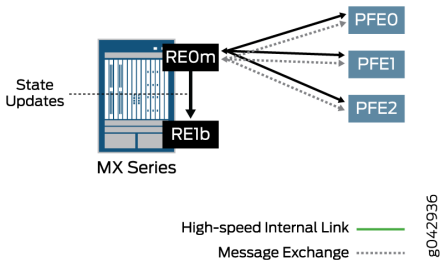
   Checks are made for the following:

   - Disk space is available for the **/var** file system on both Routing Engines.
   - The configuration is supported by a unified ISSU.
   - The PICs are supported by a unified ISSU.
   - Graceful Routing Engine switchover is enabled.
   - Nonstop active routing is enabled.

   These checks are the same as the checks made when you enter the `request system software validate in-service-upgrade` command. If there is insufficient disk space available on either of the Routing Engines, the unified ISSU process fails and returns an error message. However, unsupported PICs do not prevent a unified ISSU. If there are unsupported PICs, the system issues a warning to indicate that these PICs will restart during the upgrade. Similarly, if there is an unsupported protocol configured, the system issues a warning that packet loss might occur for the unsupported protocol during the upgrade.

   > **NOTE**: Starting in Junos OS Release 24.2R1, the primary Routing Engine will also run a check to see if the INDB has crashed. If an INDB crash is detected, the unified ISSU process will be cancelled.

2. **Figure 29: Device Status Before Starting a Unified ISSU**

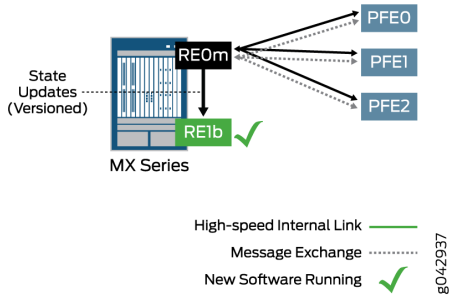


3. After the validation succeeds, the management process installs (copies) the new software image to the backup Routing Engine.

4. The backup Routing Engine is rebooted.

5. After the backup Routing Engine is rebooted and is running the new software, the kernel state synchronization process (ksyncd) synchronizes (copies) the configuration file and the kernel state from the primary Routing Engine.

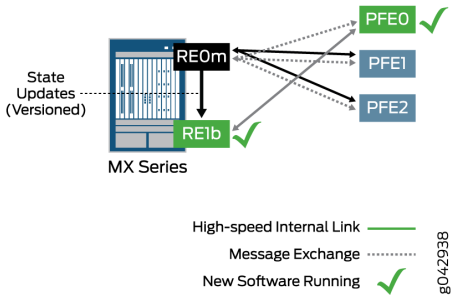
Figure 30: Device Status After the Backup Routing Engine Is Upgraded



6. After the configuration file and the kernel state are synchronized to the backup Routing Engine, the chassis process (chassisd) on the primary Routing Engine prepares other software processes for the unified ISSU. The chassis process informs the various software processes (such as rpd, apsd, bfdd, and so on) about the unified ISSU and waits for responses from them. When all the processes are ready, the chassis process sends an ISSU_PREPARE message to the FPCs installed in the router. You can display the unified ISSU process messages by using the `show log messages` command.

7. The Packet Forwarding Engine on each FPC saves its state and downloads the new software image from the backup Routing Engine. Next, each Packet Forwarding Engine sends an ISSU_READY message to the chassis process.

Figure 31: Device Status After One Packet Forwarding Engine Downloads the New Software

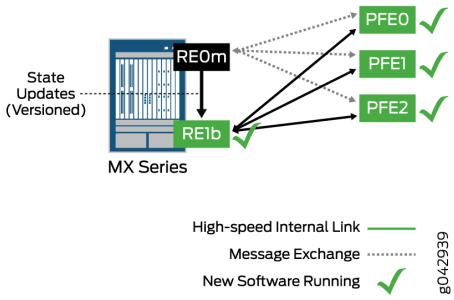


8. After receiving an ISSU_READY message from a Packet Forwarding Engine, the chassis process sends an ISSU_REBOOT message to the FPC on which the Packet Forwarding Engine resides. The FPC reboots with the new software image. After the FPC is rebooted, the Packet Forwarding Engine restores the FPC state, and a high-speed internal link is established with the backup Routing Engine running the new software. The chassis process link is also reestablished with the primary Routing Engine.

> **NOTE**: The Packet Forwarding Engine reboots that occur during a unified ISSU are known as the "dark window". You can expect to see up to 2 seconds of traffic loss during this window of downtime.

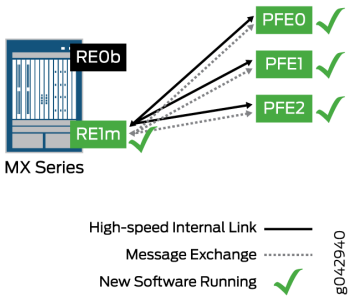
9. After all Packet Forwarding Engines have sent a READY message using the chassis process on the primary Routing Engine, other software processes are prepared for a Routing Engine switchover. The system is ready for a switchover at this point.

Figure 32: Device Status Before the Routing Engine Switchover



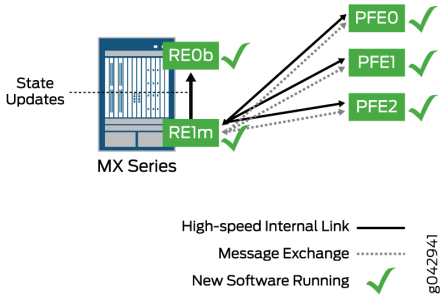
10. The Routing Engine switchover occurs, and the Routing Engine (re1) that was the backup now becomes the primary Routing Engine.

Figure 33: Device Status After the Routing Engine Switchover



11. The new backup Routing Engine is now upgraded to the new software image. (This step is skipped if you have specified the `no-old-master-upgrade` option in the `request system software in-service-upgrade` command.)

Figure 34: Device Status After the Unified ISSU Is Complete



12. When the backup Routing Engine has been successfully upgraded, the unified ISSU is complete.

Understanding the Unified ISSU Process on the TX Matrix Router

IN THIS SECTION

This topic describes the processes that take place on a TX Matrix router when you initiate a unified in-service software upgrade (ISSU).

Unified ISSU Process on the TX Matrix Router

This section describes the processes that take place on a TX Matrix router and the routers acting as connected line-card chassis (LCCs).

NOTE: A routing matrix is a multichassis architecture that consists of a TX Matrix router and from one to four T640 routers. From the perspective of the user interface, the routing matrix appears as a single router. The TX Matrix router controls all the T640 routers in the routing matrix.

Each router has dual Routing Engines.

After you use the *request system software in-service-upgrade* command on a TX Matrix router, the following process occurs:

1. The management process (mgd) on the primary Routing Engine of the TX Matrix router (global primary) checks the current configuration.

   Checks are made for the following:

   - Disk space is available for the **/var** file system on all Routing Engines.
   - The configuration is supported by a unified ISSU.
   - The PICs are supported by a unified ISSU.
   - Graceful Routing Engine switchover is enabled.
   - Nonstop active routing is enabled.

2. After successful validation of the configuration, the management process copies the new image to the backup Routing Engines on the TX Matrix router and the T640 routers.

3. The kernel synchronization process (ksyncd) on the backup Routing Engines synchronizes the kernels on the backup Routing Engines with the kernels on the primary Routing Engines.

4. The global backup Routing Engine is upgraded with the new software. Next the global backup Routing Engine is rebooted. Then the global backup Routing Engine synchronizes the configuration and kernel state from the global primary Routing Engine.

5. The LCC backup Routing Engines are upgraded and rebooted. Then the LCC backup Routing Engines connect with the upgraded global backup Routing Engine and synchronize the configuration and kernel state.

6. The unified ISSU control moves from the management process to the chassis process (chassisd). The chassis process informs the various software processes (such as rpd, apsd, bfdd, and so on) about the unified ISSU and waits for responses from them.

7. After receiving messages from the software processes indicating that the processes are ready for unified ISSU, the chassis process on the global primary Routing Engine sends messages to the chassis process on the routing nodes to start the unified ISSU.

8. The chassis process on the routing nodes sends ISSU_PREPARE messages to the field-replaceable units (FRUs), such as FPCs and intelligent PICs.

9. After receiving an ISSU_PREPARE message, the Packet Forwarding Engines save the current state information and download the new software image from the backup Routing Engines. Next, each Packet Forwarding Engine sends ISSU_READY messages to the chassis process. You can display the unified ISSU process messages by using the `show log messages` command.

10. After receiving an ISSU_READY message from the Packet Forwarding Engines, the chassis process sends an ISSU_REBOOT message to the FRUs. While the upgrade is in progress, the FRUs keep sending ISSU_IN_PROGRESS messages to the chassis process on the routing nodes. The chassis process on each routing node, in turn, sends an ISSU_IN_PROGRESS message to the chassis process on the global primary Routing Engine.

    NOTE: The Packet Forwarding Engine reboots that occur during a unified ISSU are designed to have a very short window of down time.

11. After the unified ISSU reboot, the Packet Forwarding Engines restore the saved state information and connect back to the routing nodes. The chassis process on each routing node sends an ISSU_READY message to the chassis process on the global primary Routing Engine. The CM_MSG_READY message from the chassis process on the routing nodes indicate that the unified ISSU is complete on the FRUs.

12. The unified ISSU control moves back to the management process on the global primary Routing Engine.

13. The management process initiates Routing Engine switchover on the primary Routing Engines.

14. Routing Engine switchover occurs on the TX Matrix router and the T640 routers.

15. After the switchover, the FRUs connect to the new primary Routing Engines. Then the chassis manager and Packet Forwarding Engine manager on the T640 router FRUs connect to the new primary Routing Engines on the T640 routers.

16. The management process on the global primary Routing Engine initiates the upgrade process on the old primary Routing Engines on the T640 routers. (This step is skipped if you have specified the `no-old-master-upgrade` option in the `request system software in-service-upgrade` command.)

17. After the Routing Engines that were previously the primaries on the T640 routers are upgraded, the management process initiates the upgrade of the Routing Engine that was previously the global primary on the TX Matrix router.

18. After a successful unified ISSU, the TX Matrix router and the T640 routers are rebooted if you specified the `reboot` option in the `request system software in-service-upgrade` command.

Understanding In-Service Software Upgrade (ISSU)

IN THIS SECTION

An in-service software upgrade (ISSU) enables you to upgrade between two different Junos OS releases with minimal disruption on the control plane and with minimal disruption of traffic. During an ISSU, the Junos OS runs in two separate virtual machines (VMs)—one VM is in the primary role acting as the primary Routing Engine, and the other VM is in the backup role acting as the backup Routing Engine. The Junos OS is upgraded on the backup VM. After a successful software upgrade, the backup VM then becomes the primary VM, and the original primary VM is no longer needed and is shut down.

Video: How Does ISSU Work on the QFX5100?

ISSU provides the following benefits:

- Eliminates network downtime during software image upgrades
- Reduces operating costs, while delivering higher service levels
- Allows fast implementation of new features

In-Service Software Upgrade Process

When you request an ISSU on a standalone device:

1. The management process (mgd) verifies that non-stop routing (NSR), graceful Routing Engine switchover (GRES), and non-stop bridging (NSB) are enabled.
2. The switch downloads and validates the software package.
3. The ISSU state machine spawns the backup Routing Engine (RE) with the newer software.
4. The ISSU state machine checks to see if the backup RE has synchronized all of the data with the primary RE.

5. The ISSU state machine moves the devices (for example, forwarding ASIC, FPGA, management port and serial console) from the primary RE to the backup RE.
6. The primary role is switched between the REs, so the backup RE becomes the primary RE.
7. The old primary RE is shut down.

Understanding In-Service Software Upgrade (ISSU) in ACX5000 Series Routers

IN THIS SECTION

An in-service software upgrade (ISSU) enables you to upgrade between two different Junos OS releases with minimal disruption on the control plane and with minimal disruption of traffic. During an ISSU, the Junos OS runs in two separate virtual machines (VMs)—one VM is in the primary role acting as the primary Routing Engine, and the other VM is in the backup role acting as the backup Routing Engine. The Junos OS is upgraded on the backup VM. After a successful software upgrade, the backup VM then becomes the primary VM, and the original primary VM is no longer needed and is shut down.

NOTE: ISSU is supported in Junos OS Release 15.1X54–D60 or later for ACX5000 Series routers.

ISSU provides the following benefits:

- Eliminates network downtime during software image upgrades
- Reduces operating costs, while delivering higher service levels
- Allows fast implementation of new features

In-Service Software Upgrade Process

When you request an ISSU on a standalone device:

1. The management process (mgd) verifies that non-stop routing (NSR), graceful Routing Engine switchover (GRES), and non-stop bridging (NSB) are enabled.

2. The router downloads and validates the software package.
3. The ISSU state machine spawns the backup Routing Engine (RE) with the newer software.
4. The ISSU state machine checks to see if the backup RE has synchronized all of the data with the primary RE.
5. The ISSU state machine moves the devices (for example, forwarding ASIC, FPGA, management port and serial console) from the primary RE to the backup RE.
6. The primary role is switched between the REs, so the backup RE becomes the primary RE.
7. The old primary RE is shut down.

RELATED DOCUMENTATION

Understanding High Availability Features on Juniper Networks Routers | 2

Unified ISSU System Requirements | 489

Example: Performing a Unified ISSU

request system software validate in-service-upgrade

Unified ISSU System Requirements

SUMMARY

Unified in-service software upgrade (ISSU) requires you to meet the device and configuration requirements listed below.

IN THIS SECTION

The unified in-service software upgrade (ISSU) feature enables you to upgrade your device between two different Junos OS releases with no disruption on the control plane and with minimal disruption of traffic. Unified ISSU is supported only on dual Routing Engine platforms. In addition, the graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) features must be enabled.

To verify platform support for unified ISSU, use Feature Explorer.

This section contains the following topics:

General Unified ISSU Considerations for All Platforms

Unified ISSU has the following caveats:

- To upgrade to Junos OS Releases 21.2R1 or 22.1R1, you need to include the `no-validate` option when issuing the in-service-upgrade command. The syntax for this command is `request system software in-service-upgrade /var/tmp/`*`package-name`*`.tgz no-validate`

  Junos OS releases prior to 20.4R2 do not support the `no-validate` option with unified ISSU. In order to upgrade from an older release to Junos OS Releases 21.2R1 or 22.1R1 with unified ISSU, you must first upgrade to a release that supports the `no-validate` option for unified ISSU, such as 20.4R2.

- Using unified ISSU to upgrade from an earlier Junos OS release to Junos OS Release 17.1R1 or later does not work if VPLS dynamic profiles are configured and enhanced subscriber management is not configured.

- The primary Routing Engine and backup Routing Engine must be running the same software version before you can perform a unified ISSU.

- The unified ISSU process is terminated and a message is displayed if the Junos OS version specified for installation is a version earlier than the one currently running on the device.

- The unified ISSU process is terminated if the specified upgrade has conflicts with the current configuration, components supported, and so forth.

- You cannot take PICs offline or bring them online during a unified ISSU.

- User-initiated GRES is blocked when the device is undergoing a unified ISSU.

- Unified ISSU does not support extension application packages developed with the Junos SDK.

- To downgrade from a unified ISSU-capable release to a previous software release (unified ISSU-capable or not), use the `request system software add` *`package-name`* command. Unlike an upgrade using the unified ISSU process, a downgrade using the `request system software add` *`package-name`* command can cause network disruptions and loss of data. For more information about the use of the `request system software add` *`package-name`* command, see the Junos OS Software Installation and Upgrade Guide.

- Unicast reverse-path-forwarding (RPF)-related statistics are not saved across a unified ISSU, and the unicast RPF counters are reset to zero during a unified ISSU.

- BGP session uptime and downtime statistics are not synchronized between the primary and backup Routing Engines during a unified ISSU. The backup Routing Engine maintains its own session uptime based on the time when the backup first becomes aware of the established sessions. For example, if the backup Routing Engine is rebooted (or if you run `restart routing` on the backup Routing Engine), the backup Routing Engine uptime is a short duration, because the backup has just learned about the established sessions. If the backup is operating when the BGP sessions first come up on the primary, the uptime on the primary and the uptime on the backup are almost the same duration. After a Routing Engine switchover, the new primary continues from the time left on the backup Routing Engine.

- If proxy ARP is enabled on your device, you must delete the `unconditional-src-learn` statement from the `[edit interfaces` *`interface-name`* `unit 0 family inet]` hierarchy level before the unified ISSU process begins and include it after the unified ISSU process is complete. Note that the `unconditional-src-learn` statement is not included by default.

Unified ISSU Considerations for MX Series Routers

Unified ISSU has the following caveats for MX Series routers:

- Unified ISSU for MX Series routers does not support the IEEE 802.1ag OAM and IEEE 802.3ah protocols.

- If clock synchronization is configured, the unified ISSU process terminates. For Junos OS Releases 22.1R1 and above, you can use the `request system software in-service-upgrade` command with the `handle-incompatible-config` option to automatically deactivate/activate clock synchronization for PTP and Synchronous Ethernet.

- On MX Series routers with MPC/MIC interfaces, the policers for transit traffic and statistics are disabled temporarily during the unified ISSU process.

- On MX Series MPCs, interface-specific and firewall filter statistics are preserved across a unified ISSU. During the unified ISSU, counter and policer operations are disabled.

- To preserve statistics across a unified ISSU on MX Series routers with MPC/MIC interfaces, the router stores the statistics data as binary large objects. The router collects the statistics before the unified ISSU is initialized, and restores the statistics after the unified ISSU completes. No statistics are collected during the unified ISSU process.
- After a unified ISSU operation is completed, an MPC reboot is required for MACsec to work.
- When there is a large number of subscribers configured, the Layer 2 scheduler can become oversubscribed. The unified ISSU process might terminate when the system runs out of schedulers. The system generates log messages with ISSU failures and CRC errors on the control plane. If you encounter this issue, please contact JTAC for assistance in eliminating the Layer 2 scheduler oversubscription in your configuration.
- MX Series routers support Link Aggregation Control Protocol (LACP) with fast hellos during unified ISSU. This support is disabled by default. You must enable the fast-hello-issu option on the main router and on the peer routers before starting unified ISSU. Note that the peer router must also be an MX Series router for this functionality to work.

Unified ISSU Considerations for PTX Series Routers

Unified ISSU has the following caveats for PTX Series routers:

- Link Aggregation Control Protocol (LACP) is not supported during unified ISSU on PTX Series routers. You must disable the `lacp` statement at the `[edit interfaces` *`interface-name`* `aggregated-ether-options]` hierarchy level before the unified ISSU process begins and enable it after the unified ISSU process is complete.

Unified ISSU Platform Support

To find out which platforms support ISSU, please use the Feature Explorer tool on the Juniper Networks website.

Unified ISSU Feature Support

Unified ISSU supports most Junos OS features starting in Junos OS Release 9.0. However, the following constraints apply:

- Link Aggregation Control Protocol (LACP)—Link changes are not processed until after the unified ISSU is complete.
- Automatic Protection Switching (APS)—Network changes are not processed until after the unified ISSU is complete.
- Ethernet Operation, Administration, and Management (OAM) as defined by IEEE 802.3ah and by IEEE 802.1ag—When a Routing Engine switchover occurs, the OAM hello message times out, triggering protocol convergence.
- Ethernet circuit cross-connect (CCC) encapsulation—Circuit changes are not processed until after the unified ISSU is complete.
- Logical systems—On devices that have logical systems configured on them, only the primary logical system supports unified ISSU.

> **NOTE**: When performing a unified ISSU from a FreeBSD 6.1-based Junos OS to an upgraded FreeBSD 10.x-based Junos OS, the configuration must be validated on a remote host or on a Routing Engine. The remote host or the Routing Engine must be running a Junos OS with an upgraded FreeBSD. In addition, only a few selected directories and files are preserved while upgrading from FreeBSD 6.1-based Junos OS to FreeBSD 10.x-based Junos OS. See Upgrading Junos OS with Upgraded FreeBSD.

Unified ISSU PIC Support Considerations

The following sections list information about PIC support for unified ISSU.

> **NOTE**: For information about ISSU support on individual PICs based on device and release, use the Feature Explorer tool.

> **NOTE**: For information about Flexible PIC Concentrator (FPC) types, FPC/PIC compatibility, and the initial Junos OS release in which a particular PIC is supported on an FPC, see the PIC guide for your platform.

PIC Considerations

Take the following PIC restrictions into consideration before performing a unified ISSU:

- Unsupported PICs—If a PIC is not supported by unified ISSU, at the beginning of the upgrade, the software issues a warning that the PIC will be taken offline. After the PIC is brought offline and the unified ISSU is complete, the PIC is brought back online with the new firmware.
- PIC combinations—For some PICs, newer Junos OS services can require significant Internet Processor ASIC memory, and some configuration rules might limit certain combinations of PICs on particular platforms. With a unified ISSU:
  - If a PIC combination is not supported by the software version that the device is being upgraded from, the validation check displays a message and terminates the upgrade.
  - If a PIC combination is not supported by the software version to which the device is being upgraded, the validation check displays a message and terminates the upgrade, even if the PIC combination is supported by the software version from which the device is being upgraded.
- Interface statistics—Interface statistics might be incorrect because:
  - During bootup of the new microkernel on the Packet Forwarding Engine, host-bound traffic is not handled and might be dropped, causing packet loss.
  - During the hardware update of the Packet Forwarding Engine and its interfaces, traffic is halted and discarded. (The duration of the hardware update depends on the number and type of interfaces and on the device configuration.)
  - During a unified ISSU, periodic statistics collection is halted. If hardware counters saturate or wrap around, the software does not display accurate interface statistics.
- CIR oversubscription—If oversubscription of the committed information rate (CIR) is configured on logical interfaces:
  - And the sum of the CIR exceeds the physical interface's bandwidth, after a unified ISSU is performed, each logical interface might not be given its original CIR.
  - And the sum of the delay buffer rate configured on logical interfaces exceeds the physical interface's bandwidth, after a unified ISSU is performed, each logical interface might not receive its original delay-buffer-rate calculation.

Unified ISSU Support on MX Series 3D Universal Edge Routers

The following sections list the Dense Port Concentrators (DPCs), Flexible PIC Concentrators (FPCs), Modular Port Concentrators (MPCs), and Modular Interface Cards (MICs) that are supported during a unified ISSU on MX Series routers.

Unified ISSU DPC and FPC Support on MX Series Routers

Unified ISSU supports all DPCs except the Multiservices DPC on MX Series routers. Unified ISSU also supports Type 2 FPC (`MX-FPC2`) and Type 3 FPC (`MX-FPC3`) on MX Series routers.

Unified ISSU MIC and MPC Support on MX Series Routers

Unified ISSU supports all the Modular Port Concentrators (MPCs) and Modular Interface Cards (MICs) listed in Table 20 on page 495 and Table 21 on page 497.

In the MPCs on MX Series routers, interface-specific and firewall filter statistics are preserved across a unified ISSU. During the unified ISSU, counter and policer operations are disabled.

To preserve statistics across a unified ISSU on MX Series routers with MPC/MIC interfaces, the router stores the statistics data as binary large objects. The router collects the statistics before the unified ISSU is initialized, and restores the statistics after the unified ISSU completes. No statistics are collected during the unified ISSU process.

To verify that statistics are preserved across the unified ISSU, you can issue CLI operational commands such as `show interfaces statistics` after the unified ISSU completes.

Table 20: Unified ISSU Support: MX Series Router MPCs

MPC Type	Number of Ports	Model Number	Platform
MPC1	—	MX-MPC1-3D	MX Series routers
MPC1E	—	MX-MPC1E-3D	MX Series routers
MPC1 Q	—	MX-MPC1-3D-Q	MX Series routers
MPC1E Q	—	MX-MPC1E-3D-Q	MX Series routers
MPC2	—	MX-MPC2-3D	MX Series routers
MPC2E	—	MX-MPC2E-3D	MX Series routers
MPC2 Q	—	MX-MPC2-3D-Q	MX Series routers

Table 20: Unified ISSU Support: MX Series Router MPCs *(Continued)*

MPC Type	Number of Ports	Model Number	Platform
MPC2E Q	—	MX-MPC2E-3D-Q	MX Series routers
MPC2 EQ	—	MX-MPC2-3D-EQ	MX Series routers
MPC2E EQ	—	MX-MPC2E-3D-EQ	MX Series routers
16x10GE MPC	16	MPC-3D-16XGE-SFPP	MX Series routers
MPC3E	—	MX-MPC3E-3D	MX Series routers
32x10GE MPC4E	32	MPC4E-3D-32XGE-SFPP	MX Series routers
2x100GE + 8x10GE MPC4E	10	MPC4E-3D-2CGE-8XGE	MX Series routers
6x40GE + 24x10GE MPC5E	30	MPC5E-40G10G	MX Series routers
6x40GE + 24x10GE MPC5EQ	30	MPC5EQ-40G10G	MX Series routers
2x100GE + 4x10GE MPC5E	6	MPC5E-100G10G	MX Series routers
2x100GE + 4x10GE MPC5EQ	6	MPC5EQ-100G10G	MX Series routers
MPC6E	2	MX2K-MPC6E	MX Series routers
MPC7E (multi-rate)	12	MPC7E-MRATE	MX Series routers
MPC7E 10G	40	MPC7E-10G	MX Series routers
MPC8E	—	MX2K-MPC8E	MX Series routers

Table 20: Unified ISSU Support: MX Series Router MPCs *(Continued)*

MPC Type	Number of Ports	Model Number	Platform
MPC9E	—	MX2K-MPC9E	MX Series routers

Table 21: Unified ISSU Support: MX Series Router MICs

MIC Type	Number of Ports	Model Number	Platform
ATM MIC with SFP	8	MIC-3D-8OC3-2OC12-ATM	MX Series routers
Channelized SONET/SDH OC192/ STM64 MIC with XFP	4	MIC-3D-1OC192-XFP	MX Series routers
Channelized OC3/STM1 (Multi-Rate) Circuit Emulation MIC with SFP	4	MIC-3D-4COC3-1COC12-CE	MX Series routers
Channelized E1/T1 Circuit Emulation MIC	16	MIC-3D-16CHE1-T1-CE	MX Series routers
Channelized SONET/SDH OC3/STM1 (Multi-Rate) MICs with SFP	4	MIC-3D-4CHOC3-2CHOC12	MX Series routers
Channelized SONET/SDH OC3/STM1 (Multi-Rate) MICs with SFP	8	MIC-3D-4CHOC3-2CHOC12	MX Series routers
Channelized DS3/E3 MIC	8	MIC-3D-8CHDS3-E3-B	MX Series routers
DS3/E3	8	MIC-3D-8DS3-E3	MX Series routers
See MIC MRATE for MIC Type	12	MIC MRATE	MX Series routers
40-Gigabit Ethernet MIC with QSFPP	2	MIC3-3D-2X40GE-QSFPP	MX Series routers

Table 21: Unified ISSU Support: MX Series Router MICs *(Continued)*

MIC Type	Number of Ports	Model Number	Platform
10-Gigabit Ethernet MIC with SFPP	10	MIC3-3D-10XGE-SFPP	MX Series routers
100-Gigabit Ethernet MIC with CXP	1	MIC3-3D-1X100GE-CXP	MX Series routers
100-Gigabit Ethernet MIC with CFP	1	MIC3-3D-1X100GE-CFP	MX Series routers
Gigabit Ethernet MIC with SFP	20	MIC-3D-20GE-SFP	MX Series routers
10-Gigabit Ethernet MIC with SFP+ (24 Ports)	24	MIC6-10G	MX Series routers
10-Gigabit Ethernet DWDM OTN MIC (non-OTN mode only)	24	MIC6-10G-OTN	MX Series routers
100-Gigabit Ethernet MIC with CFP2 (non-OTN mode only)	2	MIC6-100G-CFP2	MX Series routers
100-Gigabit Ethernet MIC with CXP (4 Ports)	4	MIC6-100G-CXP	MX Series routers
10-Gigabit Ethernet MICs with XFP	2	MIC-3D-2XGE-XFP	MX Series routers
10-Gigabit Ethernet MICs with XFP	4	MIC-3D-4XGE-XFP	MX Series routers
SONET/SDH OC3/STM1 (Multi-Rate) MICs with SFP	4	MIC-3D-4OC3OC12-1OC48	MX Series routers
SONET/SDH OC3/STM1 (Multi-Rate) MICs with SFP	8	MIC-3D-8OC3OC12-4OC48	MX Series routers
Tri-Rate Copper Ethernet MIC	40	MIC-3D-40GE-TX	MX Series routers

Table 21: Unified ISSU Support: MX Series Router MICs *(Continued)*

MIC Type	Number of Ports	Model Number	Platform
100-Gigabit DWDM OTn MIC with CFP2-ACO	1	MIC3-100G-DWDM	MX960 routers

NOTE: Consider the following guidelines before performing a unified ISSU on an MX Series router with ATM interfaces at scale:

- The PPP keepalive interval must be 10 seconds or greater. PPP requires three keepalives to fail before it brings down the session. Thirty seconds (ten seconds multiplied by three) provides a safe margin to maintain PPP sessions across the unified ISSU in case of any traffic loss during the operation. Configure the interval with the `keepalives` statement at the `[edit interfaces at-`*`interface-name`*`]` or `[edit interfaces at-`*`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level.
- The OAM F5 loopback cell period must be 20 seconds or greater to maintain ATM connectivity across the unified ISSU. Configure the interval with the `oam-period` statement at the `[edit interfaces at-`*`interface-name`* `unit` *`logical-unit-number`*`]` hierarchy level.

Unified ISSU Limitations on MX Series Routers

Unified ISSU is currently not supported when clock synchronization is configured for Synchronous Ethernet, Precision Time Protocol (PTP). For Junos OS Releases 22.1R1 and above, you can use the `request system software in-service-upgrade` command with the `handle-incompatible-config` option to automatically deactivate/activate clock synchronization for PTP and Synchronous Ethernet.

NOTE: Before enabling ISSU on MX routers, when upgrading from a Junos OS Release 14.1 or earlier to Junos OS Release 14.2 or later, you must disable IGMP snooping, and PIM snooping, in all protocol hierarchies. This includes the bridge-domain and routing-instances hierarchies.

NOTE: On MX Series routers with MPC/MIC interfaces, the policers for transit traffic and statistics are disabled temporarily during the unified ISSU process.

RELATED DOCUMENTATION

- Getting Started with Unified In-Service Software Upgrade
- Example: Performing a Unified ISSU
- Configuring LACP for Aggregated Ethernet Interfaces
- *request system software validate on (Junos OS with Upgraded FreeBSD)*

Performing a Unified ISSU

SUMMARY

Follow the steps below to perform a unified ISSU.

IN THIS SECTION

Best Practices for Performing a Unified ISSU

When you are planning to perform a unified in-service software upgrade (ISSU), choose a time when your network is as stable as possible. As with a normal upgrade, Telnet sessions, SNMP, and CLI access are briefly interrupted. In addition, the following restrictions apply:

- The primary Routing Engine and backup Routing Engine must be running the same software version before you can perform a unified ISSU.

- Verify that your platform supports the unified ISSU feature.
- Read the “Unified ISSU Considerations” topic in the chapter "Unified ISSU System Requirements" on page 489 to anticipate any special circumstances that might affect your upgrade.

Example: Performing a Unified ISSU

IN THIS SECTION

This example shows how to perform a unified in-service software upgrade (ISSU).

Requirements

This example uses the following hardware and software components:

- MX480 router with dual Routing Engines
- Junos OS Release 13.3R6 as the starting release
- Junos OS Release 14.1R4 as the ending release

Before You Begin

Before you perform a unified ISSU, be sure you:

- Perform a compatibility check to ensure that the software and hardware components and the configuration on the device support unified ISSU by using the *request system software validate in-service-upgrade* command
- Read the chapter "Unified ISSU System Requirements" on page 489 to anticipate any special circumstances that might affect your upgrade.
  - Verify that your platform supports the unified ISSU feature.
  - Verify that the field-replaceable units (FRUs) installed in your platform support the unified ISSU feature or that you can accept the results of performing the upgrade with some FRUs that do not support unified ISSU.
  - Verify that the protocols and features configured on your platform support the unified ISSU feature or that you can accept the results of performing the upgrade with some protocols and features that do not support unified ISSU.
- Download the software package from the Juniper Networks Support website at https://www.juniper.net/support/ and place the package on your local server.

BEST PRACTICE: When you access the Download Software web page for your device, record the md5 checksum. After downloading the software package to your device, confirm that it is not modified in any way by using the `file checksum md5` command.

NOTE: Starting with Junos OS Release 16.1R1, while performing a unified ISSU from a FreeBSD 6.1 based Junos OS to an upgraded FreeBSD 10.x based Junos OS, the configuration must be validated on a remote host or on a routing engine. The remote host or the routing engine must be running a Junos OS with an upgraded FreeBSD. In addition, only a few selected directories and files will be preserved while upgrading from FreeBSD 6.1 based Junos OS to FreeBSD 10.x based Junos OS. See Upgrading Junos OS with Upgraded FreeBSD and request system software valdiate on (Junos OS with Upgraded FreeBSD)

Overview

IN THIS SECTION

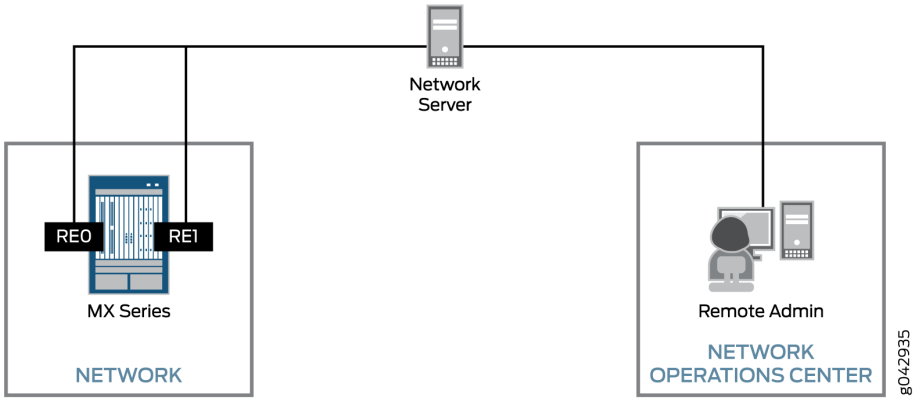
This procedure can be used to upgrade M Series, T Series, MX Series, EX Series, and PTX Series devices that have dual Routing Engines installed and support unified ISSU.

In the example, the hostnames, filenames, and FRUs are representational. When you perform the procedure on your device, the hostnames, filenames, and FRUs are different. The command output is truncated to only show the text of interest in this procedure.

Topology

Figure 35 on page 503 shows the topology used in this example.

Figure 35: Unified ISSU Example Topology



Configuration

There are variations of the procedure depending on if you want to install the new software on one or both Routing Engines and if you want to automatically reboot both Routing Engines or manually reboot one of the Routing Engines.

In all cases, you must verify that dual Routing Engines are installed and that graceful Routing Engine switchover (GRES) and nonstop active routing (NSR) are enabled. We recommend that you back up the device software before the upgrade.

To perform a unified ISSU, select the appropriate tasks from the following list:

- "Verifying Dual Routing Engines and Enabling GRES and NSR" on page 504
- "Verifying the Software Versions and Backing Up the Device Software" on page 506
- "Adjusting Timers and Changing Feature-Specific Configuration" on page 508
- "Upgrading and Rebooting Both Routing Engines Automatically" on page 510

- "Restoring Feature-Specific Configuration" on page 517
- "Upgrading Both Routing Engines and Rebooting the New Backup Routing Engine Manually" on page 519
- "Upgrading and Rebooting Only One Routing Engine" on page 528

Verifying Dual Routing Engines and Enabling GRES and NSR

IN THIS SECTION

- Procedure | 504

Procedure

Step-by-Step Procedure

Enabling GRES and NSR is required regardless of which variation of the unified ISSU procedure you use.

To verify that your device has dual Routing Engines and to enable GRES and NSR:

1. Log in to your device.
2. Verify that dual Routing Engines are installed in your device by using the `show chassis hardware` command.

```
user@host> show chassis hardware
Routing Engine 0 REV 01   740-051822   9013086837        RE-S-1800x4
Routing Engine 1 REV 01   740-051822   9013086740        RE-S-1800x4
```

   The command output contains lines listing Routing Engine 0 and Routing Engine 1.

3. By default, GRES is disabled; if you have not already done so, enable GRES by including the `graceful-switchover` statement at the `[edit chassis redundancy]` hierarchy level on the primary Routing Engine.

```
[edit ]
user@host# set chassis redundancy graceful-switchover
```

4. By default, NSR is disabled; if you have not already done so, enable NSR by including the `nonstop-routing` statement at the `[edit routing-options]` hierarchy level.

```
[edit]
user@host# set routing-options nonstop-routing
```

5. When you configure NSR, you must also include the `commit synchronize` statement at the `[edit system]` hierarchy level so that configuration changes are synchronized on both Routing Engines.

```
[edit]
user@host# set system commit synchronize
```

6. After you have verified your configuration and are satisfied with it, commit the changes by using the `commit` command.

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

When you enable GRES and commit the configuration, the CLI prompt changes to indicate which Routing Engine you are using. For example:

```
{master} [edit]
user@host#
```

7. Exit configuration mode by using the `exit` command.

```
{master} [edit]
user@host# exit
Exiting configuration mode
```

8. Verify that NSR is configured on the primary Routing Engine (`re0`) by using the `show task replication` command.

```
{master}
user@host> show task replication
        Stateful Replication: Enabled
```

```
        RE mode: Master

    Protocol                Synchronization Status
    OSPF                    Complete
    IS-IS                   Complete
```

In the output, verify that the `Synchronization Status` field displays `Complete`.

9. Verify that GRES is enabled on the backup Routing Engine (`re1`) by using the `show system switchover` command.

```
user@host> request routing-engine login re1
{backup}
user@host> show system switchover
Graceful switchover: On
Configuration database: Ready
Kernel database: Ready
Peer state: Steady State
```

In the output, verify that the `Graceful switchover` field state displays `On`. For more information about the `show system switchover` command, see *show system switchover*.

Verifying the Software Versions and Backing Up the Device Software

IN THIS SECTION

- Procedure | **506**

Procedure

Step-by-Step Procedure

Unified ISSU requires that both Routing Engines are running the same version of Junos OS before the upgrade. As a preventive measure in case any problems occur during an upgrade, it is a best practice to back up the system software to the device hard disk.

To verify the software versions and back up the device software:

1. Verify that the same version of Junos OS is installed and running on both Routing Engines by using the `show version` command.

```
{backup}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 13.3R6.5
JUNOS Base OS boot [13.3R6.5]
JUNOS Base OS Software Suite [13.3R6.5]
JUNOS 64-bit Kernel Software Suite [13.3R6.5]
JUNOS Crypto Software Suite [13.3R6.5]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [13.3R6.5]
JUNOS Packet Forwarding Engine Support (MX Common) [13.3R6.5]
JUNOS Online Documentation [13.3R6.5]

re1:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 13.3R6.5
JUNOS Base OS boot [13.3R6.5]
JUNOS Base OS Software Suite [13.3R6.5]
JUNOS 64-bit Kernel Software Suite [13.3R6.5]
JUNOS Crypto Software Suite [13.3R6.5]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [13.3R6.5]
JUNOS Packet Forwarding Engine Support (MX Common) [13.3R6.5]
JUNOS Online Documentation [13.3R6.5]
```

2. Back up the system software to the device hard disk by using the `request system snapshot` command on *each* Routing Engine.

NOTE: The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. After you issue the `request system snapshot` command, the device flash and hard

disks are identical. You can return to the previous version of the software only by booting the device from removable media.

```
{backup}
user@host> request system snapshot
user@host> request routing-engine login re0
{master}
user@host> request system snapshot
```

Adjusting Timers and Changing Feature-Specific Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

If you have any of the following feature-specific configuration on your device, perform the appropriate steps.

To adjust timers and change feature-specific configuration:

1. Bidirectional Forwarding Detection (BFD) sessions temporarily increase their detection and transmission timers during unified ISSU procedures. After the upgrade, these timers revert to the values in use before the unified ISSU started.

   If BFD is enabled on your device and you want to disable the BFD timer negotiation during the unified ISSU, include the `no-issu-timer-negotiation` statement at the `[edit protocols bfd]` hierarchy level.

```
{master} [edit]
user@host# set protocols bfd no-issu-timer-negotiation
```

> **NOTE**: If you include this statement, the BFD timers maintain their original values during the unified ISSU, and the BFD sessions might flap during the unified ISSU or Routing Engine switchover, depending on the detection intervals.

2. If proxy ARP is enabled on your M Series, MX Series, or EX 9200 Series device, remove the `unconditional-src-learn` statement from the `[edit interfaces interface-name unit 0 family inet]` hierarchy level.

   By default the statement is not included. This example shows the ge-0/0/1 interface only.

```
{master} [edit]
user@host# delete interfaces ge-0/0/1 unit 0 family inet unconditional-src-learn
```

3. If LACP is enabled on your PTX Series device, remove the `lacp` statement from the `[edit interfaces interface-name aggregated-ether-options]` hierarchy level.

```
{master} [edit]
user@host# delete interfaces aex aggregated-ether-options lacp
```

4. If ATM Point-to-Point Protocol (PPP) is enabled on your M Series or T Series device, set the keepalive interval to 10 seconds or greater.

   PPP requires three keepalives to fail before it brings down the session. Thirty seconds (10 seconds x three) provides a safe margin to maintain PPP sessions in case of any traffic loss during the unified ISSU operation.

   This example shows the at-0/0/1 interface only.

```
{master} [edit]
user@host# set interfaces at-0/0/1 unit 0 keepalives interval 10
```

5. If ATM OAM is enabled on your M Series or T Series device, set the OAM F5 loopback cell period to 20 seconds or greater to maintain ATM connectivity across the unified ISSU.

   Include the `oam-period` statement at the `[edit interfaces interface-name unit logical-unit-number]` hierarchy level and specify 20 seconds. This example shows the at-0/0/1 interface only.

```
{master} [edit]
user@host# set interfaces at-0/0/1 unit 0 oam-period 20
```

6. After you have verified your configuration and are satisfied with it, commit the changes by using the `commit` command.

```
{master} [edit]
user@host# commit
commit complete
```

7. Exit configuration mode by using the **exit** command.

```
{master} [edit]
user@host# exit
{master}
user@host>
```

Upgrading and Rebooting Both Routing Engines Automatically

IN THIS SECTION

Procedure

Step-by-Step Procedure

In this procedure, both Routing Engines automatically reboot. Rebooting both Routing Engines automatically is the most common scenario. Variations to this procedure are described in other sections.

Table 22 on page 510 shows the Routing Engine status prior to starting the unified ISSU.

Table 22: Routing Engine Status Before Upgrading

RE0	RE1
Primary	Backup
Old software version installed	Old software version installed

Table 22: Routing Engine Status Before Upgrading *(Continued)*

RE0	RE1
Old software version running	Old software version running

To upgrade and reboot both Routing Engines automatically:

1. Copy the Junos OS software package to the device by using the `file copy ftp://`*`username@hostname.net/filename`* `/var/tmp/`*`filename`* command.

   We recommend that you copy the package to the **/var/tmp** directory, which is a large file system on the hard disk.

   ```
   {master}
   user@host> file copy ftp://myid@myhost.mydomain.net/jinstall64-14.1R4.10-domestic-signed.tgz
   /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
   ```

   BEST PRACTICE: When you access the Download Software web page for your device, record the md5 checksum. After downloading the software package to your device, confirm that it is not modified in any way by using the `file checksum md5` command.

2. On the primary Routing Engine, start the upgrade by using the `request system software in-service-upgrade` *`package-name`* `reboot` command.

   NOTE: Do not try running any additional commands until after the `Connection closed` message is displayed and your session is disconnected.

   ```
   {master}
   user@host> request system software in-service-upgrade
   /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz reboot
   Checking compatibility with configuration
   Initializing...
   Using jbase-13.3R6.5
   Verified manifest signed by PackageProductionEc_2015
   Using /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
   Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
   Using jinstall64-14.1R4.10-domestic.tgz
   ```

```
Using jbundle64-14.1R4.10-domestic.tgz
Checking jbundle requirements on /
Using jbase-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jbase-14.1R4.10 signed by PackageProductionEc_2015
Using /var/v/c/tmp/jbundle/jboot-14.1R4.10.tgz
Using jcrypto64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jcrypto64-14.1R4.10 signed by PackageProductionEc_2015
Using jdocs-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jdocs-14.1R4.10 signed by PackageProductionEc_2015
Using jkernel64-14.1R4.10.tgz
Using jpfe-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M10-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M120-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M160-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M320-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M40-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M7i-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-T-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X2000-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X960-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-common-14.1R4.10.tgz
Using jplatform-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jplatform-14.1R4.10 signed by PackageProductionEc_2015
Using jroute-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jroute-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime64-14.1R4.10 signed by PackageProductionEc_2015
Using jservices-14.1R4.10.tgz
Using jservices-crypto-14.1R4.10.tgz
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
mgd: commit complete
Validation succeeded
ISSU: Preparing Backup RE
```

```
Pushing /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz to re1:/var/tmp/
jinstall64-14.1R4.10-domestic-signed.tgz
Installing package '/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz' ...
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionRSA_2015
Adding jinstall64...
Verified manifest signed by PackageProductionEc_2015

WARNING:     This package will load JUNOS 14.1R4.10 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in /var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving state for rollback ...
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
```

```
  FPC 0          Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
ISSU: Upgrading Old Master RE
Installing package '/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz' ...
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionRSA_2015
Adding jinstall64...
Verified manifest signed by PackageProductionEc_2015

WARNING:     This package will load JUNOS 14.1R4.10 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in /var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall64-14.1R4.10-domestic-signed.tgz ...
Saving state for rollback ...
ISSU: Old Master Upgrade Done
ISSU: IDLE
Shutdown NOW!
[pid 10149]

{backup}
user@host>

{backup}
user@host>
*** FINAL System shutdown message from user@host ***
```

```
System going down IMMEDIATELY


Connection closed by foreign host.
```

When the Routing Engine that was previously the primary is rebooted, you are logged out of the device.

3. Wait a few minutes and then log in to the device again.

   Table 23 on page 515 shows the Routing Engine status after the unified ISSU.

   Table 23: Routing Engine Status After Upgrading and Rebooting Both Routing Engines

RE0	RE1
Backup	Primary
New software version installed	New software version installed
New software version running	New software version running

   You are logged in to the new backup Routing Engine (`re0`).

4. Verify that both Routing Engines have been upgraded by using the `show version` command.

```
{backup}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]
```

```
re1:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]
```

5. If you want to, you can optionally display the unified ISSU log messages by using the `show log messages` command.

6. If you want to, you can optionally make `re0` the primary Routing Engine by using the `request chassis routing-engine master acquire` command.

```
{backup}
user@host> request chassis routing-engine master
acquire
Attempt to become the master routing engine ? [yes,no] (no) yes

Resolving mastership...
Complete. The local routing engine becomes the master.

{master}
user@host>
```

Table 24 on page 516 shows the Routing Engine status after Step 5 is completed.

Table 24: Routing Engine Status After Upgrading, Rebooting, and Switching Primary Role

RE0	RE1
Primary	Backup
New software version installed	New software version installed

Table 24: Routing Engine Status After Upgrading, Rebooting, and Switching Primary Role *(Continued)*

RE0	RE1
New software version running	New software version running

7. Perform the applicable steps in "Restoring Feature-Specific Configuration" on page 517.

8. If you are satisfied with the results of your testing, you can optionally back up the system software to the device's hard disk by using the `request system snapshot` command on *each* Routing Engine.

> **NOTE**: The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. After you issue the `request system snapshot` command, you cannot easily return to the previous version of the software, because the device flash and hard disks are identical. To return to the previous version of the software, you must boot the device from removable media.

```
{master}
user@host> request system snapshot
user@host> request routing-engine login re1
{backup}
user@host> request system snapshot
```

Restoring Feature-Specific Configuration

IN THIS SECTION

- Procedure | 517

Procedure

Step-by-Step Procedure

If you have any of the following feature-specific configuration on your device, perform the appropriate steps.

To restore feature-specific configuration:

1. If BFD is enabled on your device and you previously disabled the BFD timer negotiation, delete the `no-issu-timer-negotiation` statement at the `[edit protocols bfd]` hierarchy level.

```
{master} [edit]
user@host# delete protocols bfd no-issu-timer-negotiation
```

2. If proxy ARP is enabled on your M Series, MX Series, or EX9200 device and you previously removed the `unconditional-src-learn` statement, include the statement again.

   This example shows the ge-0/0/1 interface only.

```
{master} [edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet unconditional-src-learn
```

3. If LACP is enabled on your PTX Series device and you previously removed the `lacp` statement, include the statement again.

```
{master} [edit]
user@host# set interfaces aex aggregated-ether-options lacp
```

4. If ATM PPP is enabled on your M Series or T Series device and you previously set the keepalive interval to 10 seconds or greater, restore the original value.

   This example shows the at-0/0/1 interface only and shows the interval being set to the default 3 seconds.

```
{master} [edit]
user@host# set interfaces at-0/0/1 unit 0 keepalives interval 3
```

5. If ATM OAM is enabled on your M Series or T Series device and you previously set the OAM F5 loopback cell period to 20 seconds or greater, change the configuration back to the original value.

   This example shows the at-0/0/1 interface only and shows the period being set to 10 seconds.

```
{master} [edit]
user@host# set interfaces at-0/0/1 unit 0 oam-period 10
```

6. After you have verified your configuration and are satisfied with it, commit the changes by using the `commit` command.

```
{master} [edit]
user@host# commit
commit complete
```

7. Exit configuration mode by using the **exit** command.

```
{master} [edit]
user@host# exit
{master}
user@host>
```

Upgrading Both Routing Engines and Rebooting the New Backup Routing Engine Manually

IN THIS SECTION

Procedure

Step-by-Step Procedure

In certain circumstances, you might want to install the new software on only one Routing Engine and reboot only the primary until after you can test the new software. A Routing Engine does not start running the new software until after it is rebooted.

The advantage is if the results of your testing requires you to downgrade the software, you can switch Routing Engines to run the old software on one Routing Engine and then install the old software on the other Routing Engine. This is not the typical scenario.

To upgrade both Routing Engines and to reboot the new backup Routing Engine manually:

1. Perform the steps in "Verifying Dual Routing Engines and Enabling GRES and NSR" on page 504.
2. Perform the steps in "Verifying the Software Versions and Backing Up the Device Software" on page 506.

3. Perform the steps in "Adjusting Timers and Changing Feature-Specific Configuration" on page 508.

4. Copy the Junos OS software package to the device using the `file copy ftp://username@hostname.net/filename /var/tmp/filename` command.

   We recommend that you copy the package to the **/var/tmp** directory, which is a large file system on the hard disk.

```
{master}
user@host> file copy ftp://myid@myhost.mydomain.net/jinstall64-14.1R4.10-domestic-signed.tgz
/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
```

BEST PRACTICE: When you access the Download Software web page for your device, record the md5 checksum. After downloading the software package to your device, confirm that it is not modified in any way by using the `file checksum md5` command.

Table 25 on page 520 shows the Routing Engine status prior to starting the unified ISSU.

Table 25: Routing Engine Status Before Upgrading and Manually Rebooting the Backup Routing Engine

RE0	RE1
Primary	Backup
Old software version installed	Old software version installed
Old software version running	Old software version running

5. On the primary Routing Engine, start the upgrade by using the `request system software in-service-upgrade package-name` command without the reboot option.

```
{master}
user@host> request system software in-service-upgrade
/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
Checking compatibility with configuration
Initializing...
```

```
Using jbase-13.3R6.5
Verified manifest signed by PackageProductionEc_2015
Using /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Using jinstall64-14.1R4.10-domestic.tgz
Using jbundle64-14.1R4.10-domestic.tgz
Checking jbundle requirements on /
Using jbase-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jbase-14.1R4.10 signed by PackageProductionEc_2015
Using /var/v/c/tmp/jbundle/jboot-14.1R4.10.tgz
Using jcrypto64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jcrypto64-14.1R4.10 signed by PackageProductionEc_2015
Using jdocs-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jdocs-14.1R4.10 signed by PackageProductionEc_2015
Using jkernel64-14.1R4.10.tgz
Using jpfe-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M10-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M120-14.1R4.10.tgz

Verified SHA1 checksum of jpfe-M160-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M320-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M40-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M7i-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-T-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X2000-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X960-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-common-14.1R4.10.tgz
Using jplatform-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jplatform-14.1R4.10 signed by PackageProductionEc_2015
Using jroute-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jroute-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime64-14.1R4.10 signed by PackageProductionEc_2015
Using jservices-14.1R4.10.tgz
```

```
Using jservices-crypto-14.1R4.10.tgz
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
mgd: commit complete
Validation succeeded
ISSU: Preparing Backup RE
Pushing /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz to re1:/var/tmp/
jinstall64-14.1R4.10-domestic-signed.tgz
Installing package '/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz' ...
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionRSA_2015
Adding jinstall64...
Verified manifest signed by PackageProductionEc_2015

WARNING:     This package will load JUNOS 14.1R4.10 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in /var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving state for rollback ...
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
```

```
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
ISSU: Upgrading Old Master RE
Installing package '/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz' ...
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionRSA_2015
Adding jinstall64...
Verified manifest signed by PackageProductionEc_2015

WARNING:     This package will load JUNOS 14.1R4.10 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.


Saving the config files ...
NOTICE: uncommitted changes have been saved in /var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall64-14.1R4.10-domestic-signed.tgz ...
Saving state for rollback ...
ISSU: Old Master Upgrade Done
ISSU: IDLE
```

Table 26 on page 524 shows the Routing Engine status after the unified ISSU and before manually rebooting the backup Routing Engine.

Table 26: Routing Engine Status After Upgrading and Before Manually Rebooting the Backup Routing Engine

RE0	RE1
Backup	Primary
New software version installed	New software version installed
Old software version running	New software version running

6. Verify that the new backup, (old primary) Routing Engine (`re0`), is still running the previous software image and that the new primary Routing Engine (`re1`) is running the new software image, by using the `show version` command.

```
{backup}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 13.3R6.5
JUNOS Base OS boot [13.3R6.5]
JUNOS Base OS Software Suite [13.3R6.5]
JUNOS 64-bit Kernel Software Suite [13.3R6.5]
JUNOS Crypto Software Suite [13.3R6.5]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [13.3R6.5]
JUNOS Packet Forwarding Engine Support (MX Common) [13.3R6.5]
JUNOS Online Documentation [13.3R6.5]

re1:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
```

```
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]
```

7. At this point, if you do not want to install the newer software version on the new backup Routing Engine (re0), issue the `request system software delete` *package-name* command on it.

   Otherwise, to complete the upgrade, go to the next step.

8. Reboot the new backup Routing Engine (re0) by issuing the `request system reboot` command.

```
{backup}
user@host> request system reboot
Reboot the system ? [yes,no] (no) yes


*** FINAL System shutdown message from remote@host ***

System going down IMMEDIATELY


Shutdown NOW!
[pid 38432]

{backup}
user@home> Connection closed by foreign host.
```

If you are not on the console port, you are disconnected from the device session.

Table 27 on page 525 shows the Routing Engine status after the unified ISSU, after rebooting the backup Routing Engine, but before switching primary role.

Table 27: Routing Engine Status After Upgrading, Manually Rebooting, and Before Switching Primary Role

REO	RE1
Backup	Primary
New software version installed	New software version installed

Table 27: Routing Engine Status After Upgrading, Manually Rebooting, and Before Switching Primary Role *(Continued)*

RE0	RE1
New software version running	New software version running

9. Wait a few minutes, then log in to the device again.

   You are logged in to the new backup Routing Engine (`re0`).

10. Verify that both Routing Engines have been upgraded by using the `show version` command.

```
{backup}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]

re1:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]
```

11. If you want to, you can optionally display the unified ISSU log messages by using the `show log messages` command.

12. If you want to, you can optionally make `re0` the primary Routing Engine by using the `request chassis routing-engine master acquire` command:

```
{backup}
user@host> request chassis routing-engine master
acquire
Attempt to become the master routing engine ? [yes,no] (no) yes

Resolving mastership...
Complete. The local routing engine becomes the master.

{master}
user@host>
```

Table 28 on page 527 shows the Routing Engine status after the unified ISSU, after rebooting the backup Routing Engine, and after switching primary role.

Table 28: Routing Engine Status After Upgrading, Manually Rebooting, and Switching Primary Role

RE0	RE1
Primary	Backup
New software version installed	New software version installed
New software version running	New software version running

13. Perform the applicable steps in "Restoring Feature-Specific Configuration" on page 517.

14. If you are satisfied with the results of your testing, you can optionally back up the system software to the device's hard disk by using the `request system snapshot` command on *each* Routing Engine.

NOTE: The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. After you issue the `request system snapshot` command, you cannot easily return to the previous version of the software, because the device flash and hard

disks are identical. To return to the previous version of the software, you must boot the device from removable media.

```
{master}
user@host> request system snapshot
user@host> request routing-engine login re1
{backup}
user@host> request system snapshot
```

Upgrading and Rebooting Only One Routing Engine

IN THIS SECTION

Procedure

Step-by-Step Procedure

In certain circumstances you might want to install the new software on only one Routing Engine.

The advantage is if the results of your testing requires you to downgrade the software, you can switch Routing Engines to run the old software on one Routing Engine and then install the old software on the other Routing Engine. This is not the typical scenario.

Table 29 on page 528 shows the Routing Engine status prior to starting the unified ISSU.

Table 29: Routing Engine Status Before Upgrading and Rebooting One Routing Engine

RE0	RE1
Primary	Backup
Old software version installed	Old software version installed
Old software version running	Old software version running

To upgrade and rebooting only one Routing Engine:

1. Perform the steps in "Verifying Dual Routing Engines and Enabling GRES and NSR" on page 504.

2. Perform the steps in "Verifying the Software Versions and Backing Up the Device Software" on page 506.

3. Perform the applicable steps in "Adjusting Timers and Changing Feature-Specific Configuration" on page 508.

4. Copy the Junos OS software package to the device by using the `file copy ftp://`*`username`*`@`*`hostname.net`*`/`*`filename`* `/var/tmp/`*`filename`* command.

   We recommend that you copy the package to the **/var/tmp** directory, which is a large file system on the hard disk.

   ```
   {master}
   user@host> file copy ftp://myid@myhost.mydomain.net/jinstall64-14.1R4.10-domestic-signed.tgz
   /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
   ```

   BEST PRACTICE: When you access the Download Software web page for your device, record the md5 checksum. After downloading the software package to your device, confirm that it is not modified in any way by using the `file checksum md5` command.

5. On the primary Routing Engine, start the upgrade by using the `request system software in-service-upgrade` *`package-name`* `no-old-master-upgrade` command.

   ```
   {master}
   user@host> request system software in-service-upgrade
   /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz no-old-master-upgrade
   Checking compatibility with configuration
   Initializing...
   Using jbase-13.3R6.5
   Verified manifest signed by PackageProductionEc_2015
   Using /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
   Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
   Using jinstall64-14.1R4.10-domestic.tgz
   Using jbundle64-14.1R4.10-domestic.tgz
   Checking jbundle requirements on /
   Using jbase-14.1R4.10.tgz
   ```

```
Verified manifest signed by PackageProductionEc_2015
Verified jbase-14.1R4.10 signed by PackageProductionEc_2015
Using /var/v/c/tmp/jbundle/jboot-14.1R4.10.tgz
Using jcrypto64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jcrypto64-14.1R4.10 signed by PackageProductionEc_2015
Using jdocs-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jdocs-14.1R4.10 signed by PackageProductionEc_2015
Using jkernel64-14.1R4.10.tgz
Using jpfe-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M10-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M120-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M160-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M320-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M40-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M7i-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-T-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X2000-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X960-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-common-14.1R4.10.tgz
Using jplatform-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jplatform-14.1R4.10 signed by PackageProductionEc_2015
Using jroute-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jroute-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime64-14.1R4.10 signed by PackageProductionEc_2015
Using jservices-14.1R4.10.tgz
Using jservices-crypto-14.1R4.10.tgz
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
mgd: commit complete
Validation succeeded
ISSU: Preparing Backup RE
Pushing /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz to re1:/var/tmp/
jinstall64-14.1R4.10-domestic-signed.tgz
Installing package '/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz' ...
```

```
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionRSA_2015
Adding jinstall64...
Verified manifest signed by PackageProductionEc_2015

WARNING:     This package will load JUNOS 14.1R4.10 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in /var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving state for rollback ...
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
```

```
ISSU: RE switchover Done
Skipping Old Master Upgrade
ISSU: IDLE
```

Table 30 on page 532 shows the Routing Engine status after the unified ISSU upgrades the primary Routing Engine but before the backup Routing Engine is upgraded.

Table 30: Routing Engine Status After Upgrading One Routing Engine and Before Upgrading the Other Routing Engine

REO	RE1
Backup	Primary
Old software version installed	New software version installed
Old software version running	New software version running

6. Verify that the new backup, (old primary) Routing Engine (`re0`), is still running the previous software image and that the new primary Routing Engine (`re1`) is running the new software image, by using the `show version` command.

```
{backup}
user@host> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 13.3R6.5
JUNOS Base OS boot [13.3R6.5]
JUNOS Base OS Software Suite [13.3R6.5]
JUNOS 64-bit Kernel Software Suite [13.3R6.5]
JUNOS Crypto Software Suite [13.3R6.5]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [13.3R6.5]
JUNOS Packet Forwarding Engine Support (MX Common) [13.3R6.5]
JUNOS Online Documentation [13.3R6.5]

re1:
--------------------------------------------------------------------------
Hostname: host
```

```
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]
```

7. If your testing is complete and you want to install the new software on the backup Routing Engine, you must first disable GRES and NSR on both Routing Engines and commit the configuration.

```
{backup} [edit ]
user@host# delete chassis redundancy graceful-switchover
user@host# delete routing-options nonstop-routing
user@host# commit
warning: Graceful-switchover is enabled, commit on backup is not recommended
Continue commit on backup RE? [yes,no] (no) yes
re0:
configuration check succeeds
re1:
commit complete
re0:
commit complete
[edit ]
user@host#
```

8. Install the new software on the backup Routing Engine (`re0`) by using the `request system software add /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz` command.

```
user@host> request system software add /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
NOTICE: Validating configuration against jinstall64-14.1R4.10-domestic-signed.tgz.
NOTICE: Use the 'no-validate' option to skip this if desired.
Checking compatibility with configuration
Initializing...
Using jbase-13.3R6.5
Verified manifest signed by PackageProductionEc_2015
Using /var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Using jinstall64-14.1R4.10-domestic.tgz
```

```
Using jbundle64-14.1R4.10-domestic.tgz
Checking jbundle requirements on /
Using jbase-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jbase-14.1R4.10 signed by PackageProductionEc_2015
Using /var/v/c/tmp/jbundle/jboot-14.1R4.10.tgz
Using jcrypto64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jcrypto64-14.1R4.10 signed by PackageProductionEc_2015
Using jdocs-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jdocs-14.1R4.10 signed by PackageProductionEc_2015
Using jkernel64-14.1R4.10.tgz
Using jpfe-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M10-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M120-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M160-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M320-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M40-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-M7i-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-T-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X2000-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-X960-14.1R4.10.tgz
Verified SHA1 checksum of jpfe-common-14.1R4.10.tgz
Using jplatform-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jplatform-14.1R4.10 signed by PackageProductionEc_2015
Using jroute-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jroute-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime-14.1R4.10 signed by PackageProductionEc_2015
Using jruntime64-14.1R4.10.tgz
Verified manifest signed by PackageProductionEc_2015
Verified jruntime64-14.1R4.10 signed by PackageProductionEc_2015
Using jservices-14.1R4.10.tgz
Using jservices-crypto-14.1R4.10.tgz
Hardware Database regeneration succeeded
Validating against /config/juniper.conf.gz
mgd: commit complete
Validation succeeded
Installing package '/var/tmp/jinstall64-14.1R4.10-domestic-signed.tgz' ...
```

```
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionEc_2015
Verified jinstall64-14.1R4.10-domestic.tgz signed by PackageProductionRSA_2015
Adding jinstall64...
Verified manifest signed by PackageProductionEc_2015

WARNING:     This package will load JUNOS 14.1R4.10 software.
WARNING:     It will save JUNOS configuration files, and SSH keys
WARNING:     (if configured), but erase all other files and information
WARNING:     stored on this machine.  It will attempt to preserve dumps
WARNING:     and log files, but this can not be guaranteed.  This is the
WARNING:     pre-installation stage and all the software is loaded when
WARNING:     you reboot the system.

Saving the config files ...
NOTICE: uncommitted changes have been saved in /var/db/config/juniper.conf.pre-install
Installing the bootstrap installer ...

WARNING:     A REBOOT IS REQUIRED TO LOAD THIS SOFTWARE CORRECTLY. Use the
WARNING:     'request system reboot' command when software installation is
WARNING:     complete. To abort the installation, do not reboot your system,
WARNING:     instead use the 'request system software delete jinstall'
WARNING:     command as soon as this operation completes.

Saving package file in /var/sw/pkg/jinstall64-14.1R4.10-domestic-signed.tgz ...
Saving state for rollback ...
```

9. Reboot `re0` by using the `request system reboot` command.

```
user@host> request system reboot
Reboot the system ? [yes,no] (no) yes


*** FINAL System shutdown message from user@host ***

System going down IMMEDIATELY


Shutdown NOW!
[pid 22857]

user@host> Connection closed by foreign host.
```

If you are not on the console port, you are disconnected from the router session.

10. After waiting a few minutes, log in to the device again.

    You are logged in to the backup Routing Engine (`re0`).

11. Verify that both Routing Engines are running the new software image by using the `show version` command.

```
{backup}
user@host> show version invoke-on all-routing-engines
Hostname: host
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]

re1:
--------------------------------------------------------------------------
Hostname: host
Model: mx480
Junos: 14.1R4.10
JUNOS Base OS boot [14.1R4.10]
JUNOS Base OS Software Suite [14.1R4.10]
JUNOS Packet Forwarding Engine Support (M/T/EX Common) [14.1R4.10]
JUNOS Packet Forwarding Engine Support (MX Common) [14.1R4.10]
JUNOS platform Software Suite [14.1R4.10]
JUNOS Runtime Software Suite [14.1R4.10]
JUNOS Online Documentation [14.1R4.10]
```

12. If you want to, you can optionally display the unified ISSU log messages by using the `show log messages` command.

13. If you want to, make `re0` the primary Routing Engine by using the `request chassis routing-engine master acquire` command.

```
{backup}
user@host> request chassis routing-engine master
acquire
Attempt to become the master routing engine ? [yes,no] (no) yes

Resolving mastership...
Complete. The local routing engine becomes the master.

user@host>
```

Table 31 on page 537 shows the Routing Engine status after the unified ISSU, after rebooting the backup Routing Engine, and after switching primary role.

Table 31: Routing Engine Status After Upgrading, Manually Rebooting, and Switching Primary Role

RE0	RE1
Primary	Backup
New software version installed	New software version installed
New software version running	New software version running

14. Enable GRES and NSR again by performing the steps in "Verifying Dual Routing Engines and Enabling GRES and NSR" on page 504.

15. Perform the applicable steps in "Restoring Feature-Specific Configuration" on page 517.

16. If you are satisfied with the results of your testing, you can optionally back up the system software to the device's hard disk by using the `request system snapshot` command on *each* Routing Engine.

NOTE: The root file system is backed up to **/altroot**, and **/config** is backed up to **/altconfig**. After you issue the `request system snapshot` command, you cannot easily return to the previous version of the software, because the device flash and hard

disks are identical. To return to the previous version of the software, you must boot the device from removable media.

```
{master}
user@host> request system snapshot
user@host> request routing-engine login re1
{backup}
user@host> request system snapshot
```

Performing an In-Service Software Upgrade (ISSU) with Non-Stop Routing

IN THIS SECTION

You can use an in-service software upgrade with non-stop routing to upgrade the software running on the switch with minimal traffic disruption during the upgrade.

NOTE: Starting with Junos OS Release 18.2R1 on the QFX5200 switch, we recommend that you wait at least five minutes between in-service software upgrades.

NOTE: Starting with Junos OS Release 17.1R1, on QFX5100 and EX4600 switches, you cannot perform an ISSU from a Junos OS Release earlier than 17.1R1 to Junos OS Release 17.1R1.

This topic covers:

Preparing the Switch for Software Installation

Before you begin software installation using ISSU:

NOTE: Before you perform an in-service software upgrade, if applicable, remove the `set system internet-options no-tcp-reset drop-all-tcp` command from the configuration, otherwise the upgrade will fail and an error message will be displayed.

NSB and non-stop routing enable NSB-supported Layer 2 protocols to synchronize protocol information between the primary and backup Routing Engines.

- Enable non-stop routing. See "Configuring Nonstop Active Routing" on page 280for information on how to enable it.
- Enable nonstop bridging (NSB). See "Configuring Nonstop Bridging" on page 258 for information on how to enable it.
- Configure the Bidirectional Forwarding Detection Protocol (BFD) timeout to be more than one second, otherwise you will receive an error.

Upgrading the Software Using ISSU

This procedure describes how to upgrade the software running on a standalone switch:

NOTE: If the Host OS software needs to be updated, you cannot perform an ISSU. Instead, perform a standard software upgrade.

To upgrade the switch using ISSU:

1. Download the software package by following the procedure in the Downloading Software Files with a Browser section in *Installing Software Packages on QFX Series Devices*.
2. Copy the software package or packages to the switch. We recommend that you copy the file to the `/var/tmp` directory.
3. Log in to the console connection. Using a console connection allows you to monitor the progress of the upgrade.
4. Start the ISSU:
   - On the switch, enter:

     ```
     user@switch> request system software in-service-upgrade /var/tmp/package-name.tgz
     ```

     where *package-name*`.tgz` is, for example, `jinstall-host-qfx-5e-18.1R1-secured-signed.tgz`.

NOTE: During the upgrade, you will not be able to access the Junos OS CLI.

The switch displays status messages similar to the following messages as the upgrade executes:

```
ISSU: Validating Image

 PRE ISSU CHECK:
 ---------------
 PFE Status                          : Online
 Member Id zero                      : Valid
 VC not in mixed or fabric mode      : Valid
 Member is single node vc            : Valid
 BFD minimum-interval check done     : Valid
 GRES enabled                        : Valid
 GR enabled                          : Valid
 drop-all-tcp not configured         : Valid
 Ready for ISSU                      : Valid

warning: Do NOT use /user during ISSU. Changes to /user during ISSU may get lost!


Pushing Junos image package to the host...
Installing /var/tmp/install-media-qfx-5e-junos-2018-secure.tgz
Extracting the package ...
total 1110328
-rw-r--r-- 1 18735 758 237044439 Oct 26 05:11 jinstall-qfx-5e-junos-2018-secure-linux.tgz
-rw-r--r-- 1 18735 758 899918118 Oct 26 05:11 jinstall-qfx-5e-junos-2018-secure-app.tgz

============================================
Current Host kernel version : 3.14.52-rt50-WR7.0.0.9_ovp
Package Host kernel version : 3.14.52-rt50-WR7.0.0.9_ovp
Current Host version        : 3.0.7
Package Host version        : 3.0.7
Min host version required for applications: 3.0.7
Min host version required for in-service-upgrade: 3.0.7
============================================

Setting up Junos host applications for in-service-upgrade ...
------------------------------------------------
Running Junos application installer for in-service-upgrade
------------------------------------------------
```

```
------------------------------
Installing /var/sw/applications/qfx-5e-flex-2018.tgz
------------------------------
pkg_install_rpms: qfx-5e-base-1.0-0-2018.x86_64.rpm
Installing qfx-5e-control-plane-flex-1.0-0-2018.x86_64.rpm ...
========================================
Loading cache...
Updating cache...               ######################################## [100%]

Committing transaction...
Preparing...                    ######################################## [  0%]
   1:Installing qfx-5e-contro.. ######################################## [100%]

Output from qfx-5e-control-plane-flex-1.0-0@x86_64:
------------------------------
Installing JUNOS image: jinstall-jcp-i386-flex-18.12018.img.gz
------------------------------
Extracting jinstall-jcp-i386-flex-18.12018.img.gz to /recovery/junos/jinstall-jcp-i386-
flex-18.12018-2018.img
Prepare host for virtfs...
Integrity check passed for hash-control-plane.md5.


Installing packages (1):
  qfx-5e-control-plane-flex-1.0-0@x86_64

812.9MB of package files are needed. 821.5MB will be used.


Saving cache...

========================================
Application installed.
Waiting to sync newly setup VM disk
VM ready after 200 seconds
[Oct 26 05:19:22]:ISSU: Preparing Backup RE
Prepare for ISSU
[Oct 26 05:19:27]:ISSU: Backup RE Prepare Done
Spawning the backup RE
Spawn backup RE, index 0 successful
Starting secondary dataplane
Second dataplane container started
```

```
GRES in progress
Waiting for backup RE switchover ready
GRES operational
Copying home directories
Copying home directories successful
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
[Oct 26 05:28:33]:ISSU: Preparing Daemons
[Oct 26 05:28:39]:ISSU: Daemons Ready for ISSU
[Oct 26 05:28:43]:ISSU: Starting Upgrade for FRUs
[Oct 26 05:28:54]:ISSU: FPC Warm Booting
[Oct 26 05:29:59]:ISSU: FPC Warm Booted
[Oct 26 05:30:10]:ISSU: Preparing for Switchover
[Oct 26 05:30:14]:ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
Send ISSU done to chassisd on backup RE
Chassis ISSU Completed
Removing dcpfe1 eth1 128.0.0.16 IP
Bringing down bme01
Post Chassis ISSU processing done
[Oct 26 05:30:17]:ISSU: IDLE
Stopping primary dataplane
Clearing ISSU states
Console and management sessions will be disconnected. Please login again.
```

NOTE: If the ISSU process stops, you can look at the CLI output when you issue the `request system software in-service-upgrade` command to diagnose the problem. You can also look at syslog files for more information.

5. Log in after the reboot of the switch completes. To verify that the software has been upgraded, enter the following command:

```
user@switch> show version
```

Performing an In-Service Software Upgrade (ISSU) in ACX5000 Series Routers

IN THIS SECTION

You can use an in-service software upgrade to upgrade the software running on the router with minimal traffic disruption during the upgrade.

NOTE: ISSU is supported in Junos OS Release 15.1X54–D60 and later on ACX5000 Series routers.

This topic covers:

Preparing the Router for Software Installation

Before you begin software installation using ISSU:

NOTE: Before you perform an in-service software upgrade, if applicable, remove the `set system internet-options no-tcp-reset drop-all-tcp` command from the configuration, otherwise the upgrade will fail and an error message will be displayed.

- Ensure that nonstop active routing (NSR) and nonstop bridging (NSB) are enabled. If enabled, disable graceful restart (GR), because NSR and GR cannot be enabled simultaneously. NSB and GR enable NSB-supported Layer 2 protocols to synchronize protocol information between the primary and backup Routing Engines.
- If NSR is not enabled (`Stateful Replication` is `Disabled`), then enable NSR. NSR requires you to configure graceful Routing Engine switchover (GRES). By default, NSR is disabled.
  - To enable graceful Routing Engine switchover, include the `graceful-switchover` statement at the [`edit chassis redundancy`] hierarchy level as user@host#**set chassis redundancy graceful-switchover**.
  - To enable NSR, include the `nonstop-routing` statement at the [`edit routing-options`] hierarchy level as user@host#**set routing-options nonstop-routing**.

- Enable nonstop bridging (NSB). Nonstop bridging requires you to configure graceful Routing Engine switchover (GRES). By default, NSB is disabled.
  - To enable graceful Routing Engine switchover, include the `graceful-switchover` statement at the [`edit chassis redundancy`] hierarchy level as user@host#**set chassis redundancy graceful-switchover**.
  - To enable NSB, include the `nonstop-bridging` statement at the [`edit protocols layer2-control`] hierarchy level as user@host#**set protocols layer2-control nonstop-bridging**.
- (Optional) Back up the system software—Junos OS, the active configuration, and log files—on the router to an external storage device with the `request system snapshot` command.

On ACX5000 line of routers, you need to consider the following feature before performing ISSU:

- ISSU supports link fault management (LFM) timeout sessions of 1 second interval. During ISSU, you may notice LFM flaps for sessions having timeout interval of less than 1 second.
- Bidirectional Forwarding Detection (BFD) sessions having timeout interval of less than 1 second need to be reconfigured to 1 second before starting the ISSU process. You can restore the timeout interval to its original value after completing the ISSU process.
- ISSU supports interval slow (every 30 seconds) for periodic transmission of Link Aggregation Control Protocol (LACP) packets.
- ISSU supports Virtual Router Redundancy Protocol (VRRP) version 3.

ISSU do not support the following ACX5000 features:.

- Downgrade to an earlier version of Junos OS software. If you want to install an earlier version of Junos OS software, use the `request system software add` CLI command.
- Upgrade of Host OS software.
- Connectivity fault management (CFM).
- TWAMP, RPF, RFC2544, and clocksyncd daemon (timing functionality).
- Mirroring and pseudowire cross connect.
- IPv6 firewall, IPv6 COS (classification and rewrite), IPv6 VPN, and VPLS mesh group.
- Virtual Router Redundancy Protocol (VRRP) version 1 and 2.
- Interval fast (every second) for periodic transmission of Link Aggregation Control Protocol (LACP) packets. If the periodic interval fast is configured, then you may notice traffic drops because of LACP links going down during ISSU. ACX5000 line of routers can support LACP with fast hello by configuring the `fast-hello-issu` option (**user@host# set protocols lacp fast-hello-issu**) on the main router and peer routers before starting ISSU.

NOTE: The peer router must have Junos OS software to support this functionality.

Upgrading the Software Using ISSU

This procedure describes how to upgrade the software running on a standalone router:

NOTE: If the Host OS software needs to be updated, you cannot perform an ISSU. Instead, perform a standard software upgrade.

It is recommended to cleanup any unwanted data from the **/var** directory (**/var/log**, **/var/tmp**) before initiating the ISSU process.

To upgrade the router using ISSU:

1. Download the software package from the Juniper Networks Support website https://www.juniper.net/support/downloads/junos.html .

   NOTE: To access the download site, you must have a service contract with Juniper Networks and an access account. If you need help obtaining an account, complete the registration form at the Juniper Networks website https://www.juniper.net/registration/Register.jsp .

2. Go to ACX Series section and select the ACX5000 Series platform software you want to download.
3. Copy the software package or packages to the router. We recommend that you copy the file to the `/var/tmp` directory.
4. Log in to the console connection. Using a console connection allows you to monitor the progress of the upgrade.
5. Start the ISSU:

   - On the router, enter:

     ```
     user@host> request system software in-service-upgrade /var/tmp/package-name.tgz
     ```

     where *package-name*.tgz is, for example, `jinstall-acx5k-15.1X54-D60.9-domestic-signed.tgz`.

     NOTE: During the upgrade, you will not be able to access the Junos OS CLI.

The router displays status messages similar to the following messages as the upgrade executes:

```
PRE ISSU CHECK:
---------------
PFE Status                            : Online
BFD minimum-interval check done       : Valid
GRES enabled                          : Valid
NSR enabled                           : Valid
drop-all-tcp not configured           : Valid
OVSDB not configured                  : Valid

warning: Do NOT use /user during ISSU. Changes to /user during ISSU may get lost!
[Oct 24 00:25:37]:ISSU: Validating Image
[Oct 24 00:25:44]:ISSU: Preparing Backup RE
Prepare for ISSU
[Oct 24 00:25:49]:ISSU: Backup RE Prepare Done
Extracting jinstall-acx5k-15.1X54-D60.3-domestic ...
Install jinstall-acx5k-15.1X54-D60.3-domestic completed
Spawning the backup RE
Spawn backup RE, index 0 successful
GRES in progress
GRES done in 0 seconds
Waiting for backup RE switchover ready
GRES operational
Copying home directories
Copying home directories successful
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
[Oct 24 00:31:56]:ISSU: Preparing Daemons
[Oct 24 00:32:57]:ISSU: Daemons Ready for ISSU
[Oct 24 00:33:02]:ISSU: Starting Upgrade for FRUs
[Oct 24 00:33:23]:ISSU: FPC Warm Booting
[Oct 24 00:34:41]:ISSU: FPC Warm Booted
[Oct 24 00:34:51]:ISSU: Preparing for Switchover
[Oct 24 00:34:57]:ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
Send ISSU done to chassisd on backup RE
Chassis ISSU Completed
```

```
[Oct 24 00:35:18]:ISSU: IDLE
Console and management sessions will be disconnected. Please login again.
```

NOTE: An ISSU might stop instead of terminate if the FPC is at the warm boot stage. Also, any links that go down and up will not be detected during a warm boot of the Packet Forwarding Engine (PFE).

NOTE: If the ISSU process stops, you can look at the log files to diagnose the problem. The log files are located at `/var/log/vjunos-log.tgz`.

6. Log in after the router reboots. To verify that the software has been upgraded, enter the following command:

```
user@host> show version
```

7. Disable or delete the configuration done to enable the ISSU. This includes disabling nonstop active routing (NSR), nonstop bridging (NBR) and graceful Routing Engine (GRES).

Verifying a Unified ISSU

Verify the status of FPCs and their corresponding PICs after the most recent unified ISSU.

Issue the `show chassis in-service-upgrade` command on the primary Routing Engine.

```
user@host> show chassis in-service-upgrade
   Item           Status                  Reason
  FPC 0          Online
```

Display the unified ISSU process messages by using the `show log messages` command.

How to Use Unified ISSU with Enhanced Mode

IN THIS SECTION

Unified ISSU with Enhanced Mode Overview

Enhanced mode is an in-service software upgrade (ISSU) option available on MPC8E, MPC9E, MPC11E, and JNP10K-LC4802 line cards that eliminates packet loss during the unified ISSU process. This is achieved by taking advantage of new line card architecture improvements that make it possible to have a second copy of the Junos OS software running on the line card in standby mode ready to take over while software moves from an old image to a new one during unified ISSU. You can enable enhanced mode by adding the `enhanced-mode` option to the `request system software in-service-upgrade` CLI command.

Use this document to learn about unified ISSU with enhanced mode and how to use it.

Benefits of Unified ISSU with Enhanced Mode

Unified ISSU with enhanced mode provides the following benefits:

- Upgrades to a new software version with no loss of transit or host bound traffic
- Reduces packet loss to zero or several milliseconds depending on configuration and network conditions
- Allows software upgrades to be performed without the need for maintenance windows
- Uses the existing unified ISSU process and doesn't require any special configuration

Prerequisites for Performing Unified ISSU with Enhanced Mode

Before you begin a unified ISSU with enhanced mode, there are several prerequisites to keep in mind:

- The device running unified ISSU with enhanced mode must use an MPC8E, MPC9E, MPC11E, or JNP10K-LC4802 line card.

> **NOTE**: If you are performing unified ISSU with enhanced mode on a device that has a mix of supported and unsupported line cards, there will be sub-second traffic loss for traffic passing through the unsupported line cards.

> **NOTE**: If you are performing unified ISSU with enhanced mode on guest network functions (GNFs), then all GNFs should be using MPC8E, MPC9E, MPC11E, or JNP10K-LC4802 line cards to avoid traffic loss.

- The Linux version running on your Flexible PIC Concentrator (FPC) and the line card Linux version in the target release need to be compatible.
- Enhanced mode won't work if the target release carries changes that require the ASIC blocks to be reset.
- Forwarding memory usage should be below 75 percent to ensure no packet loss during the unified ISSU process

> **NOTE**: Unified ISSU with enhanced mode will still work if forwarding memory usage is above 75 percent, but it might introduce several milliseconds of packet loss.

- All prerequisites for unified ISSU also apply to enhanced mode. See Unified ISSU System Requirements for more information.

You can check to see if your device can use unified ISSU with enhanced mode to upgrade to a specific release by using the `request system software validate in-service-upgrade` *`package-name`*`.tgz enhanced-mode` command. If your device and the target release are not compatible with enhanced mode, you can still use regular unified ISSU to upgrade with minimal disruption of traffic.

Performing Unified ISSU with Enhanced Mode

To perform a unified ISSU with enhanced mode, follow these steps:

1. Download the software package by following the procedure in Downloading Software.
2. Copy the software package or packages to the device. We recommend that you copy the file to the `/var/tmp` directory.
3. Log in to the console connection. Using a console connection allows you to monitor the progress of the upgrade.
4. Verify that you can use unified ISSU with enhanced mode for your desired release.

a. On the device, enter:

```
user@host> request system software validate in-service-upgrade /var/tmp/package-name.tgz enhanced-
mode
```

where *package-name*.tgz is the name of the software package you downloaded in Step 1.

5. Start the unified ISSU with enhanced mode:

a. On the device, enter:

```
user@host> request system software in-service-upgrade /var/tmp/package-name.tgz enhanced-mode
reboot
```

where *package-name*.tgz is the name of the software package you downloaded in Step 1.

NOTE: During the upgrade, you will not be able to access the Junos OS CLI.

The device displays status messages similar to the following messages as the upgrade executes:

```
Chassis ISSU enhanced-mode
ISSU: set chassis enhanced-mode
Chassis ISSU Check Done
ISSU: Validating Image
..
mgd: commit complete
Validation succeeded
 Validating Image Done
 Preparing Backup RE
 Pushing /var/tmp/junos-install-mx-x86-32-20.1.tgz to re1:/var/tmp/junos-install-mx-
x86-32-20.1.tgz
 Pushing package /var/tmp/junos-install-mx-x86-32-20.1.tgz to re1 done
 Installing package /var/tmp/junos-install-mx-x86-32-20.1.tgz on re1
…
Verified sflow-mx signed by PackageDevelopmentEc_2019 method ECDSA256+SHA256
NOTICE: 'pending' set will be activated at next reboot...
ISSU: Installing package /var/tmp/junos-install-mx-x86-32-20.1.tgz on re1 done
ISSU: Rebooting Backup RE

Rebooting re1
```

```
 Backup RE Prepare Done
 Waiting for Backup RE reboot
 Backup RE reboot done. Backup RE is up
 Waiting for Backup RE state synchronization
 Backup RE state synchronization done
 GRES operational
 "Initiating Chassis In-Service-Upgrade"
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Offline Incompatible FRUs
ISSU: Starting Upgrade for FRUs
…

ISSU: Preparing for Switchover
ISSU: Ready for Switchover
 Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 1          Online (ISSU)
  FPC 2          Offline                 Configured power off
Resolving mastership...
Complete. The other routing engine becomes the master.
```

NOTE: If the unified ISSU process stops, you can look at the CLI output by using the `request system software in-service-upgrade` command to diagnose the problem. You can also look at syslog files for more information.

6. Log in after the reboot of the device is completed. To verify that the software has been upgraded, enter the following command:

```
user@host> show version
```

NOTE: When using unified ISSU with enhanced mode, the base Linux OS on your FPC cannot be upgraded as part of the ISSU process. Linux can be updated with an upgrade done through regular unified ISSU or a reboot of the FPC.

Verifying a Unified ISSU

IN THIS SECTION

Purpose

Verify the status of FPCs and their corresponding PICs after the most recent unified ISSU.

Action

Issue the `show chassis in-service-upgrade` command on the primary Routing Engine.

```
user@host> show chassis in-service-upgrade
  Item            Status                  Reason
  FPC 0           Online
  FPC 1           Online
  FPC 2           Online
    PIC 0         Online
    PIC 1         Online
  FPC 3           Online
  FPC 4           Online
    PIC 1         Online
  FPC 5           Online
    PIC 0         Online
  FPC 6           Online
    PIC 3         Online
  FPC 7           Online
```

Display the unified ISSU process messages by using the `show log messages` command.

Meaning

See *show chassis in-service-upgrade* for more information.

Troubleshooting Unified ISSU Problems

If the unified ISSU procedure stops progressing:

1. Open a new session on the primary Routing Engine and issue the `request system software abort in-service-upgrade` command.
2. Check the existing router session to verify that the upgrade has been terminated.

   An "ISSU: terminated!" message is provided. Additional system messages provide you with information about where the upgrade stopped and recommendations for the next step to take.

See *request chassis cluster in-service-upgrade abort (ISSU)* for more information.

Managing and Tracing BFD Sessions During Unified ISSU Procedures

Bidirectional Forwarding Detection (BFD) sessions temporarily increase their detection and transmission timers during unified ISSU procedures. After the upgrade, these timers revert to the values in use before the unified ISSU started. The BFD process replicates the unified ISSU state and timer values to the backup Routing Engine for each session.

No additional configuration is necessary to enable unified ISSU for BFD. However, you can disable the BFD timer negotiation during the unified ISSU by including the `no-issu-timer-negotiation` statement at the `[edit protocols bfd]` hierarchy level.

```
[edit protocols bfd]
no-issu-timer-negotiation;
```

If you include this statement, the BFD timers maintain their original values during unified ISSU.

CAUTION: The BFD sessions might flap during unified ISSU or Routing Engine switchover, depending on the detection intervals.

For more information about BFD, see the Junos OS Routing Protocols Library.

To configure unified ISSU trace options for BFD sessions, include the `issu` statement at the `[edit protocols bfd traceoptions flag]` hierarchy level.

```
[edit protocols]
bfd {
```

```
    traceoptions {
        flag issu;
    }
}
```

Change History Table

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

Release	Description
18.1R1	Starting with Junos OS Release 18.2R1 on the QFX5200 switch, we recommend that you wait at least five minutes between in-service software upgrades.
17.1R1	Starting with Junos OS Release 17.1R1, on QFX5100 and EX4600 switches, you cannot perform an ISSU from a Junos OS Release earlier than 17.1R1 to Junos OS Release 17.1R1.

Performing an In-Service Software Reboot

SUMMARY

You can perform an in-service software reboot (ISSR) by following these steps.

NOTE: We recommend that you wait at least five minutes between in-service software reboots.

When you request an in-service software reboot (ISSR) on a standalone device:

1. The management process (MGD) verifies that graceful restart (GR) or non-stop routing and graceful Routing Engine switchover (GRES) are enabled.
2. The ISSU state machine spawns the backup Routing Engine (RE) with the existing software version.
3. The ISSU state machine checks to see if the backup RE has synchronized all of the data with the primary RE.

4. The ISSU state machine requests the routing protocol process (RPD) to notify its readiness for switchover.
5. RPD initiates the GR or non-stop routing procedures by notifying all of the registered protocols.
6. RPD notifies the ISSU state machine that its ready for switchover.
7. The primary role is switched between the REs, so the backup RE becomes the primary RE.
8. The old primary RE is shut down.
9. RPD is spawned on the new primary and continues the GR or non-stop routing procedure and exits either GR or non-stop routing after the protocol state synchronizes.

To perform an ISSR:

Issue the `request system reboot in-service` command.

For example:

```
user@switch> request system reboot in-service
Reboot the system ? [yes,no]
[Feb 22 02:37:04]:ISSU: Validating Image

 PRE ISSR CHECK:
 ---------------
 PFE Status                          : Online
 Member Id zero                      : Valid
 VC not in mixed or fabric mode      : Valid
 Member is single node vc            : Valid
 BFD minimum-interval check done     : Valid
 GRES enabled                        : Valid
 NSR enabled                         : Valid
 drop-all-tcp not configured         : Valid
 Ready for ISSR                      : Valid

warning: Do NOT use /user during ISSR. Changes to /user during ISSR may get lost!
Current image is jinstall-jcp-i386-flex-18.1.img
[Feb 22 02:37:14]:ISSU: Preparing Backup RE
Prepare for ISSR
[Feb 22 02:37:19]:ISSU: Backup RE Prepare Done
Spawning the backup RE
Spawn backup RE, index 1 successful
Starting secondary dataplane
Second dataplane container started
```

```
GRES in progress
Waiting for backup RE switchover ready
GRES operational
Copying home directories
Copying home directories successful
Initiating Chassis In-Service-Upgrade for ISSR
Chassis ISSU Started
[Feb 22 02:42:55]:ISSU: Preparing Daemons
[Feb 22 02:43:00]:ISSU: Daemons Ready for ISSU
[Feb 22 02:43:05]:ISSU: Starting Upgrade for FRUs
[Feb 22 02:43:15]:ISSU: FPC Warm Booting
[Feb 22 02:44:16]:ISSU: FPC Warm Booted
[Feb 22 02:44:27]:ISSU: Preparing for Switchover
[Feb 22 02:44:31]:ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
Send ISSR done to chassisd on backup RE
Chassis ISSU Completed
Removing dcpfe0 eth1 128.168.0.16 IP
Bringing down bme00
Post Chassis ISSU processing done
[Feb 22 02:44:33]:ISSU: IDLE
Stopping primary dataplane
Clearing ISSU states
Console and management sessions will be disconnected. Please login again.
device_handoff successful ret: 0
Shutdown NOW!
[pid 14305]

*** FINAL System shutdown message from root@sw-duckhorn-01 ***

System going down IMMEDIATELY
```

RELATED DOCUMENTATION

request system reboot

14

PART

Performing Nonstop Software Upgrade (NSSU)

Understanding Nonstop Software Upgrade on EX Series Switches

SUMMARY

Nonstop software upgrade (NSSU) is a feature that enables the upgrade of all supported EX Series switches in a network with a single command.

IN THIS SECTION

Nonstop software upgrade (NSSU) enables you to upgrade the software running on Juniper Networks EX Series Ethernet Switches with redundant Routing Engines and all member switches in EX Series Virtual Chassis using a single command. During the upgrade there might be minimal network traffic disruption during primary-role switchover, and the extent of disruption could be dependent on the network topology, configuration, network traffic, and other environment factors .

Performing an NSSU provides these benefits:

- No disruption to the control plane—An NSSU takes advantage of *graceful Routing Engine switchover* (GRES) and *nonstop active routing* (NSR) to ensure no disruption to the control plane. During the upgrade process, interface, kernel, and routing protocol information is preserved.
- Minimal disruption to network traffic—An NSSU minimizes network traffic disruption by upgrading member switches one at a time in other EX Series Virtual Chassis while permitting traffic to continue to flow through the members that are not being upgraded.

  To achieve minimal disruption to traffic, you must configure link aggregation groups (LAGs) such that the member links of each LAG reside on different line cards or Virtual Chassis members. When one member link of a LAG is down, the remaining links are up, and traffic continues to flow through the LAG.

NOTE: Because NSSU upgrades the software on each line card or on each Virtual Chassis member one at a time, an upgrade using NSSU can take longer than an upgrade using the `request system software add` command.

Requirements for Performing an NSSU

The following requirements apply to all switches and Virtual Chassis:

NOTE: NSSU can only upgrade up to three major releases ahead of the current release on a device. To upgrade to a release more than three releases ahead of the current release on a device, use the NSSU process to upgrade the switch to one or more intermediate releases until the switch is within three major releases of the target release.

- All Virtual Chassis members and all Routing Engines must be running the same Junos OS release.
- Graceful Routing Engine switchover (GRES) must be enabled.
- Nonstop active routing (NSR) must be enabled.

  NOTE: Although nonstop bridging (NSB) does not have to be enabled to perform an NSSU, we recommend enabling NSB before performing an NSSU. Enabling NSB ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover that is part of the NSSU.

- For minimal traffic disruption, you must define link aggregation groups (LAGs) such that the member links reside on different Virtual Chassis members or on different line cards.

  NOTE: During an NSSU operation, if you try to view LAG interface status on the primary Routing Engine member using the `show interfaces ae-`*`ae-interface-number`* CLI command, you might see incorrect or zero traffic counts. To work around this problem, run the command on the backup Routing Engine member instead if that member is already loaded and running.

The following are requirements for performing NSSU on an EX Series Virtual Chassis:

- The Virtual Chassis members must be connected in a ring topology so that no member is isolated as a result of another member being rebooted. This topology prevents the Virtual Chassis from splitting during an NSSU.
- The Virtual Chassis primary and backup must be adjacent to each other in the ring topology. Adjacency permits the primary and backup to always be in sync, even when the switches in linecard roles are rebooting.
- The Virtual Chassis must be preprovisioned so that the linecard role has been explicitly assigned to member switches acting in a linecard role. During an NSSU, the Virtual Chassis members must maintain their roles—the primary and backup must maintain their primary and backup roles (although primary role will change), and the remaining switches must maintain their linecard roles.
- A two-member Virtual Chassis must have `no-split-detection` configured so that the Virtual Chassis does not split when an NSSU upgrades a member.

How an NSSU Works

This section describes what happens when you request an NSSU on EX Series switches and Virtual Chassis.

NSSU Limitations

You cannot use an NSSU to downgrade the software—that is, to install an earlier version of the software than is currently running on the switch. To install an earlier software version, use the `request system software add` command.

You cannot roll back to the previous software version after you perform an upgrade using NSSU. If you need to roll back to the previous software version, you can do so by rebooting from the alternate root partition if you have not already copied the new software version into the alternate root partition.

NSSU and Junos OS Release Support

A Virtual Chassis must be running a Junos OS release that supports NSSU before you can perform an NSSU. If a Virtual Chassis is running a software version that does not support NSSU, use the `request system software add` command.

Nonstop software upgrade (NSSU) lists the EX Series switches and Virtual Chassis that support NSSU and the Junos OS release at which they began supporting it.

Overview of NSSU Configuration and Operation

You must ensure that the configuration of the switch or Virtual Chassis meets the requirements described in "Requirements for Performing an NSSU" on page 559. NSSU requires no additional configuration.

You perform an NSSU by executing the `request system software nonstop-upgrade` command. For detailed instructions on how to perform an NSSU, see the topics in Related Documentation.

RELATED DOCUMENTATION

Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure) | **1176**

Configuring Nonstop Active Routing | **280**

Configuring Graceful Routing Engine Switchover in a Virtual Chassis

Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade

PART 15

Multinode High Availability

Multinode High Availability

SUMMARY

Learn about the Multinode High Availability solution and how you can use it in simple and reliable deployment models.

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 (MNHA), 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 Layer 2 (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, also allows to run two different version of Junos.

We support Multinode High Availability in active/backup mode and in active-active mode (with support of multiple services redundancy groups (SRGs). For the complete list of supported features and platforms, see Multinode High Availability in Feature Explorer.

Supported Features

MNHA infrastructure provides redundancy through two deployment models:

- **Two-node MNHA:** A pair of firewalls operating together for high availability.
- **Four-node MNHA:** Two pairs of firewalls distributed across domains for enhanced resilience.

SRX Series Firewalls with Multinode High Availability support the firewall and advanced security services—such as application security, Content Security, intrusion prevention system (IPS), firewall user authentication, NAT, ALG.

For the complete list of devices and features supported with Multinode High Availability, see Feature Explorer .

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

How Is Multinode High Availability Different from Chassis Cluster?

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

Multinode High Availability uses an encrypted logical interchassis link (ICL). The ICL connects the nodes over a routed path instead of a dedicated Layer 2 network. This routed path can use one or more revenue ports for best resiliency, it's even possible to dedicate its own routing instance to these ports and paths to ensure total isolation which maximizes the resiliency of the solution.

Figure 36 on page 566 and Figure 37 on page 567 and show two architectures.

Figure 36: Chassis Cluster Topology in a Layer 2 Network

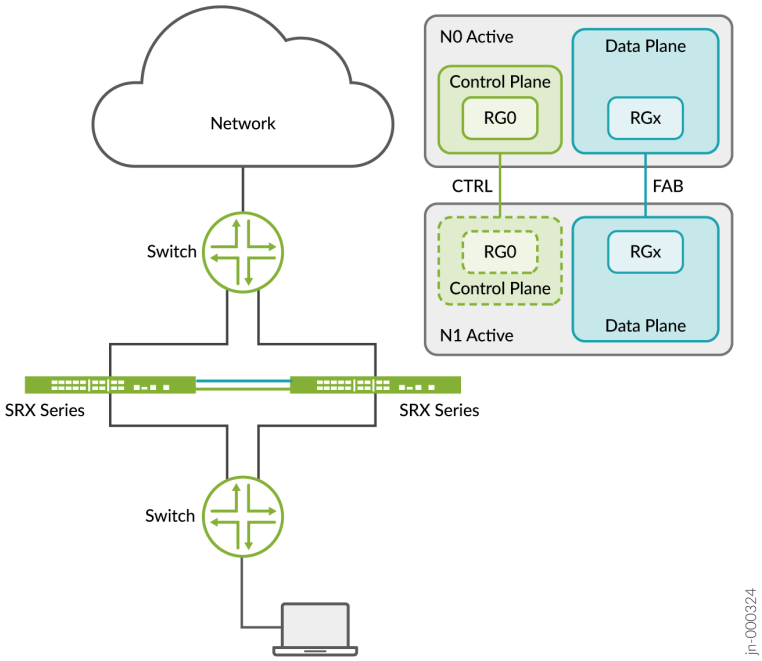


Figure 37: Multinode High Availability in a Layer 3 Network

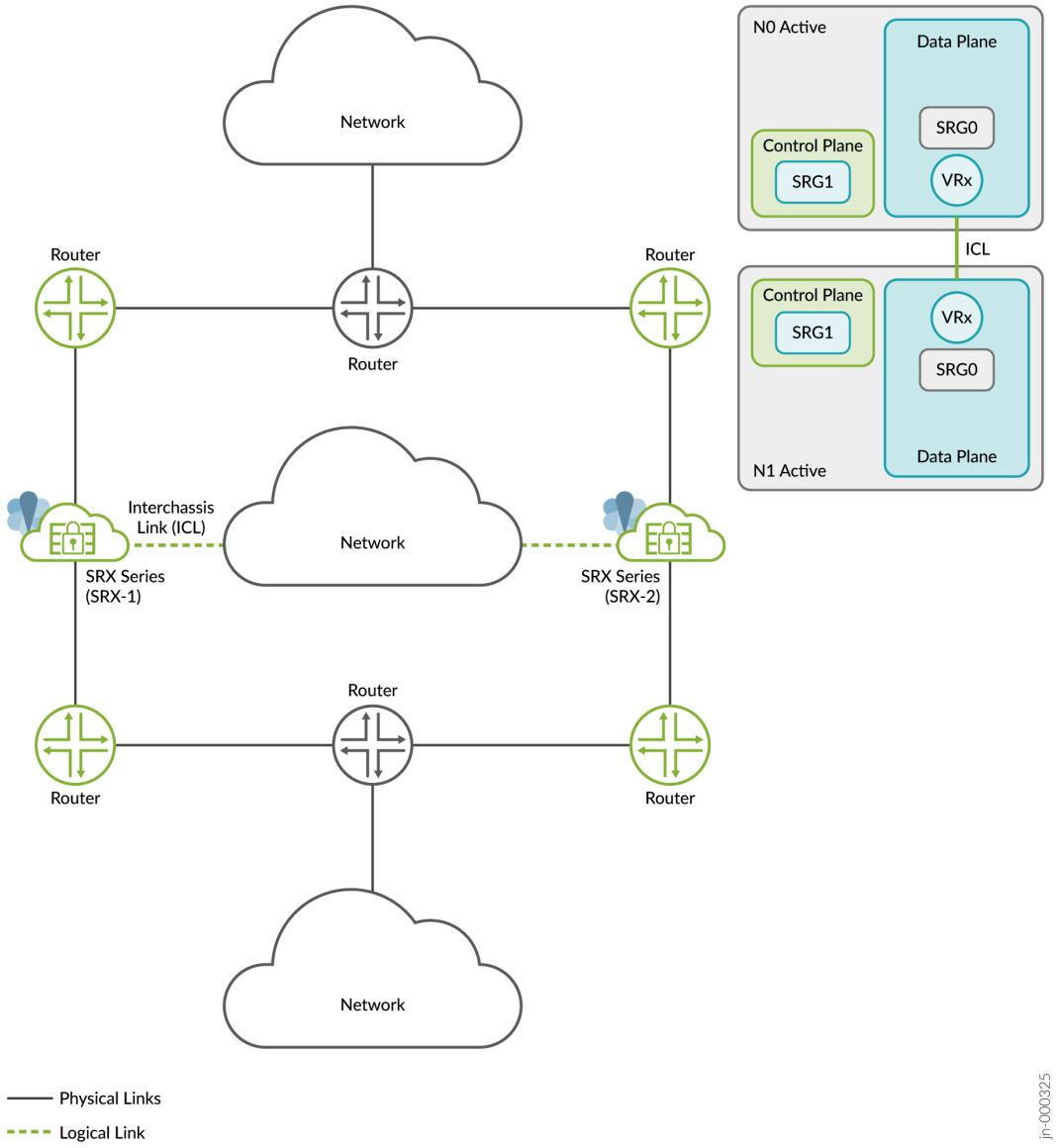


Table 32 on page 568 lists the differences between the two architectures

Table 32: Comparing Chassis Cluster and Multinode High Availability

Parameters	Chassis Cluster	Multinode High Availability
Network topology	Nodes connect to a broadcast domain	Nodes connect to a router, a broadcast domain, or a combination of both. • Nodes connect to a router • Broadcast domain • Combination of both above
Network environment	Layer 2	• Layer 3 (Route mode) • Layer 2 (defaut gateway mode • Combination of Layer 3 and Layer 2 (hybrid mode) • Public cloud (AWS) deployments
Traffic switchover approach	SRX Series Firewall sends GARP to the switch	Switchover using IP path selection by a peer Layer 3 router or Layer 2 GARP from an SRX Series Firewall to a peer Layer 2 switch • Route mode: Switchover using IP path selection (route updates) • Hybrid mode: Switchover using IP path selection (route updates), and sending GARP to the switch • Default gateway mode: SRX Series Firewall sends GARP to the switch
Public cloud	Not supported	Supported

Table 32: Comparing Chassis Cluster and Multinode High Availability *(Continued)*

Parameters	Chassis Cluster	Multinode High Availability
Dynamic routing function	Routing process active on the SRX Series where the control plane (RG0) is active	Routing process active on each SRX Series Firewall participating in Multinode High Availability
Connection between SRX Series Firewalls	• Control link (Layer 2 path) • Fabric link (Layer 2 path)	Interchassis link (Layer 3 path)
Connectivity / Geo-redundance	Requires a dedicated Layer 2 stretch between the SRX Series nodes for the control link and fabric link.	Uses any routed path between the nodes for the Interchassis link.
IP monitoring to detect network failure	• Interfaces • IP monitoring using IPv4 addresses	• Interfaces • IP monitoring using IPv4 and IPv6 addresses • Bidirectional Forwarding Detection (BFD) using IPv4 addresses and IPv6 addresses

Multinode High Availability Glossary

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

Table 33: Multinode High Availability Glossary

Term	Description
active/active state (SRG0)	All security services/flows are inspected at each node and backed up on the other node. Security flows must be symmetric.

Table 33: Multinode High Availability Glossary *(Continued)*

Term	Description
active/backup state (SRG1+)	SRG1+ remains active on one node at any given time and remains in backed up state on the other node. SRG1+ in the backup state is ready to take over traffic from the active SRG1 in case on a failure.
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	A failover happens when one node detects a failure (hardware/software and so on) and traffic transitions to the other node in a stateful manner. As result, the backup node in a high availability system takes over the task of the active node when the active node 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.
high availability/resiliency	Ability of a system to eliminate single points of failure to ensure continuous operations over an extended period of time.

Table 33: Multinode High Availability Glossary *(Continued)*

Term	Description
interchassis link	IP-based link (logical link) that connects nodes over a routed network in a Multinode High Availability deployment. The ICL link is normally bound to the loopback interfaces for most flexible deployments. Connectivity can be any routed or switched path as long as the connectivity is reachable between the two IP addresses. The security device uses the ICL to synchronize and maintain state information and to handle device failover scenarios.
Interchassis link encryption	Link encryption provides data privacy for messages traversing over the network. As the ICL link transmits private data, it is important to encrypt the link. You must encrypt the ICL using IPsec VPN.
monitoring (BFD)	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.
monitoring (IP)	Monitoring of a reliable IP address and system state in case of loss of communication with the peer node.
monitoring (path)	Method that uses ICMP to verify the reachability of the IP address. The default interval for ICMP ping probes is 1 second.
monitoring (system)	Monitoring of key hardware and software resources and infrastructures by triggering failover when a failure is detected on a node.
probing	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.

Table 33: 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 or virtual IPs in hybrid or default gateway mode.).
synchronization	Process where controls and data plane states are synchronized across the nodes.
virtual IP (VIP) address	Virtual IP addresses in hybrid or default gateway mode are used for activeness determination and enforcement on the switching side in a Multinode High Availability setup. The virtual IP is controlled by the SRG1+.
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.

RELATED DOCUMENTATION

Two-Node Multinode High Availability

SUMMARY

Learn about the two-node Multinode High Availability solution.

IN THIS SECTION

Two-node Multinode High Availability supports two SRX Series Firewalls presenting themselves as independent nodes to the rest of the network. The nodes are connected to adjacent infrastructure belonging to the same or different networks, all depending on the deployment mode. 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.

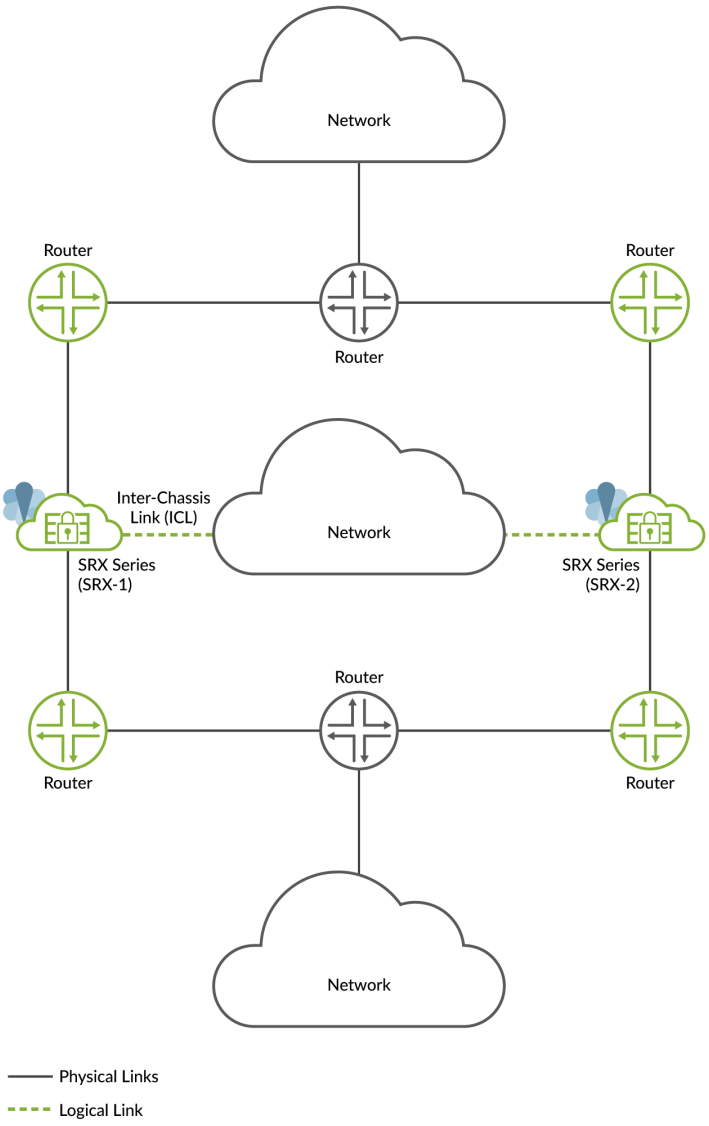
We support Multinode High Availability in active/backup mode and in active-active mode (with support of multiple services redundancy groups (SRGs). For the complete list of supported features and platforms, see Multinode High Availability in Feature Explorer.

Deployment Scenarios

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

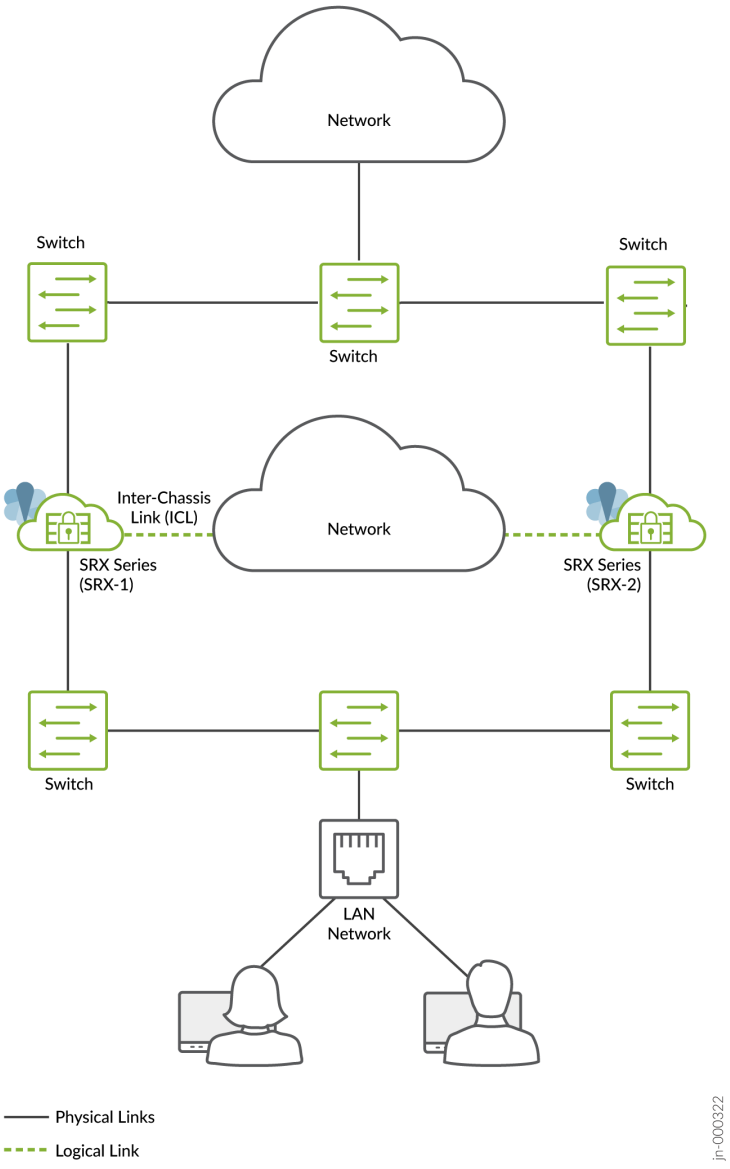
- Route mode (all interfaces connected using a Layer 3 topology)

Figure 38: Layer 3 Mode



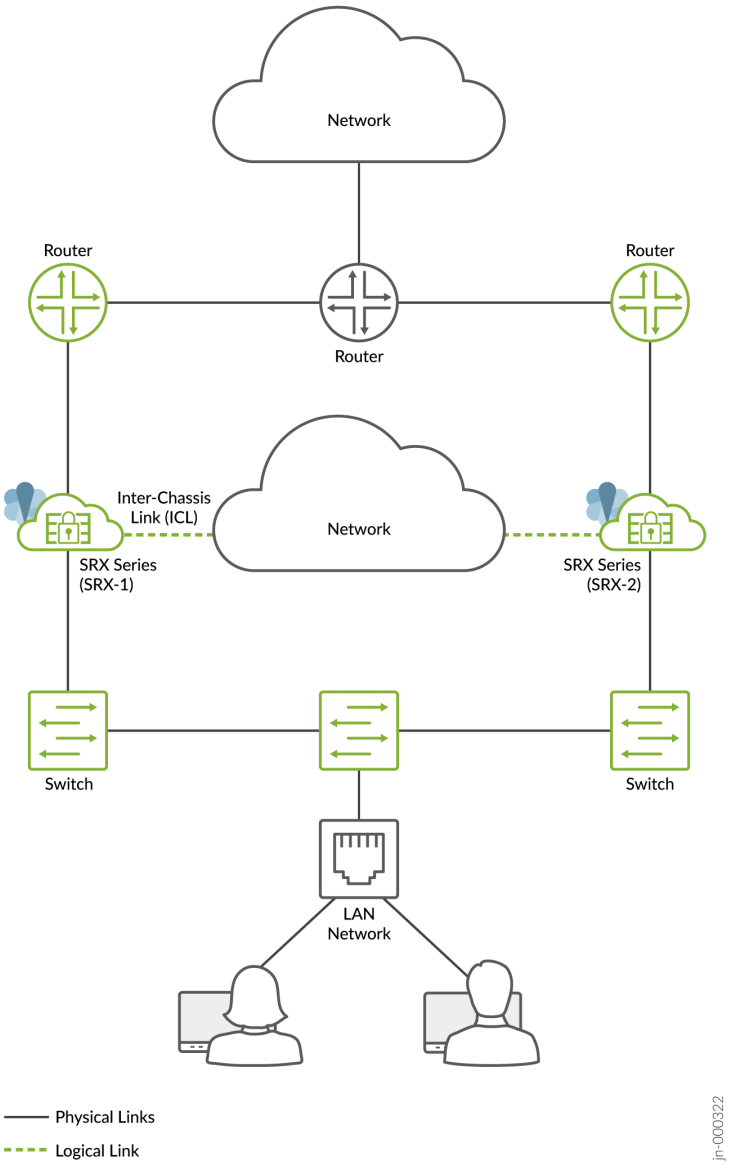
- Default gateway mode (all interfaces connected using an Layer 2 topology) used in more traditional environments. Common deployment of DMZ networks where the firewall devices act as the default gateway for the hosts and applications on the same segment.

Figure 39: Default Gateway Mode



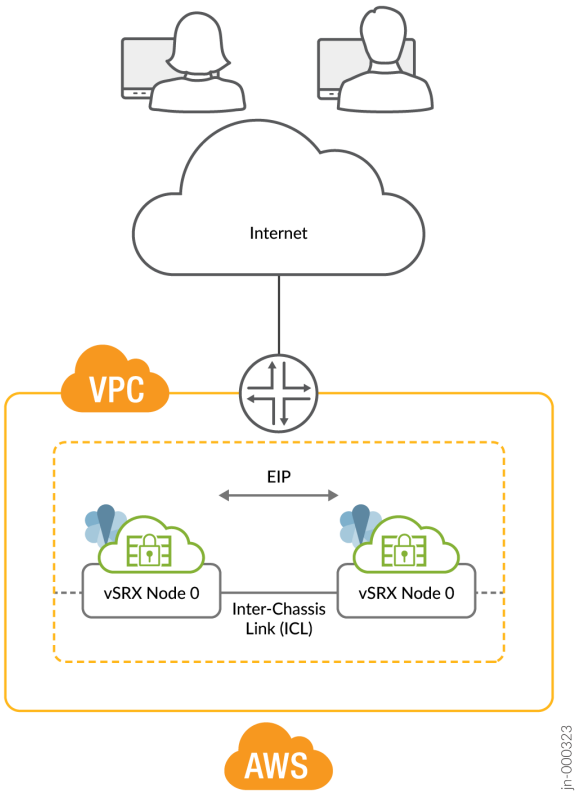
- Hybrid mode (one or more interfaces are connected using a Layer 3 topology and one or more interfaces are connected using a Layer 2 topology)

Figure 40: Hybrid Mode



- Public Cloud Deployment

Figure 41: Public Cloud Deployment (Example: AWS)



How Two-Node Multinode High Availability Works

IN THIS SECTION

NOTE: We support a two-node configuration for the Multinode High Availability solution.

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

The nodes communicate with each other using an interchassis link (ICL). The ICL link uses Layer 3 connectivity to communicate with each other, This communication can take place over a routed network (Layer 3), or a directly connected Layer 2 path. It is recommended to bind the ICL to the loopback interface and have more than one physical link (LAG/LACP) to ensure path diversity for the highest resiliency.

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

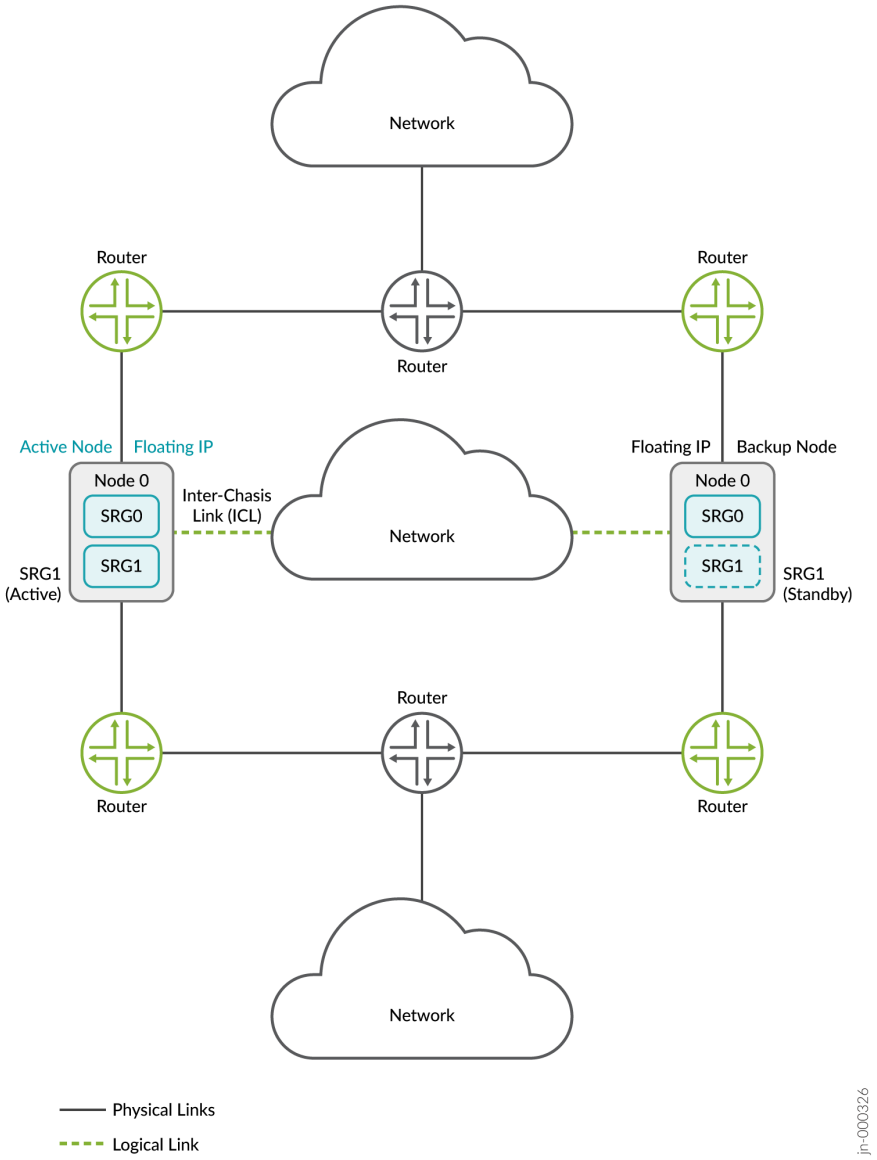
Multinode High Availability operates in:

- Active/active mode (SRG0) for the security services
- Active/backup mode (SRG1 and above) for security and system services

Floating IP addresses controlled by SRG1 or above moves between the nodes . Active SRG1+ hosts and controls the floating IP address. In failover scenarios, this IP address 'floats' to another active SRG1 based on configuration, system health, or path monitoring decisions. The the newly active SRG1+ can take on the function of a now-standby SRG1 and starts responding to incoming requests.

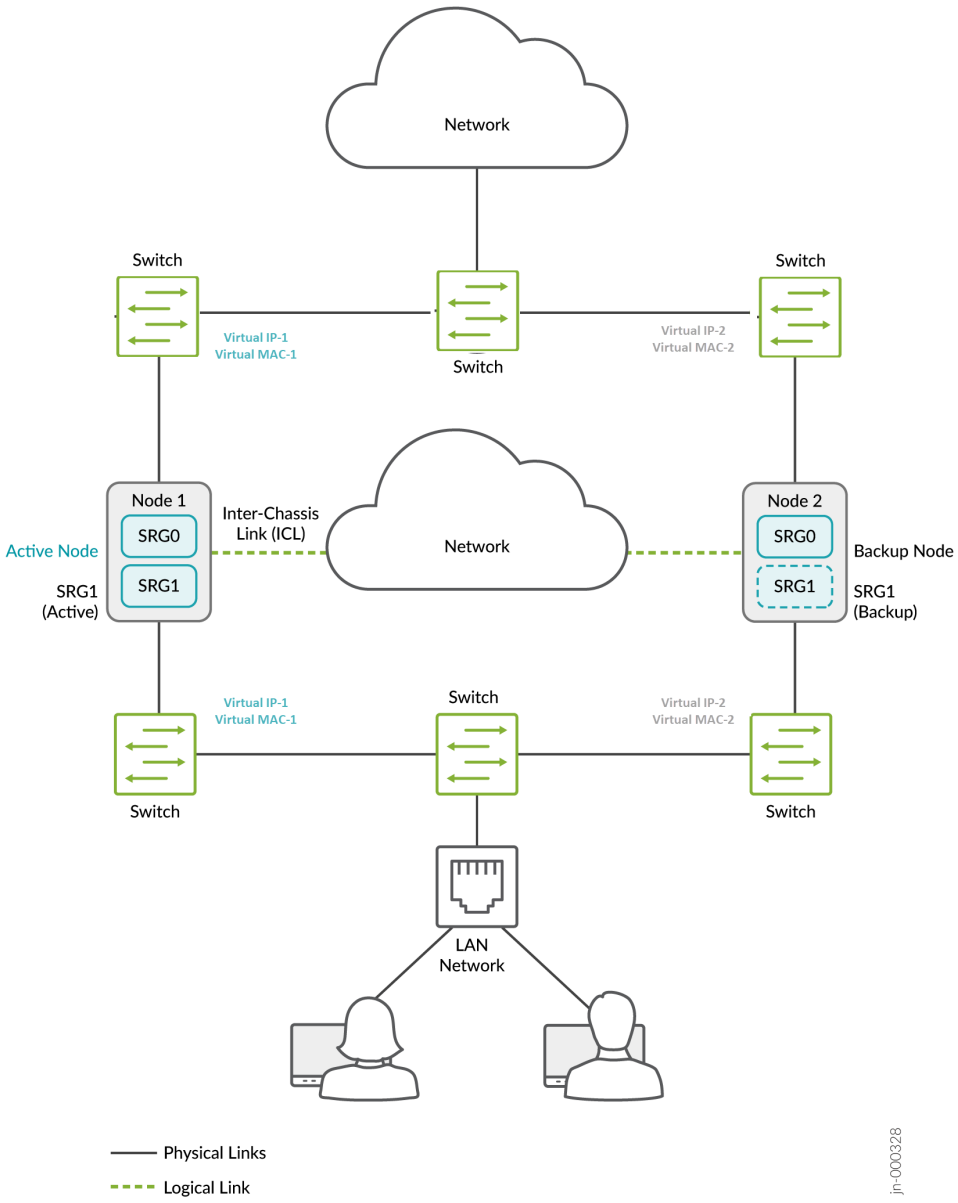
Figure 42 on page 579, Figure 43 on page 580, and Figure 44 on page 581 show deployments in Layer 3, hybrid, and default gateway modes.

Figure 42: Layer 3 Deployment



In this topology, two SRX Series Firewalls are part of a Multinode High Availability setup. The setup has Layer 3 connectivity between SRX Series Firewalls 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.

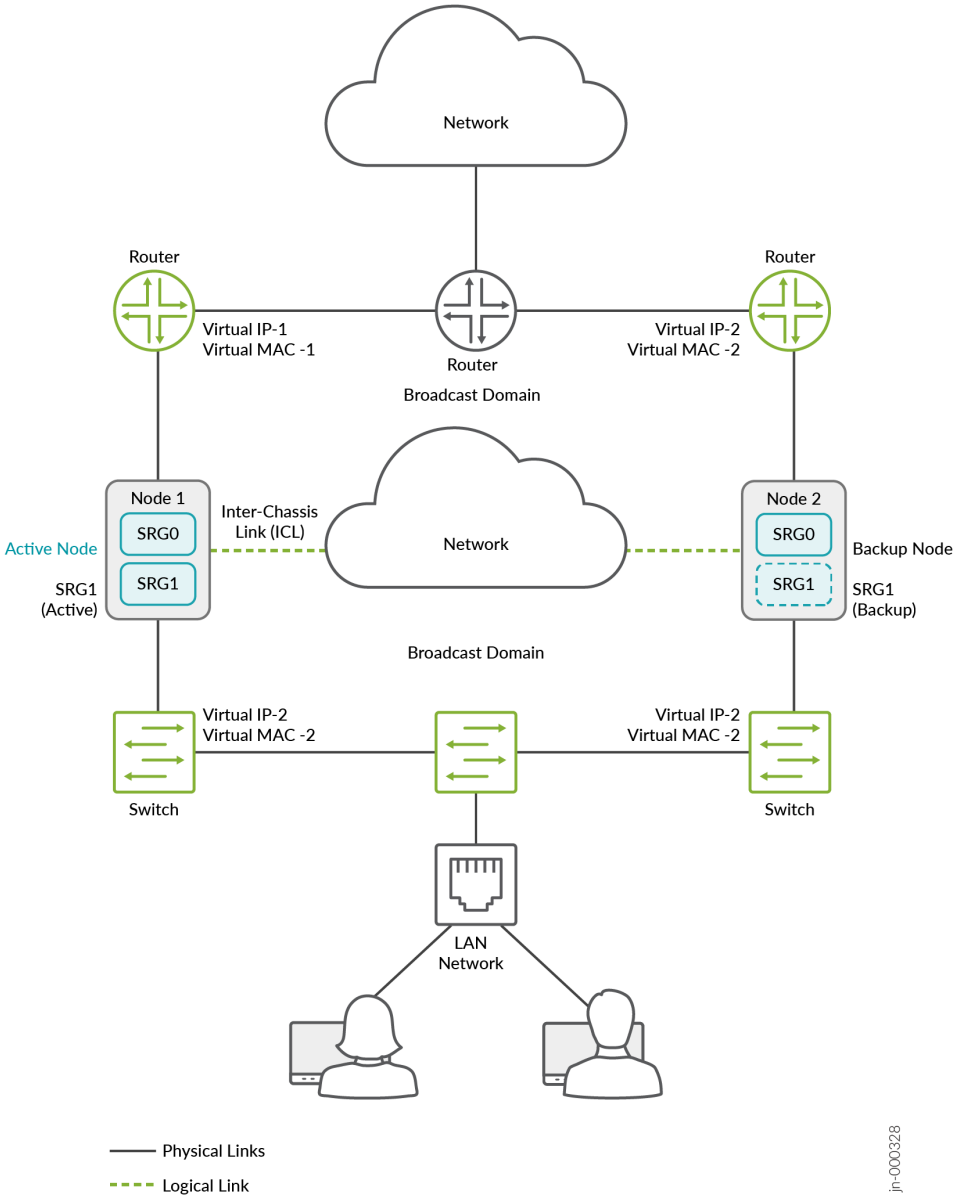
Figure 43: Default Gateway Deployment



In a typical default gateway deployment, hosts and servers in a LAN are configured with a default gateway of 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.

You must create static routes or dynamic routing on SRX Series Firewalls to reach other networks not directly connected.

Figure 44: Hybrid Deployment



In hybrid mode, an SRX Series Firewall uses a VIP address on the Layer 2 side to draw traffic toward it. You can optionally configure the static ARP for the VIP using the VMAC address 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. There are two types of SRGs:

- SRG0—Manages security service from Layer 4-Layer 7 except IPsec VPN services. The SRG0 operates in active mode on both nodes at any point in time. On SRG0, each security session must traverse the node in a symmetric flow, Backup of these flows are fully state-synchronized to the other node,
- SRG1+—Manages IPsec services and virtual IPs for hybrid and default gateway mode and are backed up to the other node. The SRG1 operates in active mode on one node and in backup node on another node.

Figure 45 on page 582 shows SRG0 and SRG1 in a Multinode High Availability setup.

Figure 45: Single SRG Support in Multinode High Availability (Active-Backup Mode)

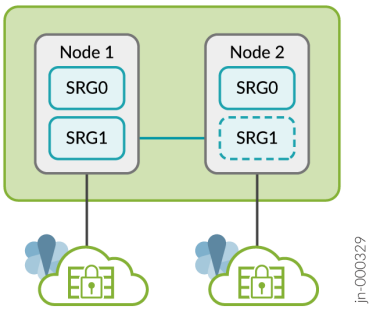
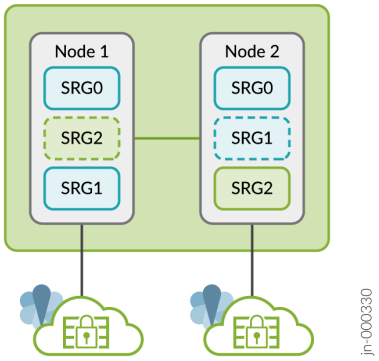


Figure 46 on page 583 shows SRG0 and SRG1+ in a Multinode High Availability setup.

Figure 46: Multi SRG Support in Multinode High Availability (Active-Active Mode)



Starting in Junos OS Release 22.4R1, you can configure Multinode High Availability to operate in active-active mode with support of multi SRG1s (SRG1+). In this mode, some SRGs remain active on one node and some SRGs remain active on another node. A particular SRG always operates in active-backup mode; it operates in active mode on one node and backup mode on another node. In this case, both the nodes can have the active SRG1 forwarding stateful services. Each node has a different set of floating IP addresses assigned to SRG1+.

NOTE: Starting in Junos OS Release 22.4R1, you can configure upto 20 SRGs in a Multinode Highavailability setup.

Table 34 on page 583 explains the behavior of SRGs in a Multinode High Availability setup.

Table 34: Services Redundancy Group Details in Multinode High Availability

Related Services Redundancy Group (SRG)	Managed Services	Operates in	Synchronization Type	When Active Node Fails	Configuration Options
SRG0	Manages security service L4-L7 except IPsec VPN.	Active/active mode	Stateful synchronization of security services	Traffic processed on the failed node will transition to the healthy node in a stateful manner.	• Shutdown on failure option • Install on failure route options

Table 34: Services Redundancy Group Details in Multinode High Availability *(Continued)*

Related Services Redundancy Group (SRG)	Managed Services	Operates in	Synchronization Type	When Active Node Fails	Configuration Options
SRG1+	Manages IPsec and virtual-IP addresses with associated security services	Active/backup mode	Stateful synchronization of security services	Traffic processed on the failed node will transition to the healthy node in a stateful manner.	• Active/ backup signal route • Deployment type • Activeness priority and preemption • Virtual IP address (for default gateway deployments) • Activeness probing • Process packet on backup option • BFD monitoring • IP monitoring • Interface monitoring

NOTE: When you configure monitoring (BFD or IP or Interface) options on SRG1+, we recommend not to configure the shutdown-on-failure option on SRG0.

Starting in Junos OS Release 23.4R1, the Multinode High Availability setup operates in a combined mode. You don't have to reboot the system when you add or delete any SRG (SRG0 or SRG1+) configurations.

Activeness Determination and Enforcement

In a Multinode High Availability setup, activeness is determined at the service level, not at the node level. 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

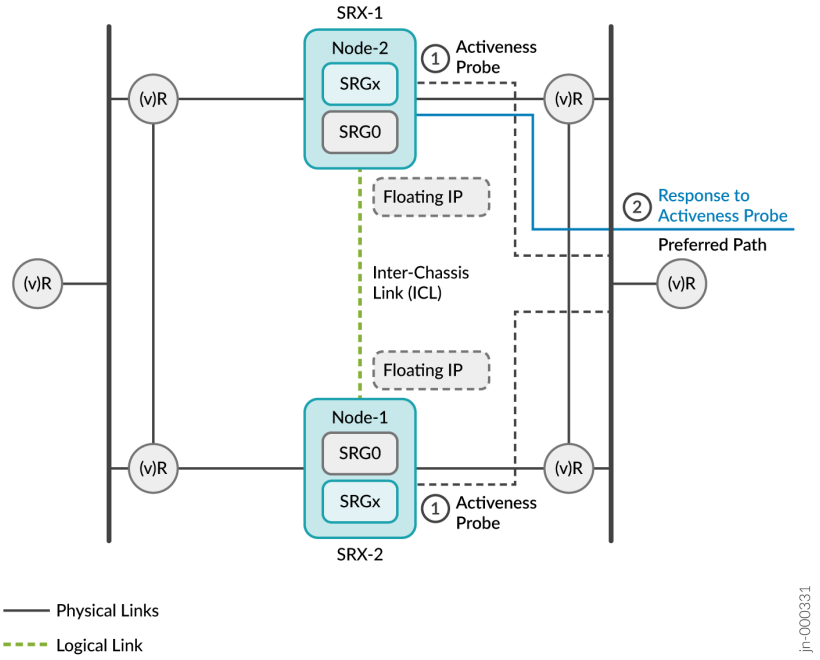
If you prefer a certain node to take over as the active node on boot, you can do one of the followings:

- 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 Firewalls 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 Firewalls. However, the router can have its own preferences on one of the paths depending on the configured metrics.

Figure 47 on page 586 represents the sequence of events for activeness determination and activeness enforcement.

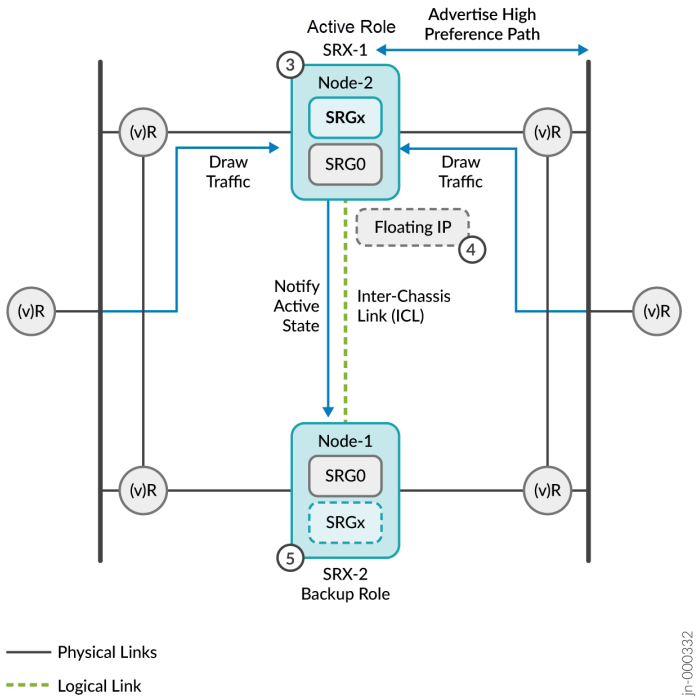
Figure 47: Activeness Determination and Enforcement



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 Firewall that is available on its preferred routing path. In the following example, SRX-1 gets the response from the upstream router.

Figure 48: 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 Firewalls. 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 SRG1+ 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 the Layer 3 side, when the configured signal route enforce activeness with the corresponding route advertisements. On the Layer 2 side, the SRX Series Firewall triggers a Gratuitous ARP (GARP) to the switch layer and owns the VIP and VMAC addresses

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 preemption priority (1-254) for SRG1+. You must configure the preemption value on both 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 698 for the complete configuration example.

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

Configuring Activeness Probe Settings

Starting in Junos OS 22.4R1, default gateway (switching) and in hybrid deployments of Multinode High Availability, you can optionally configure activeness probe parameters using the following statements:

```
[edit]
user@host# set chassis high-availability services-redundancy-group 1 activeness-probe multiplier
<>
user@host# set chassis high-availability services-redundancy-group 1 activeness-probe minimal-
interval <>
```

The probe interval sets the time period between the probes sent to the destination IP addresses. You can set the probe interval as 1000 milliseconds.

The multiplier value determines the time period, after which the backup node transitions to active state, if the backup node fails to receive response to the activeness-probes from the peer node.

The default is 2, and the minimum value is 2, and the maximum is 15.

Example: If you configure the multiplier value to two, the backup node will transition to the active state if it does not receive a response to activeness probing request from the peer node after two seconds.

You can configure `multiplier` and `minimal-interval` in switching and hybrid deployments.

In hybrid mode deployments, if you've configured the probe destination IP details for activeness determination (by using the `activeness-probe dest-ip` statement), then do not configure the `multiplier` and `minimal-interval` values. Configure these parameters when you are using VIP-based activeness probing.

Resiliency and Failover

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

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, the active and backup nodes communicate with each other using an interchassis link (ICL) connected over a routed network or connected directly. The ICL is a logical IP link and it is established using IP addresses that are routable in the network.

Nodes use the ICL to synchronize control plane and data plane states between them. ICL communication could go over a shared or untrusted network and packets sent over the ICL may traverse a path that is not always trusted. Therefore, you must secure the packets traversing the ICL by encrypting the traffic 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 Firewall 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 Firewalls 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.

We recommend following for setting up an ICL:

- 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 Firewall)
- Bind the ICL to the loopback interface (lo0) or an aggregated Ethernet interface (ae0) and have more than one physical link (LAG/LACP) that ensure path diversity for highest resiliency.

- You can use a revenue Ethernet port on the SRX Series Firewalls to setup an ICL connection. Ensure that you separate the transit traffic in revenue interfaces from the high availability (HA) traffic.
- A validation checks have been introduced to restrict the configuration of tunnel MTU for HA link encryption tunnels in a Multinode High Availability setup. The validation check ensures that the end-to-end MTU for HA links using IPv6 encryption meets the minimum requirement of 2000 bytes, helping maintain optimal performance and reliability during high availability operations.

  For example, if your configuration includes the following stanza where tunnel-mtu is less than 2000, you'll receive a commit check error: user@host# set security ipsec vpn L3HA_IPSEC_VPN tunnel-mtu <bytes>

See "Configuring Multinode High Availability" on page 698 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.

Table 35: Commands for Node-Specific PKI Objects

Generating a private/public key pair for a local node	request security pki node-local generate-key-pair

Generating and enrolling a local digital certificate in a local node	• request security pki node-local generate-certificate-request • request security pki node-local key-pair export • 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
Clear node-specific certificates	• clear security pki node-local certificate-request • clear security pki node-local local-certificate • clear security pki node-local key-pair |
Display node-specific local certificates and certificate requests.	• show security pki node-local local-certificate • show security pki node-local certificate-request

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

IN THIS SECTION

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

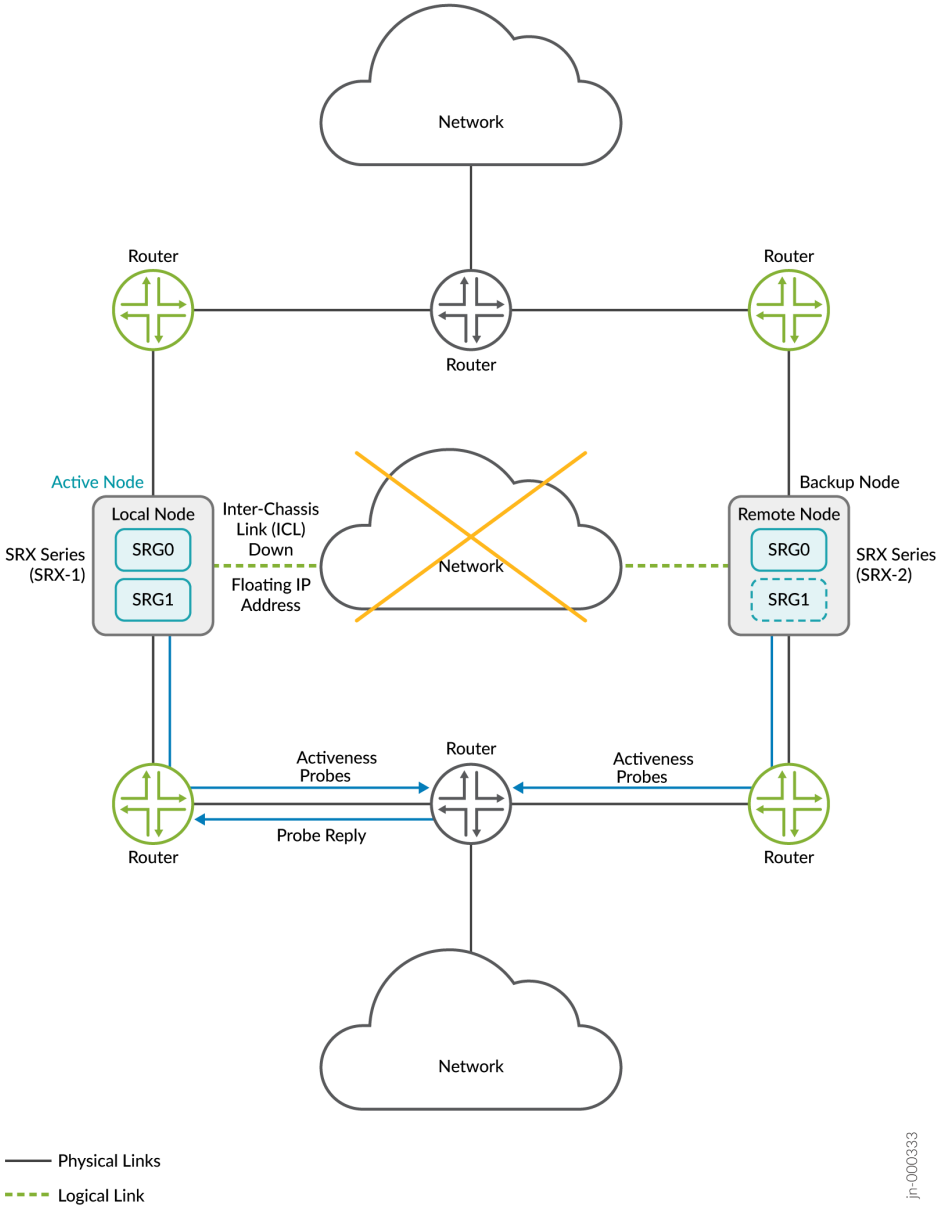
ICMP-Based Split-Brain Probing

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

When the ICL between the nodes goes down, both nodes initiate a 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.

Case 1: If Active Node is Up

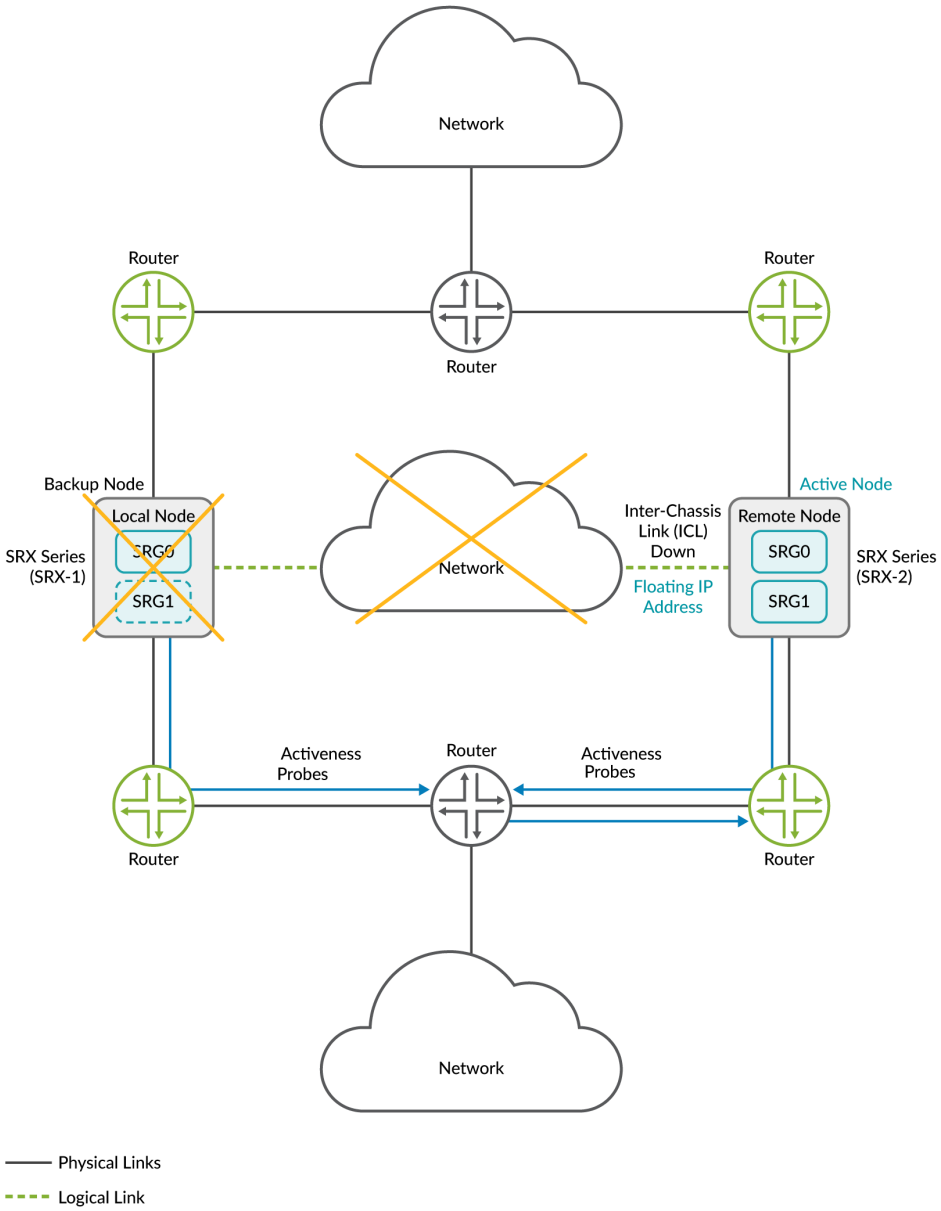
Figure 49: Active Node is Up and ICL is Down



- 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 higher preference path for the active node
- The active node retains the active role.

If Active Node is Down:

Figure 50: Active Node is Down and ICL is Down



- 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.

BFD-Based Split-Brain Probing

In Junos OS Release 23.4R1, we support BFD-based split-brain probing for default gateway and hybrid mode of deployments for Multinode High Availability.

Interchassis Link (ICL) failure can often be attributed to two key factors: network disruptions or inconsistent configurations. You can use activeness probe to determine the node that can take active role for each SRG1+. Based on the probe result, one of the node transitions to the active state and this action prevents spilt-brain scenario.

With BFD-based split-brain probing, you can now have more granular control on the probes as you can define the interface, minimal-interval, and multipliers. In the BFD-based split-brain probing, the probing starts immediately after an SRG is configured and starts functioning. In the default ICMP-based split-brain probing, the probing starts only after ICL link goes down.

In comparison, the BFD-based probing is much more proactive in the following ways to ensures a quicker response to prevent split-brain scenarios:

- The probing initiates directly post an SRG configuration.
- If both ICL BFD and split-brain probe break at the same time, the backup node immediately assumes the active role and takes over the VIP.

This ensures a quicker response to prevent split-brain scenarios.

How it Works?

When the ICL is down and both devices are starting up, the nodes initially enter a HOLD state and wait for the peer node comes up and connect. For any reason if the other node doesn't come up, the system initiates split brain probes to the IP addresses hosted on different device in network. If the process completes successfully, one node transitions to active and the other to backup. Before probe success, if any path monitoring failure/internal hardware monitor failure occurs, then both nodes become Ineligible to prevent a split-brain scenario.

If the split-brain probe fails for any reason, the nodes will remain in the HOLD state and continue probing. The split-brain probe IP must always be available on the network. Except for IPsec, all other application traffic will not experience loss on SRX as long as routing is available, even in the HOLD state.

When both nodes are in Hold or Ineligible state, no traffic will be forwarded until the node becomes active/backup again.

Note:

- Split brain is based on the **activeness probes** different from **path monitor probes**. It only triggered when ICL/communication is broken b/w MNHA nodes
- When the Interchassis Link (ICL) between nodes is broken, both nodes initiate split brain probes. The active node retains mastership as long as its probe does not fail. It is recommended to host the probing IP on a path that ensures continuous reachability, provided the SRX Series node is healthy. A state change is triggered only if the probe from the current active node fails and the probe from the current backup node succeeds.
- In switching and hybrid modes, traffic steering uses the Virtual IP (VIP), which only works in the ACTIVE state. The system should not stay in the HOLD state after the hold timer expires, as it will probe the MNHA peer to resolve the split-brain situation.

Difference in ICMP-Based and BFD-Based Probing

The following table shows differences in ICMP-based probing and BFD-based probing for split-brain detection.

Table 36: Difference in ICMP-Based and BFD-Based Probing

Parameters	ICMP-Based Probing	BF
Probe type	ICMP packet	BF
Minimal interval	1000 ms	SR Fo SR SR
SRG backup node probes	Yes	Yes
SRG active node probes	No	Yes
SRG split-brain resolving when ICL down	Only when ICL goes down.	Aft
	Not possible	Po

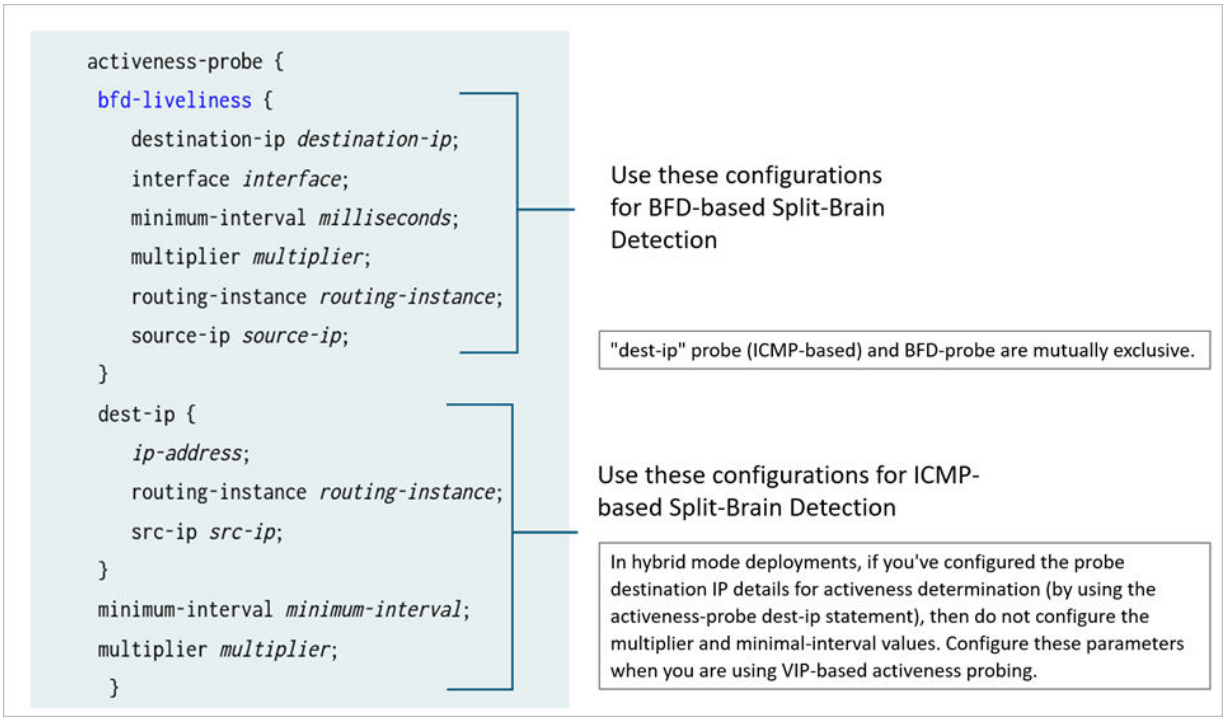
Table 36: Difference in ICMP-Based and BFD-Based Probing *(Continued)*

Parameters	ICMP-Based Probing	BF

Configuration options	show chassis
	high-availability services-redundancy-group 1
	activeness-probe
	dest-ip {
	192.168.21.1;
	src-ip 192.168.21.2;
	}
		sho
		hig
		act
		bfd
		
		int
		
		SRX
		
		}

The following figure shows configuration options for ICMP-based probing and BFD-based probing for split-brain detection.

Figure 51: Activeness-Probes Configuration for ICMP-Based and BFD-Based Probing



NOTE: ICMP-based probing and BFD-based probes are mutually exclusive.

In hybrid mode and default gateway deployments, you can configure the activeness-probe interval and threshold at following two levels:

- Global-level which is applicable to ICMP-based split-brain probing
- BFD-Liveliness level which is specific to BFD split-brain probe. When you configure BFD-based probing, do not configure global `minimum-interval` and `multiplier` options under `activeness-probe` statement.

To configure activeness probe for default gateway deployments, use the primary virtual IP (VIP1) address interface on both nodes (local and peer) to set up your activeness probe. The destination IP is from the peer node, and the source IP is from your local node. Both VIPs must have the same index value. The IP addresses must be the inet addresses assigned to the LAN interface of the SRX Series Firewall.

Configure Split-Brain Probing

You can configure split-brain probing on a Multinode Node High Availability setup in the following ways:

- Routing and Hybrid mode —If you've configured the probe destination IP details for activeness determination (by using the `activeness-probe dest-ip` statement), then do not configure the multiplier and minimal-interval values. Configure these parameters when you are using VIP-based activeness probing.

```
[edit]
[set chassis high-availability services-redundancy-group 1 activeness-probe
dest-ip <neighbor_ip_address> src-ip <srx_anycast_IP>]
```

- Hybrid and switching mode—Layer 2 split-brain probing using ICMP. Use the probe type ICMP and set interval and timeout threshold using the following statement:

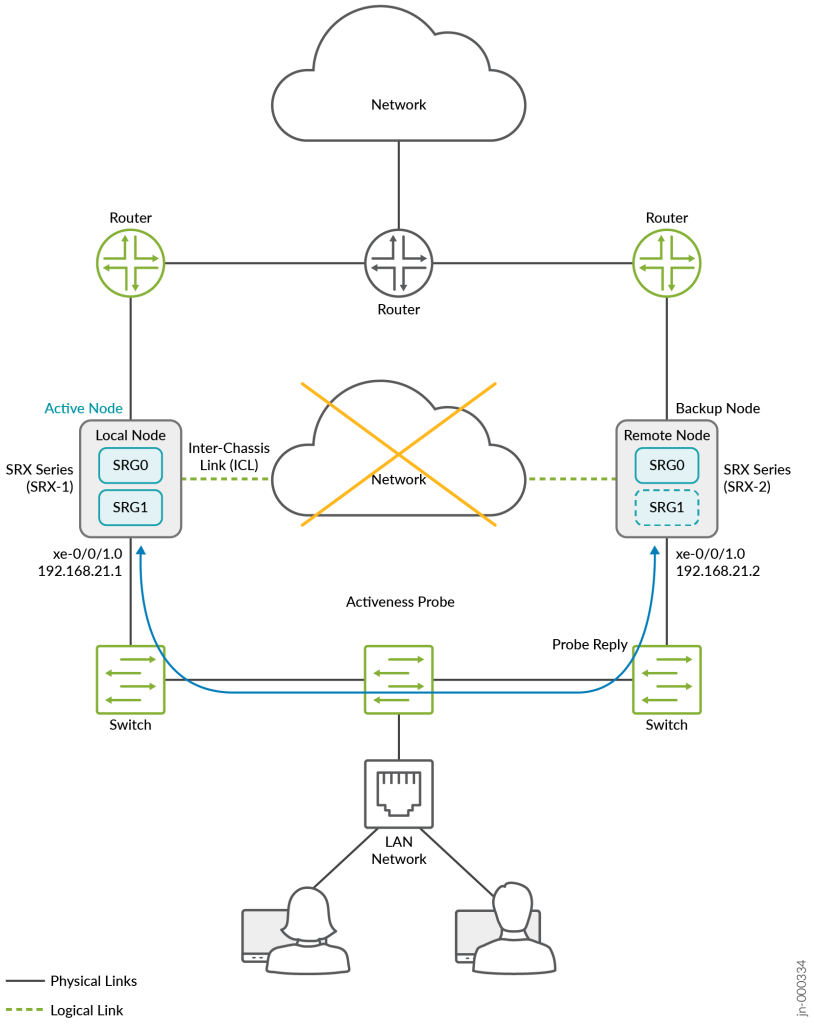
```
[edit]
[set chassis high-availability services-redundancy-group 1 activeness-probe
minimum-interval <interval> multiplier <integer>
```

- Hybrid and switching mode—Layer 2 split-brain probing using BFD. Use the probe type BFD and set the timeout threshold that can be sub-second based on the BFD minimum-interval configured.

```
[edit]
[set chassis high-availability services-redundancy-group 1 activeness-probe
bfd-liveliness source-ip <ip-address> destination-ip <ip-address> interface
<vip1_ifl_interface> minimum-interval <interval> multiplier <integer>
```

Figure 52 on page 601 shows the sample topology. Two SRX Series Firewalls are connected to adjacent routers on trust and untrust side forming a BGP neighborship. 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.

Figure 52: Multinode High Availability Configuration for Split-Brain Probing



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.

For activeness-probe, you must configure the following options:

- Source IP address: Use the virtual IP address 1 (VIP1) of SRG1 of the local node.
- Destination IP address: Use the VIP1 of SRG1 of the peer node.
- Interface: Interface associated with VIP1

In this example, assign a virtual IP (VIP) address (192.168.21.1) and an interface xe-0/0/1.0 for BFD liveliness. Here, you configure BFD-based split-brain probing by specifying source and destination IP addresses and the interface.

The nodes use the family inet address of the interface associated to virtual IP address (VIP1) of SRG1.

Both nodes initiate an activeness determination probe (BFD-based probe) as soon as the SRGs start operating.

NOTE: For BFD-based split-brain probing, you must:

- Configure matching source and destination IP addresses for the same SRG on both nodes.
- Configure the `activeness-priority` option to determine active node as a result of split-brain probing.

The following table shows how Multinode High Availability setup resolves split-brain situation with BFD-based probing when the ICL is down. Depending on node states and probe results, Multinode High Availability system selects the node to take up the active role.

Table 37 on page 602 shows how Multinode High Availability setup resolves split-brain situation with BFD-based probing when the ICL is down. Depending on node states and probe results, Multinode High Availability system selects the node to take up the active role.

In this example, we assume that SRG1 of node 1 has the higher activeness-priority.

Table 37: Activeness Determination Using Split-Brain Probes and Node States

State of Node 1	Probing State of Node 1	State of Node 2	Probing State of Node 2	Node Transitioning to SRG1 Active State
Active	Down	Ineligible	No probing	Node 1
Active	Up	Backup	Up	Node 1
Active	Up	Active	Up	Node 1 (Tie breaker)
Backup	Down	Ineligible	No probing	Node 1
Backup	Up	Backup	Up	Node 1 (Tie breaker)
Backup	Up	Active	Up	Node 2
Ineligible	No probing	Ineligible	No probing	Neither Node
Ineligible	No probing	Backup	Down	Node 2

Ineligible	No probing	Active	Down	Node 2

Sample Configuration

Node 1:

```
set chassis high-availability services-redundancy-group 1 activeness-priority 1
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
destination-ip 192.168.21.2
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness source-
ip 192.168.21.1
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
interface xe-0/0/1.0
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
minimum-interval 300
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
multiplier 3
```

Node 2:

```
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
destination-ip 192.168.21.1
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness source-
ip 192.168.21.2
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
interface xe-0/0/1.0
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
minimum-interval 300
set chassis high-availability services-redundancy-group 1 activeness-probe bfd-liveliness
multiplier 3
```

Verification

- Use the `show chassis high-availability services-redundancy-group 1` command to see the type of split-brain probe configured on the device.

(BFD-based Probing)

```
user@host> show chassis high-availability services-redundancy-group 1
..
        Split-brain Prevention Probe Info:
          DST-IP: 192.168.21.2
          SRC-IP: N/A
          Routing Instance: default
          Type: BFD Probe
          Interval: 300ms   Multiplier: 3
          Status: RUNNING
          Result: REACHABLE           Reason: N/A
..
```

(ICMP-based Probing)

```
 user@host> show chassis high-availability services-redundancy-group 1
..
Split-brain Prevention Probe Info:
          DST-IP: 192.168.21.2
          SRC-IP: 192.168.21.1
          Routing Instance: default
          Type: ICMP Probe
          Status: NOT RUNNING
          Result: N/A           Reason: N/A
..
```

- Use the `show bfd session` command to see if BFD-based probe status.

```
user@host> show bfd session
                                                Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
192.168.0.2                Up                      0.300    0.100        3
192.168.21.2               Up        xe-0/0/1.0    0.300    0.100        3
1 sessions, 1 clients
Cumulative transmit rate 0.5 pps, cumulative receive rate 0.0 pps
```

In the sample, you can notice that BFD-based split-brain probing is running for interface xe-0/0/1.0.

- Use the `show chassis high-availability services-redundancy-group 1` command to get the details pf BFD-based probes.

```
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: 1
           Status : N/A
           Health Status: SRG NOT CONFIGURED
           Failover Readiness: UNKNOWN
           Activeness Remote Priority: 100
```

Logical Systems and Tenant Systems Support

Logical systems for SRX Series Firewalls enable you to partition a single device into secure contexts and A tenant system logically partitions the physical firewall into separate and isolated logical firewall.

A tenant system logically partitions the physical firewall into separate and isolated logical firewall. Although similar to logical systems, tenant systems have much higher scalability and fewer routing features.

SRX Series Firewalls in Multinode High Availability setup support logical systems and tenant systems on services redundancy group 0 (SRG0).

The behavior of a Multinode High Availability setup with SRX Series Firewalls running logical systems is the same as that of a setup where the SRX Series nodes do not run logical systems. There is no difference in the events that trigger a node failover. Specifically, if the interface monitoring is enabled under SRG0 and a link associated with a single logical system fails (which is being monitored), the device fails over to another node. This failover occurs through route preference advertisements in the Multinode High Availability setup.

Before setting up the logical or tenant systems, you must configure the Multinode High Availability. Each node in the high availability setup must have an identical configuration. Ensure that the logical systems or tenant systems' name, profile, and corresponding security features, or interfaces within the logical systems or tenant systems are same. All logical or tenant system configurations are synchronized and replicated between the two nodes.

TIP: Use Junos configuration groups to configure features and functions, and synchronize the configuration by using the `[edit system commit peers-synchronize]` option in your Multinode High Availability setup. See Configuration Synchronization Between Multinode High Availability Nodes.

When using SRX Series Firewalls with logical systems in an Multinode High Availability, you must purchase and install the same number of licenses for each node in the setup.

For more information, see Logical Systems and Tenant Systems User Guide for Security Devices.

Change History Table

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

Release	Description
25.4R1	Validation checks to ensure that HA links using IPv6 encryption maintain an end-to-end MTU of at least 2000 bytes

Four-Node and Three-Node Multinode High Availability

SUMMARY

Learn about the four-node and three-node Multinode High Availability solution.

IN THIS SECTION

The four-node Multinode High Availability feature in Junos OS provides robust protection against service interruptions. The existing MNHA infrastructure supports redundancy with two firewalls functioning as a pair. Four-node MNHA enhance redundancy and failover capabilities by supporting additional redundancy between two pairs of MNHA devices across the domains. That is, if one pair of devices fails or goes offline, the other pair will automatically take over the services provided by the failed pair. The two pairs of devices can be located in different places, such as separate data centers, ensuring that services remain operational even if one location experiences issues.

Four-node MNHA features:

- **SRG0 Support**: Four-node MNHA support SRG0, which is suitable for services without control plane states, such as firewall and NAT. Control plane states refer to the information required to manage and control network operations.
- **SRG1+ Limitations**: Four-node MNHA currently does not support SRG1+ services, which involve control plane states, such as IPsec VPN.
- **Operating Modes**: Four-node MNHA support only the routing mode of MNHA. It will not include support for switching mode, hybrid mode, or cloud mode, which are different operational configurations for handling network traffic and redundancy.

- **Routing Limitations**: The four-node MNHA infrastructure does not support asymmetric routing within the same domain or across two domains. Therefore, external routers must send packets of the same bidirectional flow to the same node consistently.

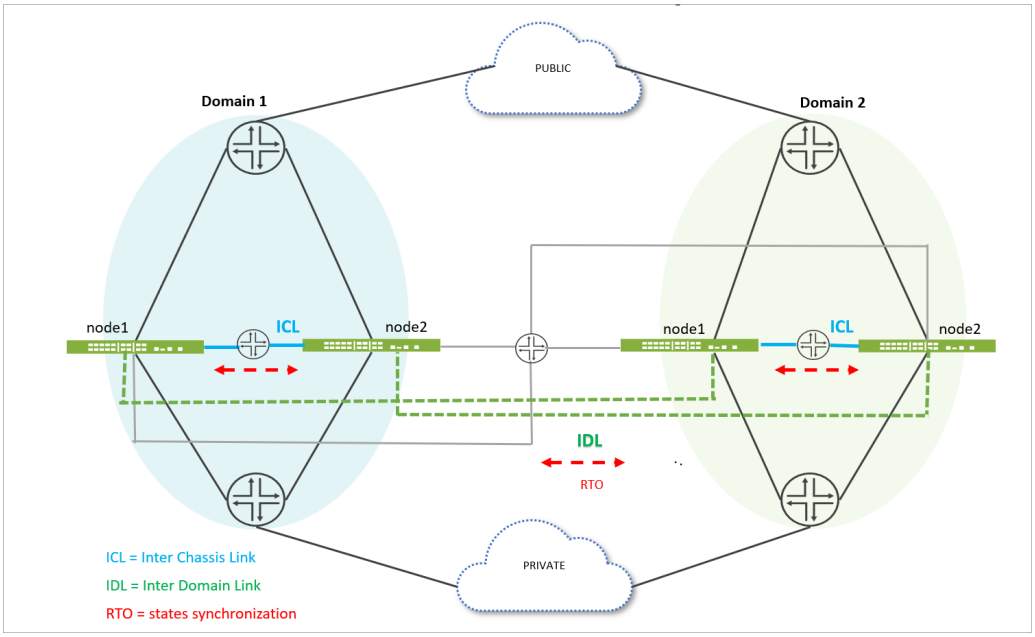
Benefits of four-node MNHA:

- Provides enhanced service continuity. It allows seamless failover of services across nodes, ensuring uninterrupted uptime for critical network functions like firewalls and NAT without compromising control plane states.
- Supports high availability across geographically distributed data centers, providing resilience against localized disruptions.

Deployment Scenario

The four-node MNHA setup involves four identical SRX Series Firewalls, all having the same hardware configurations. These devices are organized into two MNHA domains, with each domain containing two firewalls.

Figure 53: Four-Node Multinode High Availability



In the topology:

- Domain 1 consists of two firewalls—node1.1 (domain 1 node 1) and node1.2 (domain 1 node 2) .

- Domain 2 consists of two firewalls—node2.1 (domain 2 node 1) and node2.2 (domain 2 node 2) .

Inside each domain, the two nodes communicate through an Interchassis link (ICL), which is a direct connection that allows for fast and efficient communication. Across the two domains, the nodes communicate through inter-domain link (IDL). Both ICL and IDL are logical links, which can operate over dedicated interfaces or in-band traffic interfaces. Both ICL and IDL can be routed and use IPsec encryption.

IDL is a layer-three routable links that enable communication between nodes in different domains. Each IDL is configured on an interface of a node and is part of a private routing instance. This set up ensures that only internal communication packets between MNHA nodes across domains can traverse these links, maintaining security and integrity.

All four nodes are connected to external layer-three routers, which handle traffic from both public and private networks. This setup allows the Firewalls to manage network traffic efficiently.

To set up the four-node MNHA, we recommend first establish MNHA pairs within each domain by connecting two nodes through ICL. Then, connect these pairs (from different domains) through IDL to complete the full MNHA structure.

How Four-Node Multinode High Availability Works

IN THIS SECTION

The four-node MNHA architecture is structured into two domains, each consisting of two nodes. This setup enables redundancy at two levels: within a domain and across domains.

- **Within-Domain Redundancy:**
  - Each domain has two peer nodes that back each other up. If one node in the domain fails or is out of service, the other node takes over its responsibilities.
  - The nodes within a domain can operate in two modes:
    - Active-backup mode: One node is active, handling all tasks, while the other is on standby, ready to take over if the active node fails.
    - Active-active mode: Both nodes are actively handling tasks, sharing the load, and providing immediate failover capabilities.

- **Across-Domain Redundancy:**
  - The nodes in one domain are backed up by nodes in the other domain. If all nodes in one domain fail, the nodes in the other domain take over the services.
  - This redundancy also supports both active-backup and active-active modes, similar to the within-domain redundancy.
- **Communication and Synchronization:**
  - **ICL:** This is the communication link between the two nodes within the same domain. It allows them to synchronize their states and keep each other updated about their status.
  - **IDL:** This newly introduced link facilitates communication between nodes across different domains, allowing them to synchronize states and share status information.
- **State Synchronization:**
  - Each node synchronizes its states with its peer in the same domain via the ICL.
  - The node also synchronizes states with one of the nodes in the other domain via the IDL.
  - When a node receives state information from another domain via the IDL, it relays this information to its peer node in the same domain through the ICL.
    Example: In a setup shown in topology includes four nodes labelled node 1.1, node 1.2, node2.1, and node 2.2.
    - Node 1.1 synchronizes its state with node 1.2 via ICL within the local domain.
    - Node 1.1 also synchronizes with node 2.1 in the remote domain via IDL.
    - Node 2.1, upon receiving this state, forwards it to node 2.2 in the same domain via ICL.

Cold-Synchronization in Four-Node MNHA

Cold Sync of states is a mechanism used in a network of interconnected nodes to ensure that all nodes have an up-to-date and consistent replica of states. When all nodes boot up, the nodes synchronize states within their domain, then proceed to synchronize across domains. If one node reboots, it will synchronize its states with a neighboring node in the same domain first, then continue synchronizing with nodes in remote domains to ensure it has all necessary states.

Following sequences are involved in cold-sync process in an MNHA setup:

1. **Cold Sync Initiation**:

   Cold sync is initiated when a node needs to create or update its replica of states from its neighboring nodes in following scenarios:

- A node boots up and connects to other nodes.
- A node reconnects with its peer node in the same domain after a temporary disconnection (ICL flap).
- A node reconnects with its peer node in a remote domain after a temporary disconnection (IDL flap).

2. **Cold Sync Process**:
   - Nodes are identified by domain ID and node ID. The node with the lower ID will initiate the cold sync by requesting states from the node with the higher ID.
   - States include active states and any states received from remote domains.
     - Within the same domain: Nodes exchange states with each other via ICL based on their node IDs. The node with the lower ID requests states first.
     - Across different domains: After completing cold sync within the same domain, nodes begin syncing across domains via IDL.

The process ensures that all nodes eventually have a complete and consistent set of states.

Software Upgrade in a Four-Node MNHA Setup

IN THIS SECTION

Use the following steps to upgrade software in a four-Node MNHA

1. Ensure your Multinode High Availability setup is healthy, functional, and that the interchassis link (ICL) is up. This step is need to ensure that another node within the same domain is ready to take over the responsibilities of the node being upgraded.
2. Disconnect the inter-domain link (IDL) to isolate the node from the rest of the network, preventing any potential disruptions during the upgrade.
3. Initiate the software upgrade process on the backup node (node 1.2) in domain 1 and commit the configuration using the `set chassis high-availability software-upgrade` statement.
4. Confirm that the other device (node 1.1) is in an active role and is functioning normally.

5. Install the Junos OS software on the node 1.2 and perform a reboot to apply the updates.
6. Reboot the device using the `request system reboot` or `request vmhost reboot` command (depending on your platform) after successful installation.
7. Check that the node is running the correct software version post-reboot using the `show version command`..
8. Check status of the multinode high availability on the device.
9. Remove the `software-upgrade` statement on node 1.2 and commit the configuration. This steps enables the system to return to its normal operating mode.
10. Re-establish the inter-domain link to reintegrate the upgraded node into the network.
11. Repeat the steps for the next node (node 1.1) within the same domain, ensuring each node is upgraded sequentially to maintain network stability.
12. After completing upgrades in the current domain, move on to another domain (domain 2) and follow the same procedure to ensure a consistent and orderly upgrade process across the network.

Licensing and Topology Considerations

A four-node MNHA setup requires a specific four-node MNHA feature license. If the license is missing, the following warning appears during commit and in syslog:

```
License needed for the feature 4-node MNHA.
```

Licenses are unique to each SRX Series and cannot be shared between the nodes in a Multinode High Availability setup. Also ensure to use identical licenses for Layrer 7 capabilities such as AppID, IDP, Content Security on all four firewalls. If all four SRX Series Firewalls do not have an identical set of licenses, the system is not ready for the deployment.

Four-Node MNHA Configuration Requirements

In a Multinode high availability (MNHA) network setup, ensuring that sessions remain consistent across multiple nodes is crucial for maintaining seamless network operations and failover capabilities. Previously, there was a requirement that all nodes within an MNHA configuration had to use the same ingress and egress interface names, zone names, and policy names for a given session. This requirement could be challenging to meet, especially in a four-Node MNHA configuration, as it requires meticulous configuration management across all nodes.

The four-node MNHA relaxes some of these restrictions. Specifically, it removes the need for ingress and egress interfaces to have the same names across all nodes within the MNHA domain. This change

means that while previously, the same session on different nodes had to use interfaces with identical names (such as ge-0/0/0 on all nodes), the new configuration allows these interfaces to have different names on each node.

Example: A session on node1.1 have ingress interface ge-0/0/0 and egress interface ge-0/0/1, when the same session is active on node 2.1 can have ingress interface ge-0/0/2 and egress interface ge-0/0/3. But, ge-0/0/0 and ge-0/0/2 should be in the same zone, and similarly ge-0/0/1 and ge-0/0/3 must be in a same zone.

You must meet the following requirements to ensure secure and reliable session handling

1. **Zone consistency**: The interfaces used for the same session on different nodes must belong to the same zone (as described in previos section).
2. **Policy consistency**: The policies applied to these zones must remain consistent across all nodes. This setting ensures that the security and routing behaviors remain the same, even if the physical interfaces differ.
3. **Routing instances/VRF consistency**: Similarly, VR names must remain the same across nodes to ensure consistent routing behavior.

Configuration Overview

The following configuration snippets outline the high-level steps required to set up a four-node MNHA system

1. Local domain and node ID configuration:
   - **Set the local domain ID and size:**

     ```
     user@host# set chassis high-availability local-domain-id <domain-id> domain-size <size>
     ```

     - `domain-id`: Unique identifier for the local domain (either 1 or 2).
     - `size`: Number of nodes in the domain (1 or 2). For a four-node MNHA, domain size is 2.
   - **Set the local node ID:**

     ```
     user@host# set chassis high-availability local-id <local-node-id>
     ```

     `local-node-id`: Unique identifier for the local node within its domain (1 through 10).
2. Inter-Chassis Link (ICL) configuration:

This configuration is for connecting the local node to a peer node within the same domain.

- Set local node IP:

```
[edit]
user@host#  set chassis high-availability local-id local-ip <ip-address>
```

3. Peer node configuration:

```
[edit]
user@host# set chassis high-availability peer-id <peer-node-id> peer-ip <ip-address>
user@host# set chassis high-availability peer-id <peer-node-id> interface <logical-interface-
name>
user@host# set chassis high-availability peer-id <peer-node-id> liveness-detection minimum-
interval <interval-in-ms>
user@host# set chassis high-availability peer-id <peer-node-id> liveness-detection multiplier
<multiplier-value>
user@host# set chassis high-availability peer-id <peer-node-id> vpn-profile IPSEC_VPN_ICL
```

4. Inter-domain link (IDL) configuration: This configuration is for connecting nodes across different domains.

```
user@host# set chassis high-availability peer-domain-id <domain-id> domain-size <size>
user@host# set chassis high-availability peer-domain-id <domain-id> peer-id <peer-node-id>
local-ip <ip-address>
user@host# set chassis high-availability peer-domain-id <domain-id> peer-id <peer-node-id>
peer-ip <ip-address>
user@host# set chassis high-availability peer-domain-id <domain-id> peer-id <peer-node-id>
interface <logical-interface-name>
user@host# set chassis high-availability peer-domain-id <domain-id> peer-id <peer-node-id>
liveness-detection minimum-interval <interval-in-ms>
user@host# set chassis high-availability peer-domain-id <domain-id> peer-id <peer-node-id>
liveness-detection multiplier <multiplier-value>
user@host# set chassis high-availability peer-domain-id <domain-id> peer-id <peer-node-id>
vpn-profile profile-name
```

5. Service redundancy group (SRG0) configuration:

- Set peer node in the same domain ((1 through 10):

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

- Set peer node in a different domain:

```
user@host# set chassis high-availability services-redundancy-group 0 peer-domain-id
<domain-id> peer-id <peer-node-id>
```

NOTE: Ensure the following criteria is met:

- Each domain ID must be unique and the ID can only be 1 or 2.
- Node IDs are unique within the same domain.
- The MNHA infrastructure supports four-node configurations for SRG0 only if a `peer-domain-id` is configured. Without it, the setup supports a 2-node configuration.
- Reboot the device once you configure local domain ID, local node ID, peer domain ID, peer node ID, and domain size.

Use the commands such as `show chassis high-availability information` and `show chassis high-availability peer-info` to verify if your configuration is working as expected.

Inter-Domain Link (IDL) Encryption

IN THIS SECTION

Four-node Multinode High Availability (MNHA) with Inter-Domain Link (IDL) enhances high availability by extending it across data center domains. IDL facilitates communication between nodes across different domains, allowing the nodes to synchronize states and share status information.

IDL is a logical IP link and it is established using IP addresses that are routable in the network.

Nodes across the domains use the IDL to synchronize control plane and data plane states between them. IDL communication could go over a shared or untrusted network and packets sent over the IDL might traverse a path that is not always trusted. Therefore, you must secure the packets traversing the IDL by encrypting the traffic using IPsec standards.

IPsec protects traffic by establishing an encryption tunnel for the IDL. When you apply HA link encryption, the HA traffic flows between the nodes across the domains only through the secure, encrypted tunnel. Without HA link encryption, communication between the nodes might not be secure.

IDL supports IKEv2 and a custom Multi-SA implementation, to securely exchange encrypted data across domains through IPsec VPNs. The system supports robust AES-GCM-256 encryption and both PSK and PKI authentication, ensuring secure inter-domain communications.

Configuration Overview

To encrypt the HA link for the ICL:

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

```
user@host> 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 Firewalls uses the IKEv2 protocol.

```
user@host# set chassis high-availability peer-domain-id domain-id peer-id peer-node-id vpn-
profile
```

- 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.
```

- Verify the details using the `user@host> show security ipsec statistics ha-link-encryption` command to display the tunnel type (ICL or IDL).

We recommend the following settings for an IDL:

- 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 Firewall).

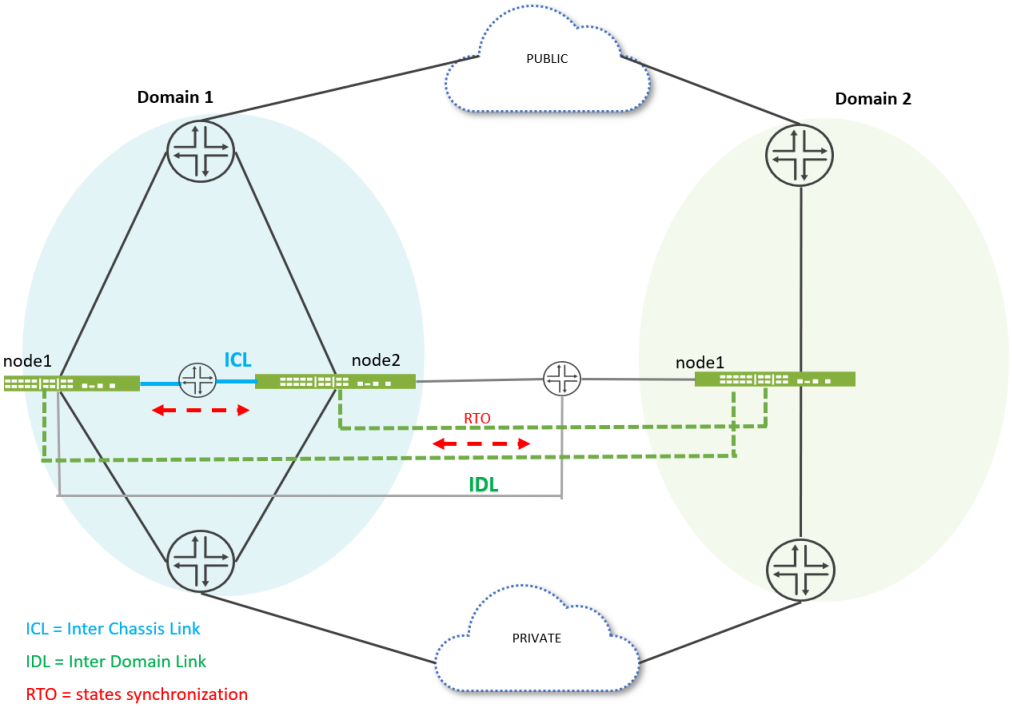
- You can use a revenue Ethernet port on the SRX Series Firewalls to setup an IDL connection. Ensure that you separate the transit traffic in revenue interfaces from the high availability (HA) traffic.

Three-Node Multinode High Availability

IN THIS SECTION

The Three-Node MNHA setup involves three identical SRX Series Firewalls, all having the same hardware configurations. These devices are organized into two MNHA domains, with one domain containing two firewalls and another domain containing one firewall.

Figure 54: Three-Node Multinode High Availability



In a three-node MNHA setup, node 1.1 and node 1.2 belong to domain 1, while node 2.1 is in domain 2. For a four-node MNHA configuration, each node is equipped with one Inter-Cluster Link (ICL) and one

Inter-Domain Link (IDL). In a three-node MNHA configuration, the two nodes in domain 1 each have one ICL and one IDL, whereas the node in domain 2 has two IDLs and no ICL.

State Synchronization:

- In domain 1, node 1.1 synchronizes its states with its peer (node 1.2) through ICL.
- Node 1.1 also synchronizes states with Node 2.1 in the other domain through the IDL.
- Node 2.1 synchronizes states to node 01.1 and Nnode 1.2 through IDL .
- When a node receives state information from another domain through the IDL, it does not relay this information to its peer node within the same domain via the ICL. This approach prevents unnecessary duplication of data transmission within the domain.

In the event of a single IDL failure—specifically, if the direct link between node 1.1 and Nnode 2.1 is down—routing is employed to maintain connectivity. Node 1.1 can still communicate with node 2.1 by using a predefined route through Nnode 1.2. This routing ensures that the IDL between node 1.1 and node 2.1 remains operational, eliminating the need for forwarding packets from IDL to ICL peers, whether for hot sync or cold sync processes.

Configuration Overview of Three-Node MNHA

In a three-node MNHA (Multinode High Availability) setup, devices are divided into two domains: one domain hosts two firewalls, and the other hosts a single firewall. The configuration snippet below demonstrates how to configure MNHA for the domain with a single node, focusing on local and peer node settings. Note that Inter-Chassis Link (ICL) configuration is not required for a single-node domain. The configuration for the other domain (domain with two nodes) is similar to the configuration for a four-node MNHA domain.

1. Configure local ID, local domain ID, and domain size.

```
[edit]
user@host# set chassis high-availability local-id <local-node-id>
user@host# set chassis high-availability local-id local-ip <local-ip>
user@host# set chassis high-availability local-domain-id <domain-id>
user@host# set chassis high-availability local-domain-id domain-size 1
```

Domain size 1 indicates a domain that includes only one node in a three-node MNHA.

2. Configure peer node.

```
[edit]
user@host# set chassis high-availability peer-domain-id 1 domain-size 2
```

3. Configure IDL for peer-node 1.

```
[edit]
user@host# set chassis high-availability peer-domain-id 1 peer-id 1 local-ip <local-ip-
address>
user@host# set chassis high-availability peer-domain-id 1 peer-id 1 peer-ip <peer-ip-address>
user@host# set chassis high-availability peer-domain-id 1 peer-id 1 interface <interface>
user@host# set chassis high-availability peer-domain-id 1 peer-id 1 vpn-profile <profile-name>
user@host# set chassis high-availability peer-domain-id 1 peer-id 1 liveness-detection
minimum-interval <interval-in-ms>
user@host# set chassis high-availability peer-domain-id 1 peer-id 1 liveness-detection
multiplier <multiplier-value>
```

4. Configure IDL for peer-node 2.

```
[edit]
user@host# set chassis high-availability peer-domain-id 1 peer-id 2 local-ip <local-ip-
address>
user@host# set chassis high-availability peer-domain-id 1 peer-id 2 peer-ip <peer-ip-address>
user@host# set chassis high-availability peer-domain-id 1 peer-id 2 interface <interface>
user@host# set chassis high-availability peer-domain-id 1 peer-id 2 vpn-profile <profile-name>
user@host# set chassis high-availability peer-domain-id 1 peer-id 2 liveness-detection
minimum-interval <interval-in-ms>
user@host# set chassis high-availability peer-domain-id 1 peer-id 2 liveness-detection
multiplier <multiplier-value>
```

5. Configure service redundancy group.

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

Use the commands such as `show chassis high-availability information` and `show chassis high-availability peer-info` to verify if your configuration is working as expected.

Prepare Your Environment for Multinode High Availability Deployment

IN THIS SECTION

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 Firewall 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.

For the complete list of supported features and platforms, see Multinode High Availability in Feature Explorer.

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 Firewall 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 Firewalls 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.

We recommend to bind the ICL to the loopback interface (lo0) or an aggregated Ethernet interface (ae0) and have more than one physical link (LAG/LACP) that ensure path diversity for highest resiliency. You can also use a revenue Ethernet port on the SRX Series Firewalls to setup an ICL connection. Ensure that you separate the transit traffic in revenue interfaces from the high availability (HA) traffic.

IP Address Consideration

Table 38 on page 622 provides details on IPv4 and IPv6 address support for Multinode High Availability deployments.

Table 38: 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

Support available for the configuration of IPv6 addresses for the active signal route, backup signal route, and install on failure route options under services-redundancy-group configurations on your MNHA setup.

NOTE: Configure only one VIP per logical interface (IFL) in a Multinode High Availability setup. Support for using multiple VIPs or dual-stack is not available.

Using IP Address Pools in Multinode High Availability Configuration

When you configure multiple SRGs (active-active mode) in Multinode High Availability, ensure that address pools used by SRGs in an access profile must not overlap. Also ensure that address and address pool configured in the RADIUS server for the hosts connected to different SRGs must be unique.

Example: Following sample shows address pool configurations with access profile `localpool` and `localpool2` for SRG1 and SRG2 respectively:

```
[edit]
set groups manha_config_group access profile localpool address-assignment pool v4-pool1
set groups manha_config_group access profile localpool2 authentication-order none
set groups manha_config_group access profile localpool2 address-assignment pool v4-pool2
set groups manha_config_group access address-assignment pool v4-pool1 family inet network
192.0.2.0/24
set groups manha_config_group access address-assignment pool v4-pool1 family inet range v41 low
```

```
192.0.2.1
set groups manha_config_group access address-assignment pool v4-pool1 family inet range v41 high
192.0.2.127
set groups manha_config_group access address-assignment pool v4-pool2 family inet network
192.0.2.0/24
set groups manha_config_group access address-assignment pool v4-pool2 family inet range v41 low
192.0.2.128
set groups manha_config_group access address-assignment pool v4-pool2 family inet range v41 high
192.0.2.255
```

In this example, Services Redundancy Groups - SRG1 and SRG2 - are in the same network (192.0.2.0/24). However, IP addresses in address pools are distributed to avoid overlapping (192.0.2.1/24—192.0.2.127 for SRG1 and 192.0.2.128—192.0.2.255 for SRG2).

Similarly you must use unique IP address and address pools for user configurations in the RADIUS server.

In case you assign same address for hosts in two SRGs, then Multinode High Availability deletes the new host and halts IKE negotiations with the following message:

```
AUTHENTICATION_FAILED as the AUTH response
```

System Log displays the following message:

```
Duplicate assigned IPv4 received, delete new peer
```

Change History Table

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

Release	Description
25.4R1	Support for IPv6 addresses for the active signal route, backup signal route, and install on failure route options under services-redundancy-group configurations in an MNHA setup.

RELATED DOCUMENTATION

- Two-Node Multinode High Availability | **573**
- Example: Configure Multinode High Availability in a Default Gateway Deployment | **743**
- Example: Configure Multinode High Availability in a Layer 3 Network | **698**

Multinode High Availability Services

IN THIS SECTION

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, Content Security, 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 Firewalls in Multinode High Availability.

To configure the control plane stateless services:

- Configure the features as you configure them on a stand-alone SRX Series Firewall.
- 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 gets 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 Firewalls. To steer the response path for the NAT traffic (destined to NAT pool IP address) to the correct SRX Series Firewall (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 Firewalls 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`).

NOTE: When synchronizing configuration between nodes, verify that authentication, policy, source zone, and destination zone details match on both nodes. Maintaining the same order in your configuration ensures successful synchronization of authentication entries across both nodes.

If you clear an authentication entry in one node using the **clear security user-identification local-authentication-table** statement, ensure that you clear the authentication entry in the other node also.

Follow the same practice in case of asymmetric traffic configuration as well.

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 authentication entries using the show commands, each node displays only those authentication 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`

Express Path Support

Express Path (formerly known as *services offloading*) in MNHA (both SRG0 and SRG1+) reduces latency by ensuring seamless packet forwarding post-failover. This feature installs stateful and static SOF sessions on the other node while the active node is operational, ensuring immediate failover readiness driven by control plane messages.

The system prevents SOF sessions from aging prematurely in the other node, maintains session integrity, and handles first packet processing seamlessly during primary role transitions. Additionally, it addresses session handling in new failed node by opting for deletion and reinstallation of SOF sessions to ensure up-to-date management.

Support for this feature is available for Multinode High Availability system operating in Layer 3 (Routing) mode. for details on Express Path, see Express Path Overview.

Configuration Synchronization Between Multinode High Availability Nodes

In Multinode High Availability, two SRX Series Firewalls act as independent devices. These devices have unique hostname and the IP address on fxp0 interface. You can configure control plane stateless services such as ALG, firewall, NAT independently on these devices. Node-specific packets are always processed on the respective nodes.

Following packets/services are node-specific (local) in Multinode High Availability:

- Routing protocols packets to Routing Engine
- Management services, such as SNMP, and operational commands (`show`, `request`)
- Processes, such as the authentication service process (authd), integrated with RADIUS and LDAP servers
- ICL encryption specific tunnel control and data packets

The configuration synchronization in Multinode High Availability is not enabled by default. If you want certain configurations to synchronize to the other node, you need to:

- Configure the feature/function as part of `groups`
- Synchronize the configuration using the `[edit system commit peers-synchronize]` option

When you enable configuration synchronization (by using the `peers-synchronize` option) on both the devices in a Multinode High Availability, configuration settings you configure on one peer under [groups] will automatically sync to the other peer upon the **commit** action.

The local peer on which you enable the `peers-synchronize` statement copies and loads its configuration to the remote peer. Each peer then performs a syntax check on the configuration file being committed. If no errors are found, the configuration is activated and becomes the current operational configuration on both peers.

For configuration example, see "Example: Configure Multinode High Availability with Junos OS Configuration Groups " on page 912.

The following configuration snippet shows VPN configuration under `avpn_config_group` on host-mnha-01. We'll synchronize the configuration to the other peer device host-mnha-02.

1. Configure the hostname and IP address of the participating peer device (host-mnha-02), authentication details, and include the `peers-synchronization` statement.

```
On host-mnha-01
[edit]
set system commit peers-synchronize
set system commit peers host-mnha-02 user user-02
set system commit peers host-mnha-02 authentication "$ABC"
set system services netconf ssh
set system static-host-mapping host-mnha-02 inet 10.157.75.129
```

2. Configure the group (avpn_config_group) and specify apply conditions (when peers host-mnha-01 and host-mnha-02)

```
On host-mnha-01
 set groups avpn_config_group when peers host-mnha-01
 set groups avpn_config_group when peers host-mnha-02
 set groups avpn_config_group security ike proposal avpn_IKE_PROP authentication-method rsa-
signatures
 set groups avpn_config_group security ike proposal avpn_IKE_PROP dh-group group14
 set groups avpn_config_group security ike proposal avpn_IKE_PROP authentication-algorithm
sha1
 set groups avpn_config_group security ike proposal avpn_IKE_PROP encryption-algorithm
aes-128-cbc
 set groups avpn_config_group security ike proposal avpn_IKE_PROP lifetime-seconds 3600
 set groups avpn_config_group security ike policy avpn_IKE_POL proposals avpn_IKE_PROP
 set groups avpn_config_group security ike policy avpn_IKE_POL certificate local-certificate
crt2k
 set groups avpn_config_group security ike gateway avpn_ike_gw ike-policy avpn_IKE_POL
 set groups avpn_config_group security ike gateway avpn_ike_gw dynamic distinguished-name
wildcard C=us,O=ixia
 set groups avpn_config_group security ike gateway avpn_ike_gw dynamic ike-user-type group-
ike-id
 set groups avpn_config_group security ike gateway avpn_ike_gw dead-peer-detection probe-idle-
tunnel
 set groups avpn_config_group security ike gateway avpn_ike_gw dead-peer-detection interval 60
```

```
 set groups avpn_config_group security ike gateway avpn_ike_gw dead-peer-detection threshold 5
 set groups avpn_config_group security ike gateway avpn_ike_gw local-identity hostname
srx.juniper.net
 set groups avpn_config_group security ike gateway avpn_ike_gw external-interface lo0.0
 set groups avpn_config_group security ike gateway avpn_ike_gw local-address 10.11.0.1
 set groups avpn_config_group security ike gateway avpn_ike_gw version v2-only
 set groups avpn_config_group security ipsec proposal avpn_IPSEC_PROP protocol esp
 set groups avpn_config_group security ipsec proposal avpn_IPSEC_PROP authentication-
algorithm hmac-sha1-96
 set groups avpn_config_group security ipsec proposal avpn_IPSEC_PROP encryption-algorithm
aes-128-cbc
 set groups avpn_config_group security ipsec proposal avpn_IPSEC_PROP lifetime-seconds 1800
 set groups avpn_config_group security ipsec policy avpn_IPSEC_POL perfect-forward-secrecy
keys group14
 set groups avpn_config_group security ipsec policy avpn_IPSEC_POL proposals avpn_IPSEC_PROP
 set groups avpn_config_group security ipsec vpn avpn_ipsec_vpn bind-interface st0.15001
 set groups avpn_config_group security ipsec vpn avpn_ipsec_vpn ike gateway avpn_ike_gw
 set groups avpn_config_group security ipsec vpn avpn_ipsec_vpn ike ipsec-policy
avpn_IPSEC_POL
 set groups avpn_config_group security ipsec vpn avpn_ipsec_vpn traffic-selector ts local-ip
10.19.0.0/8
 set groups avpn_config_group security ipsec vpn avpn_ipsec_vpn traffic-selector ts remote-ip
10.4.0.0/8
 set groups avpn_config_group security zones security-zone vpn host-inbound-traffic system-
services all
 set groups avpn_config_group security zones security-zone vpn host-inbound-traffic protocols
all
 set groups avpn_config_group security zones security-zone vpn interfaces st0.15001
 set groups avpn_config_group interfaces st0 description vpn
 set groups avpn_config_group interfaces st0 unit 15001 family inet
```

3. Use the `apply-groups` command at the root of the configuration.

```
On host-mnha-01
 set apply-groups avpn_config_group
```

When you commit the configuration, Junos checks the command and merge the correct group to match the node name.

4. Verify the syncronization status using the `show configuration system` command from the operational mode.

```
user@host-mnha-01> show configuration system
...........
commit {
    peers {
        host-mnha-02  {
            user user user-02;
            authentication "$ABC123";
        }
    }
}

static-host-mapping {
    host-mnha-02 inet 10.157.75.129;
}
............
```

The command output displays the details of the peer SRX Series Firewall under the **peers** option.

NOTE: The configuration synchronization happens dynamically and if any configuration change done when only one node is available or when the connectivity is broken between the nodes, you must issue one more commit to synchronize the configuration to the other node. Otherwise, it will lead to inconsistent configurations across nodes for the applications.

NOTE:

- The configuration synchronization is not mandatory for Multinode High Availability to work. However, for an easy configuration synchronization, we recommend using the `set system commit peers-synchronize` statement with `junos groups` configuration in one direction (node 0 to node 1 for example).
- We recommend using out of band management (fxp0) connection to form configuration sync between Multinode High Availability nodes to manage common configurations.

- For IPsec use case, if configuration synchronization is not enabled, you must commit the configuration first on the backup node and then on the active node.

RELATED DOCUMENTATION

Two-Node Multinode High Availability | 573

Prepare Your Environment for Multinode High Availability Deployment | 620

Example: Configure Multinode High Availability in a Default Gateway Deployment | 743

Example: Configure Multinode High Availability in a Layer 3 Network | 698

Example: Configure Multinode High Availability in a Hybrid Deployment | 778

Selective Session Synchronization for Multinode High Availability

SUMMARY

Learn how selective session synchronization lets you control synchronization preferences in both two-node and four-node Multinode High Availability setups.

IN THIS SECTION

The multinode high availability (MNHA) feature enhances resilience and performance by ensuring the concurrent activity of both control and data planes across participating nodes. A four-node MNHA synchronizes flow sessions between the peer nodes through:

- **Cold Synchronization**—This happens when a new node joins the MNHA cluster. The system needs to synchronize all active flow session states from the existing nodes to the new one. This ensures the new node has the same session data as its peers, so it can take over traffic seamlessly if needed.

  Benefit: Reduce the full synchronization time, which means the new node can become active faster and start participating in load sharing or failover.

- **Hot Synchronization**—This occurs continuously during normal operation. Whenever a new session is created, its state is immediately synchronized to the peer node. This ensures real-time redundancy—if one node fails, the peer can take over without losing session data.

  Benefit: Improve CPS (connections per second) performance, meaning the system can handle more new sessions per second efficiently.

This dual approach optimizes system performance by reducing the need for repeated state replications, thereby enhancing session synchronization efficiency.

Selective session synchronization allows you to manage session synchronization preferences in a two-node MNHA and in a four-node MNHA using the following options:

- **Session synchronization based on policy and age**—This option allows you to disable synchronization for short-lived sessions or set a minimum age for session synchronization.
- **Default and user-defined profiles**—This option allows you to configure default flow profile or user-defined flow profiles for session synchronization. The default profile applies if no user-defined profile is set.

Benefits of Multinode High Availability (MNHA)

- Optimizes system performance by synchronizing sessions through cold and hot synchronization methods, reducing the need for repeated state replications.
- Enables fine-grained session management through customizable sync policies, durations, and profiles—supporting both default flow profile and user-defined flow profile configurations to meet specific network needs.

Configuring Selective Session Synchronization

To configure selective session synch, you need to define the following options (sessions synch based on session age or disable session synchronize) in the default flow profile or in a user defined flow profile.

- `session-sync disabled`: Disables synchronization of sessions over both inter domain link (IDL) and interchassis link (ICL). Use this option for certain policies such as short lived sessions for DNS, HTTP.
- `session-sync-min-age`: Synchronizes the sessions only after it is established for minimum session age duration. You can set the values between 0 to 3600 seconds. By default, the value is set to 0, meaning all sessions are synchronized immediately.

Configure Defalut Flow Profile

By default all the policies use the default-profile if none of the user defined profile is attached to policy. Default values for default flow profile are sync sessions immediately over intra domain (ICL) and inter domain (IDL) links.

```
[edit]
user@host# set security flow flow-profile default_profile session-sync-min-age <0-3600>
```

Or

```
[edit]
user@host# set security flow flow-profile default_profile session-sync disabled
```

Configure User-Defined Flow Profile

You can define a profile (user defined profile) and apply it in a security policy. If user defined profile is not attached to policy, then default profile will be applied in the security policy.

1. Create a new flow profile called "p1_profile" and define session synchronization options:
   Disable session synchronization or custom synchronization

   ```
   [edit]
   user@host# set security flow flow-profile p1_profile session-sync disabled
   ```

   Or

   ```
   [edit]
   user@host# set security flow flow-profile p1_profile session-sync-min-age 5
   ```

2. Apply the profile in a security policy:

   ```
   [edit]
   user@host# set security policies from-zone npw to-zone npw policy npw_1 match source-address
   any
   user@host# set security policies from-zone npw to-zone npw policy npw_1 match destination-
   address any
   user@host# set security policies from-zone npw to-zone npw policy npw_1 match application any
   ```

```
user@host# set security policies from-zone npw to-zone npw policy npw_1 then permit
user@host# set security policies from-zone npw to-zone npw policy npw_1 then permit flow-
profile p1_profile
```

NOTE: Profile switching behavior—When switching from one profile (such as `p1_profile`) to another (such as `p2_profile`), the new profile settings apply only to newly created sessions. Existing sessions continue to operate under the previously applied profile.

Default profile usage—System policies such as pre-id-default policy or default policy, or any other policy without an explicitly configured profile will automatically use the default profile settings.

To check the session-sync status on MNHA nodes, use the following commands:

- `show security flow session summary`

  Show flow session details, including the number of sessions synchronized to ICL and IDL when selective-session-sync options are applied.

```
user@host> show security flow session summary
Unicast-sessions: 16
Multicast-sessions: 0
Services-offload-sessions: 0
Failed-sessions: 0
Sessions-in-drop-flow: 0
Selective-sync-session: 13
Sessions-in-use: 16
    Valid sessions: 16
    Pending sessions: 0
    Invalidated sessions: 0
    Sessions in other states: 0
Maximum-sessions: 262144
```

- `show security flow session selective-session-sync-disabled`

  Show details of all sessions that were not synced to the peer node due to the session-sync disabled configuration.

```
user@host> show security flow session selective-sync-disabled
Session ID: 107, Policy name: in_policy/4, HA State: Active, Timeout: 108, Session State:
Valid
In: 3.0.0.1/56206 --> 6.0.0.1/22;tcp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 24, Bytes: 3749,
HA Wing State: Active,
```

```
Out: 6.0.0.1/22 --> 3.0.0.1/56206;tcp, Conn Tag: 0x0, If: ge-0/0/1.0, Pkts: 24, Bytes: 5117,
HA Wing State: Active,

Session ID: 108, Policy name: in_policy/4, HA State: Active, Timeout: 124, Session State:
Valid
In: 3.0.0.1/58442 --> 6.0.0.1/22;tcp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 24, Bytes: 3749,
HA Wing State: Active,
Out: 6.0.0.1/22 --> 3.0.0.1/58442;tcp, Conn Tag: 0x0, If: ge-0/0/1.0, Pkts: 24, Bytes: 5117,
HA Wing State: Active,
Total sessions: 2
```

IPsec VPN Support in Multinode High Availability

IN THIS SECTION

IPsec VPN in Active-Backup Mode

SRX Series Firewalls support IPsec VPN tunnels in a Multinode High Availability setup. Prior to Junos OS Release 22.4R1, IPsec VPN tunnel anchors at SRG1, where SRG1 acts in stateful active / backup mode. In this mode, all VPN tunnels terminate on the same device where the SRG1 is active.

Multinode High Availability establishes IPsec tunnel and performs key exchanges by:

- Dynamically associating the floating IP address of the active SRG1 for the termination IP in routing deployment and assigns the termination IP, the virtual IP(VIP), which floats between the two devices in switching mode.
- Generating the CA profile, when there is a need for a dynamic CA profile to authenticate the tunnel establishment, on the node where SRG1 is active.

- Performing new authentication and loading the dynamic profile on the newly active node and clearing 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.

Process Packets on Backup Node

When you use the `process-packet-on-backup` option in Multinode High Availability, the Packet Forward 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 in active mode; 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.

IPsec VPN in Active-Active Mode

Starting in Junos OS Release 22.4R1, you can configure Multinode High Availability to operate in active-active mode with support of multi SRG1s (SRG1+) for IPsec VPN. In this mode, some SRGs remain active

on one node and some SRGs remain active on another node. A particular SRG always operates in active-backup mode; it operates in active mode on one node and backup mode on another node.

Multinode High Availability supports IPsec VPN in active-active mode with multiple SRGs (SRG1+). In this mode, you can establish multiple active tunnels from both the nodes, based on SRG activeness. Since different SRGs can be active on different nodes, tunnels belonging to these SRGs come up on both nodes independently. Having active tunnels on both the nodes enables encrypting/decrypting data traffic on both the nodes resulting in efficient use of bandwidth.

Figure 55 on page 637 and Figure 56 on page 638 show differences in active-backup and active-active Multinode High Availabilty IPsec VPN tunnels.

Figure 55: Active-Backup IPsec VPN Tunnel in Multinode High Availability

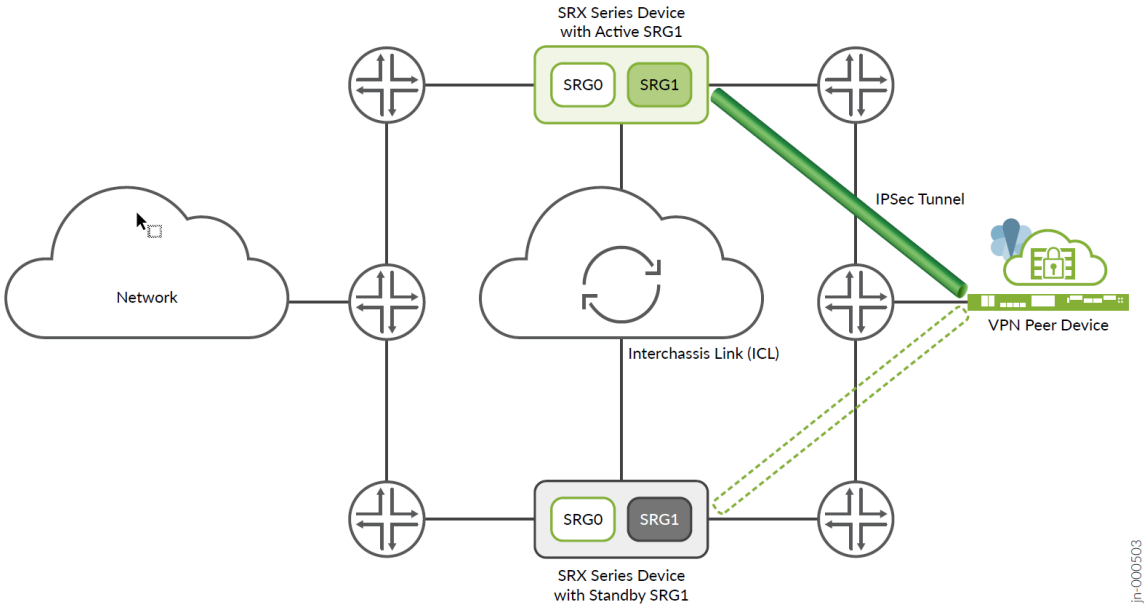
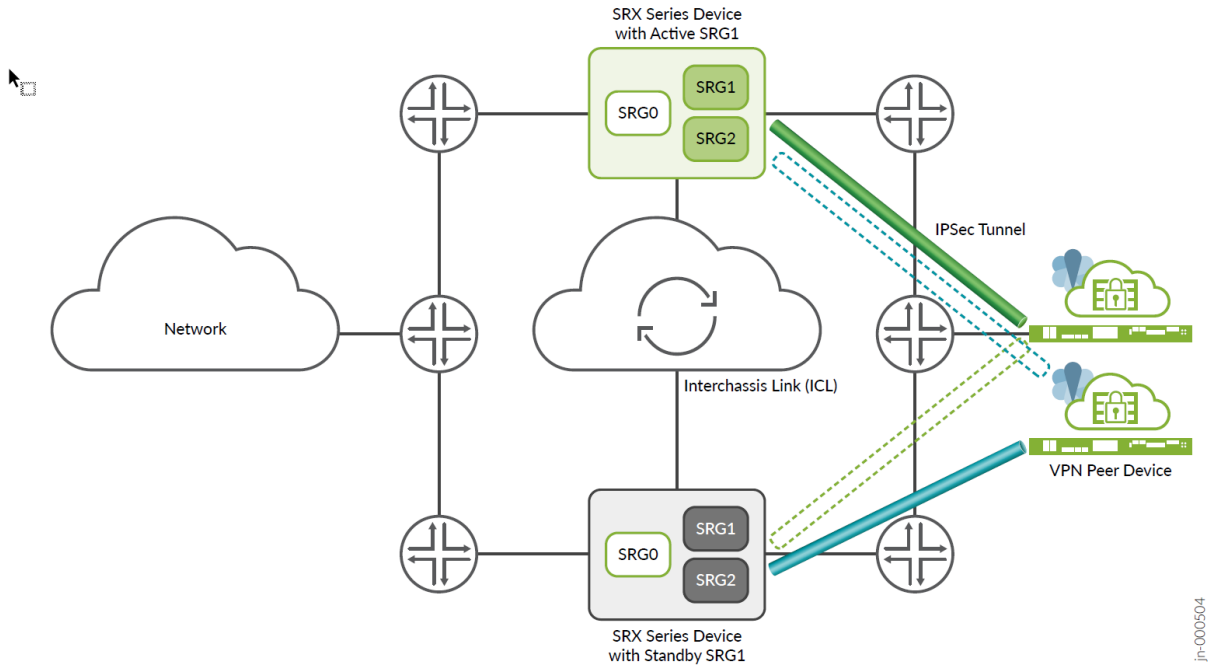


Figure 56: Active-Active IPsec VPN Tunnel in Multinode High Availability



Multinode High Availability establishes IPsec tunnel and performs key exchanges by associating termination IP address (which also identifies the tunnels ending on it) to the SRG. Since different SRG1+ can be in active state or in backup state on each of the devices, Multinode High Availability steers the matching traffic effectively to the corresponding active SRG1. Multinode High Availability also maintains the SRG ID and IP prefix mapping information.

Table 39 on page 638 and Table 40 on page 639 provide details on impact on IPsec VPN tunnels due to change in SRG1+ changes.

Table 39: Impact on IPsec VPN Tunnels Due to SRG1+ Modification

SRG1 Changes	Impact on IPSec VPN Tunnels
SRG addition	No impact on existing tunnels
SRG deletion	Deletes all routes associated with the SRG.
SRG attribute (other than prefix-list) modification	No impact on existing tunnels
SRG ID modification	Deletes all existing tunnels associated with the SRG.

Table 39: Impact on IPsec VPN Tunnels Due to SRG1+ Modification *(Continued)*

SRG1 Changes	Impact on IPSec VPN Tunnels
IP-prefix in prefix-list modification	Deletes all tunnels mapping to that particular IP prefix. No impact if there is no existing tunnel mapping to the modified IP prefix.

Table 40: Impact on IPsec VPN Tunnels Due to SRG1+ State Changes

SRG State Changes	Action from Multinode High Availability
`Active` to `Backup`	Deletes all data corresponding to that SRG, and resynchronizes from new the active SRG
`Active` to `Ineligible`	Deletes all data corresponding to that SRG, and resynchronizes from new the active SRG
`Active` to `Hold`	Not applicable
`Backup` to `Active`	No action
`Ineligible` to `Active`	No action
`Hold` to `Active`	No action
`Hold` to `Backup`	No action (possible state transition; if Active state is not involved in either pre or post state, no action is required)
`Ineligible` to `Backup`	No action (possible state transition; if Active state is not involved in either pre or post state, no action is required)
`Hold` to `Ineligible`	No action (possible state transition; if Active state is not involved in either pre or post state, no action is required)

Associate IPsec VPN Service to an SRG

Releases before 22.4R1 supported only SRG0 and SRG1, and SRG1 was associated to IPsec VPN by default. In 22.4R1, an SRG is not associated to the IPSec VPN service by default. You must associate the IPsec VPN service to any of the multiple SRGs by:

- Specifying IPsec as managed service

  Ex: `[set chassis high-availability services-redundancy-group <id> managed-services ipsec]`

- Creating an IP prefix list

  Ex: `[set chassis high-availability services-redundancy-group <id> prefix-list <name>]`

  `[set policy-options prefix-list <name> <IP address>`

When you have multiple SRGs in your Multinode High Availability setup, some SRGs are in active state on one node and some SRGs are active on another node. You can anchor certain IPsec tunnels to particular node (SRX Series firewall) by configuring an IP prefix list.

In IPsec VPN configuration, an IKE gateway initiates and terminates network connections between two security devices. The local end (local IKE gateway) is the SRX Series interface that initiates IKE negotiations. Local IKE gateway has a local IP address, a publicly routable IP address on the firewall, which the VPN connection uses as the endpoint.

IP prefix list includes a list of IPv4 or IPv6 address prefixes, which are used as local address of an IKE gateway. You can associate these IP prefixes (prefix-list) with a specified SRG1 to advertise local address of IKE gateway with a higher preference according to state of the SRG.

To anchor a certain IPsec VPN tunnel to a particular security device, then you must:

- Create an IP prefix list by including the local address of IKE gateway and associate the IP prefix list to the SRG:

  Example:

```
set chassis high-availability services-redundancy-group 1 prefix-list lo0_1
set chassis high-availability services-redundancy-group 2 prefix-list lo0_2
set policy-options prefix-list lo0_1 10.11.0.1/32
set policy-options prefix-list lo0_2 10.11.1.1/32
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 10.11.1.1/32
```

- Define the routing-instance for the prefix list.

```
set chassis high-availability services-redundancy-group 1 prefix-list lo0_1 routing-instance
rt-vr
set chassis high-availability services-redundancy-group 1 prefix-list lo0_2 routing-instance
rt-vr
```

  If you do not associate a routing-instance for the prefix-list, Multinode High Availability uses the default routing table, that might affect VPN functionality.

- Associate/enable IPsec VPN to the SRG.

  Example:

```
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 2 managed-services ipsec
```

  This configuration allows you to selectively and flexibly associate IPsec VPN to one of the multiple SRGs configured on SRX Series Firewall in a Multinode High Availability setup.

You can check the mapping of IKE/IPsec objects to the SRG by using the following command:

```
user@host# show chassis high-availability information detail
.........
Services Redundancy Group: 1
        Deployment Type: SWITCHING
        Status: BACKUP
        Activeness Priority: 200
        Hold Timer: 1
        Services: [ IPSEC ]
        Process Packet In Backup State: NO
        Control Plane State: NOT READY
        System Integrity Check: COMPLETE
        Peer Information:
        Failure Events: NONE
          Peer Id: 2
          Last Advertised HA Status: ACTIVE
          Last Advertised Health Status: HEALTHY
          Failover Readiness: N/A
```

```
.............
```

You can check the mapping of SRGs and IP prefix list by using the following command:

```
user@host> show chassis high-availability prefix-srgid-table
IP SRGID Table:
        SRGID    IP Prefix                                  Routing Table
        1        10.11.0.1/32                               rt-vr

        1        10.19.0.1/32                               rt-vr

        1        10.20.0.1/32                               rt-vr

        2        10.11.1.1/32                               rt-vr

        2        10.19.1.1/32                               rt-vr

        2        10.20.1.1/32                               rt-vr
```

If you do not configure a prefix list, you'll get the following warning message:

```
user@host> show chassis high-availability prefix-srgid-table
    Warning: prefix list not configured
```

See "Example: Configure IPSec VPN in Active-Active Multinode High Availability in a Layer 3 Network" on page 821 for details.

Dynamic Routing Protocol Support for IPsec VPN

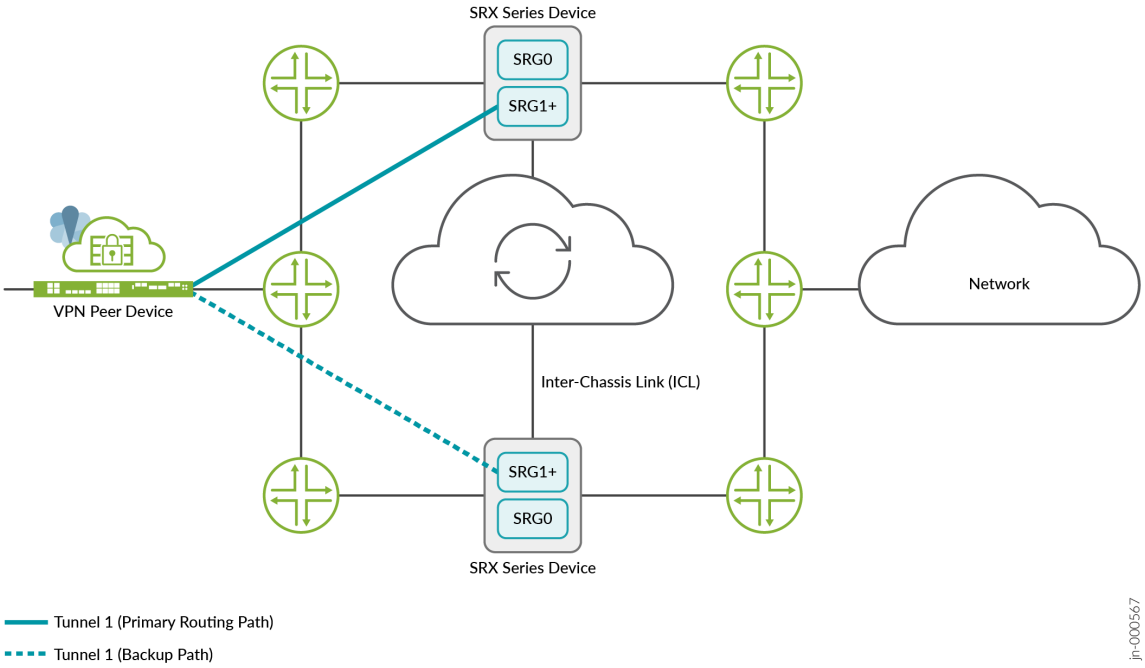
Starting in Junos OS Release 23.2R1, you can enable dynamic routing protocols for IPsec VPN in a Multinode High Availability setup by using node-local tunnels. The routes that the dynamic routing protocols add remain local to a node. These routes are not bound to any services redundancy group (SRG).

In the previous releases, Multinode High Availability supports only traffic selector deployment. That is, when you configure IPsec VPN by using traffic selectors, the configuration installs routes by considering the preference value and the routing metric based on traffic selector prefixes.

When you configure node-local tunnels, you have separate tunnels from a VPN peer device to both the nodes of the Multinode High Availability setup. That is—you have one node-local tunnel to each of the two Multinode High Availability nodes.

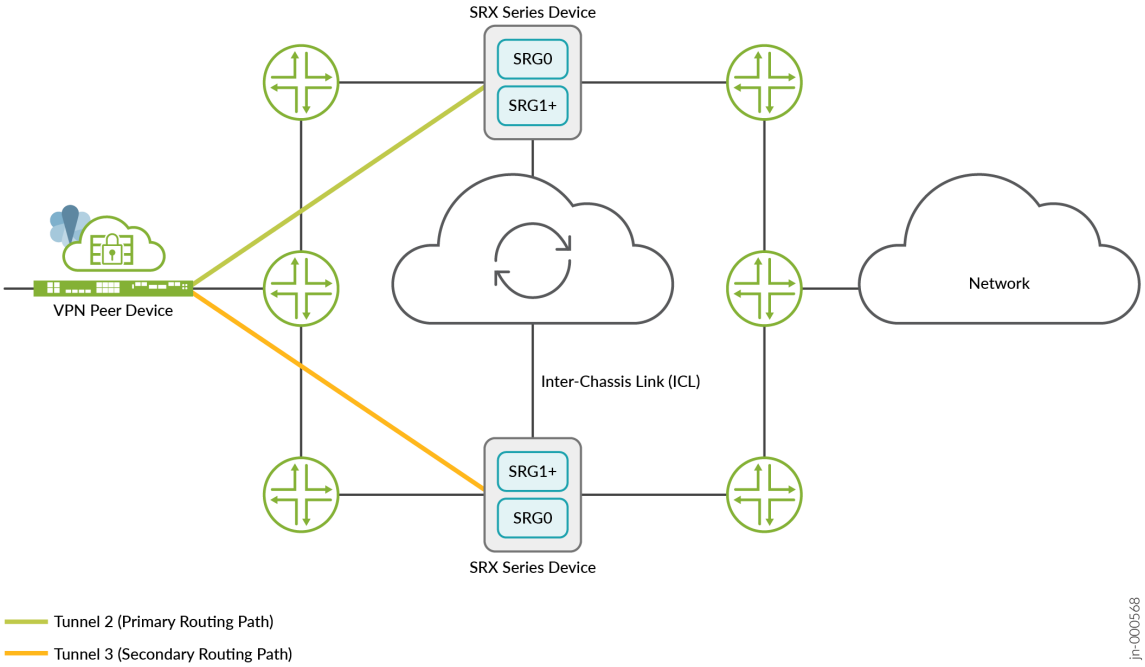
Figure 57 on page 643, Figure 58 on page 644, and Figure 59 on page 645 show a Multinode High Availability IPsec VPN deployment with synced tunnels, node-local tunnels, and a combination of synced tunnels and node-local tunnels, respectively.

Figure 57: Multinode High Availability Deployment with Synced Tunnels



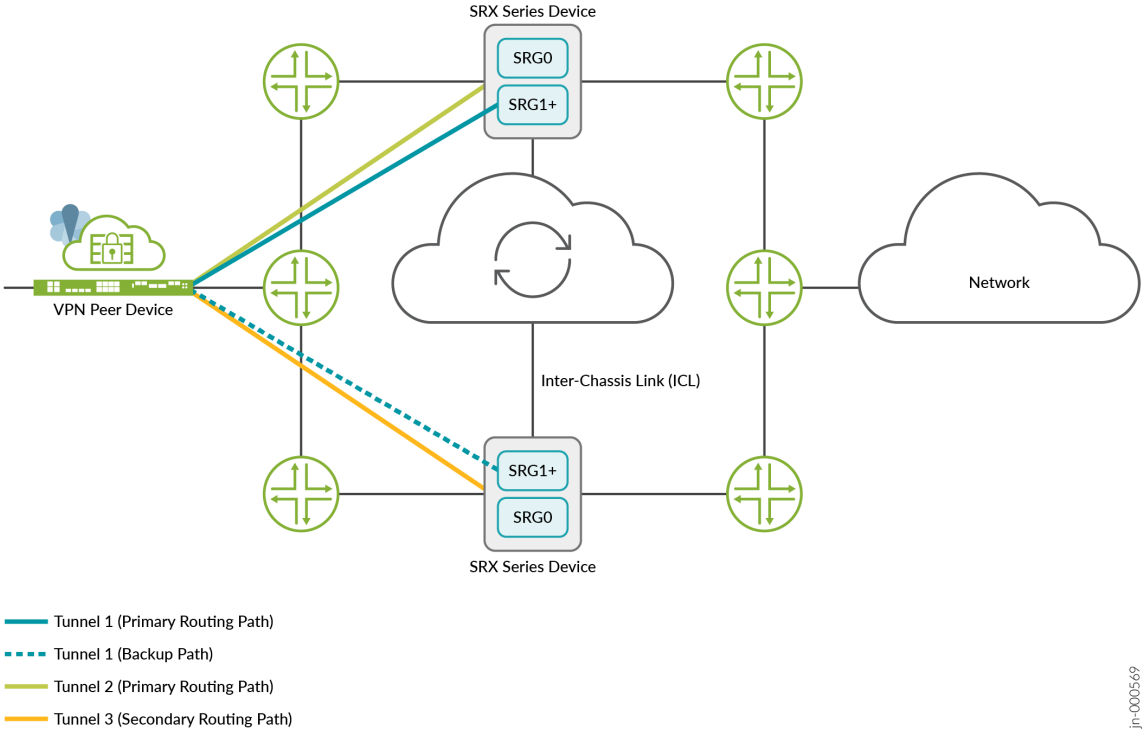
The preceding figure shows an IPsec VPN tunnel between a peer device and a Multinode High Availability setup. The IPsec VPN tunnel anchors at an active SRG1+. The tunnel remains active when the associated SRG1+ is active. In this deployment, traffic runs through the active tunnel (Tunnel 1).

Figure 58: Multinode High Availability Deployment with Node-Local Tunnel



In the preceding figure, you have two node-local tunnels between the VPN peer device and the Multinode High Availability setup. Each tunnel connects to one of the two nodes in the setup. These tunnels are not associated with any SRG1+. Either one or both the tunnels can remain active at any instant. Based on the configured routing protocol, at any instant, traffic runs either through Tunnel 2 or through Tunnel 3.

Figure 59: Multinode High Availability Deployment with Combination of Synced Tunnels and Node-Local Tunnels



The preceding figure shows an IPsec VPN tunnel between a VPN peer device and a Multinode High Availability setup. Additionally, the figure shows two node-local tunnels between the VPN peer device and the Multinode High Availability setup.

The IPsec VPN tunnel anchors at an active SRG1+ and remains active when the associated SRG1+ is active. In the case of node-local tunnels, both the tunnels remain active.

Table 41 on page 645 shows the difference between `node-local` tunnels and synced tunnels.

Table 41: Difference Between Node-Local Tunnels and Synced Tunnels

Functions	Node-Local Tunnels	Synced Tunnels
Association with SRG1+	No	Yes
Tunnel information synchronization between Multinode High Availability nodes	No	Yes

Table 41: Difference Between Node-Local Tunnels and Synced Tunnels *(Continued)*

Functions	Node-Local Tunnels	Synced Tunnels
Number of active tunnels	Two	One

Mark an IPsec VPN Tunnel as Node-Local Tunnel

You can configure an IPsec VPN tunnel as `node-local` on an SRX Series Firewall by using the following statement:

```
[edit]
user@host# set security ike gateway gateway-name node-local
```

Ensure that you configure the `node-local` option for both the nodes in a Multinode High Availability setup.

Ensure that you set a preference for one tunnel when you configure the routing policy.

ADVPN Support in Multinode High Availability

Starting in Junos OS Release 24.2R1, Multinode High Availability support ADVPN in node-local tunnel deployment.

Node-local tunnels enhance Multinode HA by providing separate tunnels from a VPN peer device to both nodes in the setup. ADVPN allows VPN tunnels to be established dynamically between spokes. Combining ADVPN with Multinode HA in node-local tunnels deployment ensures robust network connectivity, efficient resource utilization, and seamless failover.

The ADVPN protocol allows creation of shortcut path between two partners gateways to establish an optimal path for data delivery. Traditionally, in a hub-and-spoke network, traffic between two spokes traverses through the hub. With ADVPN, the hub recommends a shortcut between its peers (spoke devices) with which it has previously established an IPsec SA. The decision to suggest a shortcut depends on the duration and amount of traffic flowing between a pair of peers through the hub. These peers, called as the shortcut partners, accept or decline this recommendation, according to their own policies.

The peers accept the suggestion and establish a direct SA (shortcut) between them. A new phase1 and phase2 SA is created for each shortcut. This shortcut is then used to establish a more optimal path for data delivery. All traffic flowing between the peers now goes directly over the shortcut tunnel between the peers.

If the peers decline the recommendation, they respond back to the suggester indicating the reason for rejection. In this case the traffic continues to flow through the Shortcut Suggester.

The Multinode High Availability setup includes two SRX Series Firewalls acting as active node and backup node and two VPN peer devices with node-local configuration. In this case, an IPsec VPN tunnel is establishes between a VPN peer device and a Multinode High Availability setup.

- Shortcut suggester: Notices traffic moving between peers and suggests shortcuts.
- Shortcut partners: These are peer devices that form the shortcut tunnel. The shortcut exchange happens through an extended IKEv2 protocol.

In Multinode High Availability setup, a VPN peer device, with node-local tunnel, takes up the following roles:

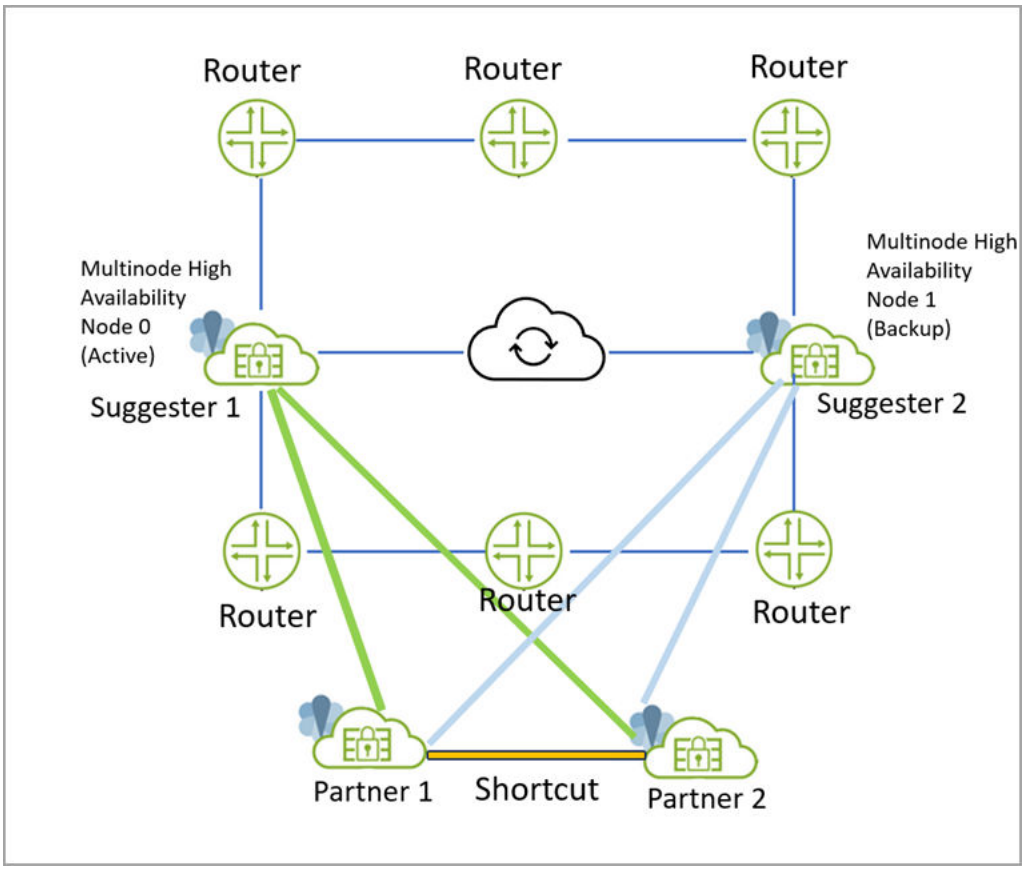
- ADVPN shortcut partner
- ADVPN shortcut suggester

With ADVPN configuration, an SRX Series Firewall can act as either a shortcut suggester or a shortcut partner, but not as both suggester and partner at a time.

The following images illustrate how VPN gateway can act as a shortcut suggester and partner.

- VPN peer device, acting as shortcut partner, establishes two tunnels– one tunnel towards each Multinode High Availability node. In this case, each node acts as an shortcut suggester.

Figure 60: VPN Gateway as Shortcut Partner

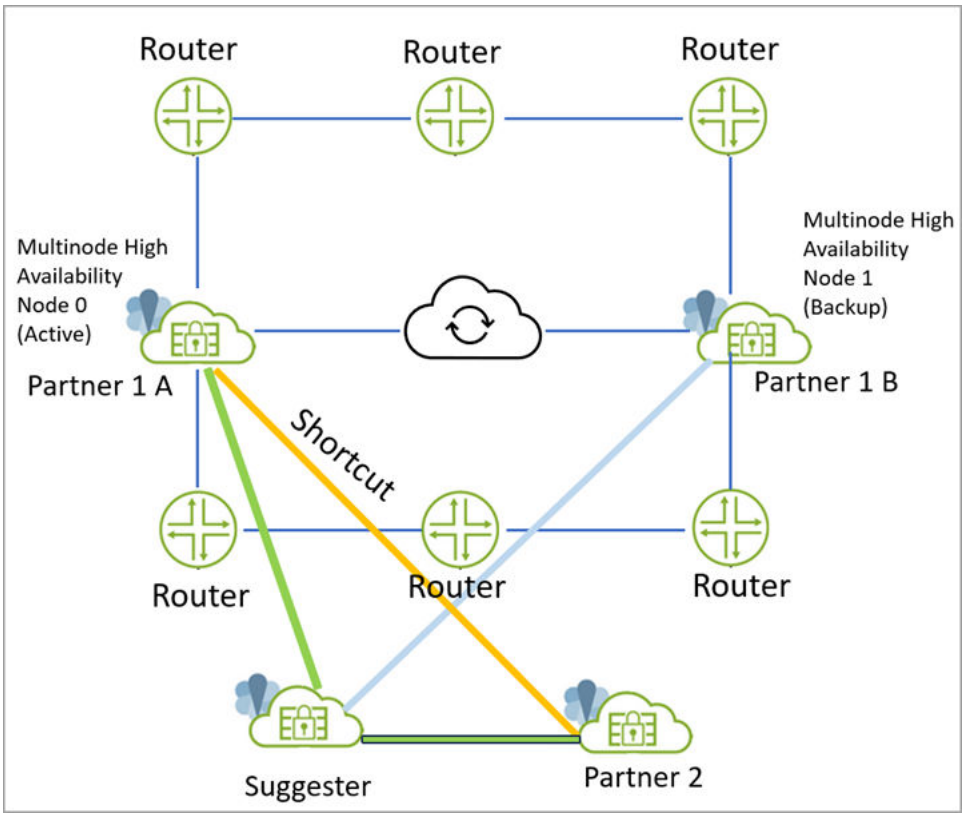


As shown in the illustration:

- Two SRX Series Firewalls in Multinode High Availability are acting as shortcut suggester—Suggester-1 (active node) and Suggester-2 (backup node).
- Two VPN Gateways are acting as shortcut partners—Partner-1 and Partner-2.
- The Partner-1 creates two tunnels; one towards Suggester-1 and another towards Sugester-2
- The Partner-2 creates two tunnels; one towards Suggester-1 and another towards Sugester-2
- Traffic from Partner-1 to Partner-2 goes through Suggester-1. Suggester-1 notifies Partner-1 and Partner-2 to create shortcut.
- In case if active node (suggester-1) fails and both tunnels from Suggester-1 to both Partner-1 and Partner-2 are down, in that case:
- - The shortcut created earlier remains active, however, the traffic flow from Partner-1 to Partner-2 passes through Suggester-2. In this case, Suggester-2 suggests shortcut between Partner-1 and Partner-2. Since the shortcut already exists between Partner-1 and Partner-2, the suggestion for the new shortcut Suggester-2 gets rejected.

- VPN peer device acting as shortcut suggester. In this case, each Multinode High Availability node acts as shortcut partner and establishes a separate tunnel towards VPN peer device.

Figure 61: VPN Gateway as Shortcut Suggester



- SRX Series Firewalls in Multinode High Availability are acting as Partner-1-A and Partner-1-B.
- One VPN gateway is acting as Suggester and another SRX Series Firewall is acting as Partner-2.
- Partner-1-A (active node) creates static tunnel (active tunnel) with Suggester.
- Partner-1-B (backup node) creates static tunnel (backup tunnel) with Suggester.
- Traffic from Partner-1-A to Partner-2 flows through Suggester.
- Suggester suggests creating shortcut between Partner-1-A and Partner-2.
- Partner-1-A and Partner-2 create a shortcut between them. Note that if the static tunnel between partners and suggester goes down, there is no impact on Shortcut tunnel. Traffic continues to flow through the Shortcut tunnel even after static tunnel goes down.

Configuration Highlights

1. Configure Multinode High Availability on SRX Series Firewalls. See Example: Configure Multinode High Availability in a Layer 3 Network.
2. Ensure that you configure the `node-local` option for both the nodes in a Multinode High Availability setup. Example:

```
set security ike gateway gateway-name node-local
```

3. Configure shortcut partner or shortcut suggester roles on SRX Series Firewall as applicable. See Auto Discovery VPNs.

By default both shortcut suggester and shortcut partner options are enabled if you configure **advpn** under IKE gateway hierarchy. You must explicitly disable suggester option or partner option to disable that particular functionality.

```
[edit security ike]
gateway gateway_1 {
    …
    node-local
    …
    advpn {
        partner disable;
    }
}
```

```
[edit security ike]
gateway gateway_1 {
    advpn {
        suggester disable;

        partner {
            connection-limit 5;
            idle-time 300;
        }
    }
}
```

Limitations

- Configuring an ADVPN suggester is only allowed on AutoVPN hubs where as partner functionality for spoke configuration.
- You cannot configure both suggester and partner roles under the same IKE gateway
- ADVPN does not support IKEv1
- You cannot create a shortcut between partners that are both behind NAT devices.

RELATED DOCUMENTATION

Example: Configure IPSec VPN in Active-Active Multinode High Availability in a Layer 3 Network | 821

Two-Node Multinode High Availability | 573

Example: Configure IPSec VPN in Active-Active Multinode High Availability in a Layer 3 Network | 821

Asymmetric Traffic Flow Support in Multinode High Availability

IN THIS SECTION

Overview

IN THIS SECTION

For stateful services or to perform deep packet inspection, a firewall requires to see both directions of each flow session. Asymmetric traffic flow happen when the flow of packets traverses from a source network to a destination network using one path (through node 1) and takes a different return path (using node 2). This asymmetric flow can occur when traffic flows across a Layer-3 routed networks.

In a typical high availability deployment, you have multiple routers and switches on the both sides of the network. The routers use a next-hop path to forward each packet flow; but routers might not use the same path for the return traffic. In a Multinode High Availability setup, routers send packets to the firewall based on current routing path, which can result in asymmetric traffic flows

This different handling of traffic directions can cause some packets to get dropped by one or both high availability nodes. This happens because neither node can capture the entire traffic flow, leading to potential inconsistencies and dropped packets.

To handle asymmetric traffic flows, the Multinode High Availability requires an additional link known as Inter Chassis Datapath (ICD). ICD can to route the traffic between two nodes. The ICD enables the nodes to redirect asymmetric traffic flows to the peer node that is originally in charge of providing stateful services for the flows.

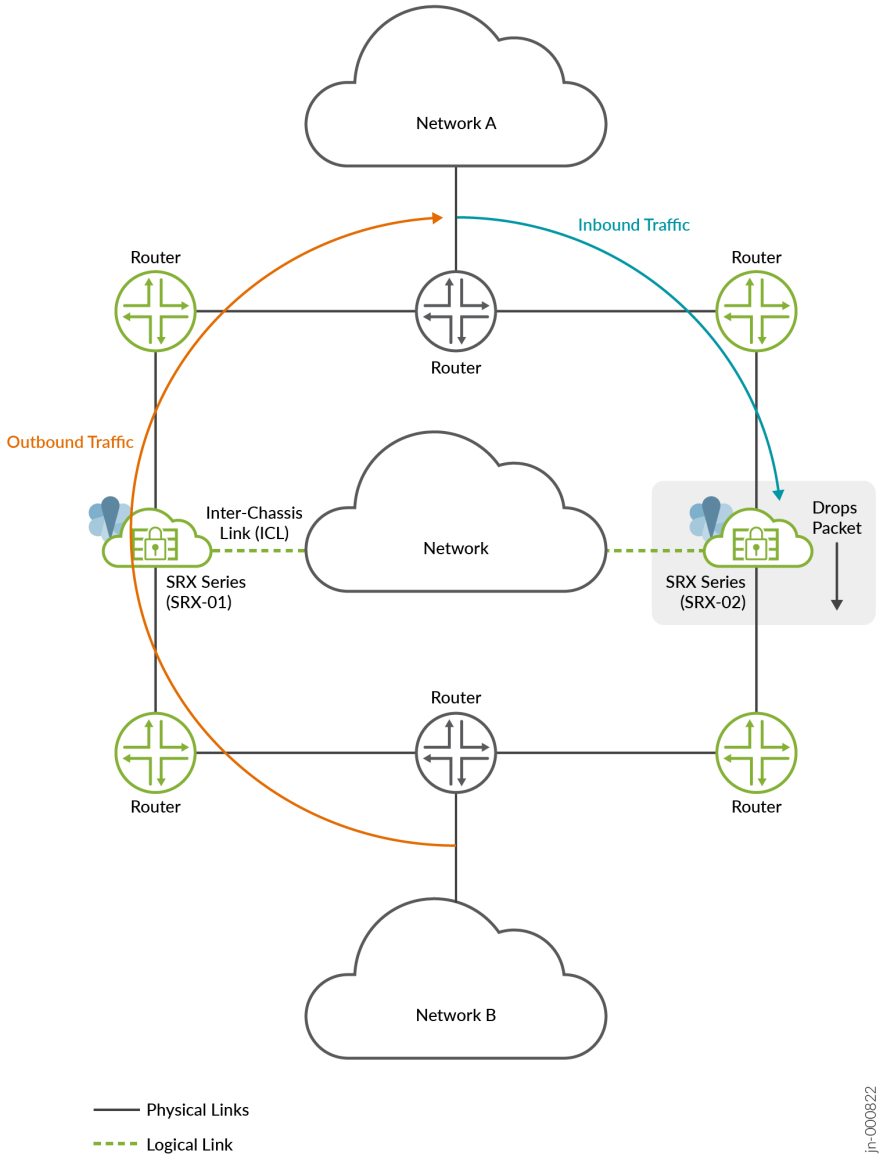
This feature ensures that security checks (such as three-way handshake and sequence check with window scale factor) can be performed for asymmetric traffic flows vs traditional (mandator) symmetric flows.

How Multinode High Availability Supports Asymmetric Traffic Flow

Without Asymmetric Traffic Flow Support

The bidirectional packets of the same flow are delivered to a different SRX Series device in Multinode High Availability setup by neighboring routers or switches as shown Figure 62 on page 653

Figure 62: Packet Flow without Asymmetric Traffic Flow Support



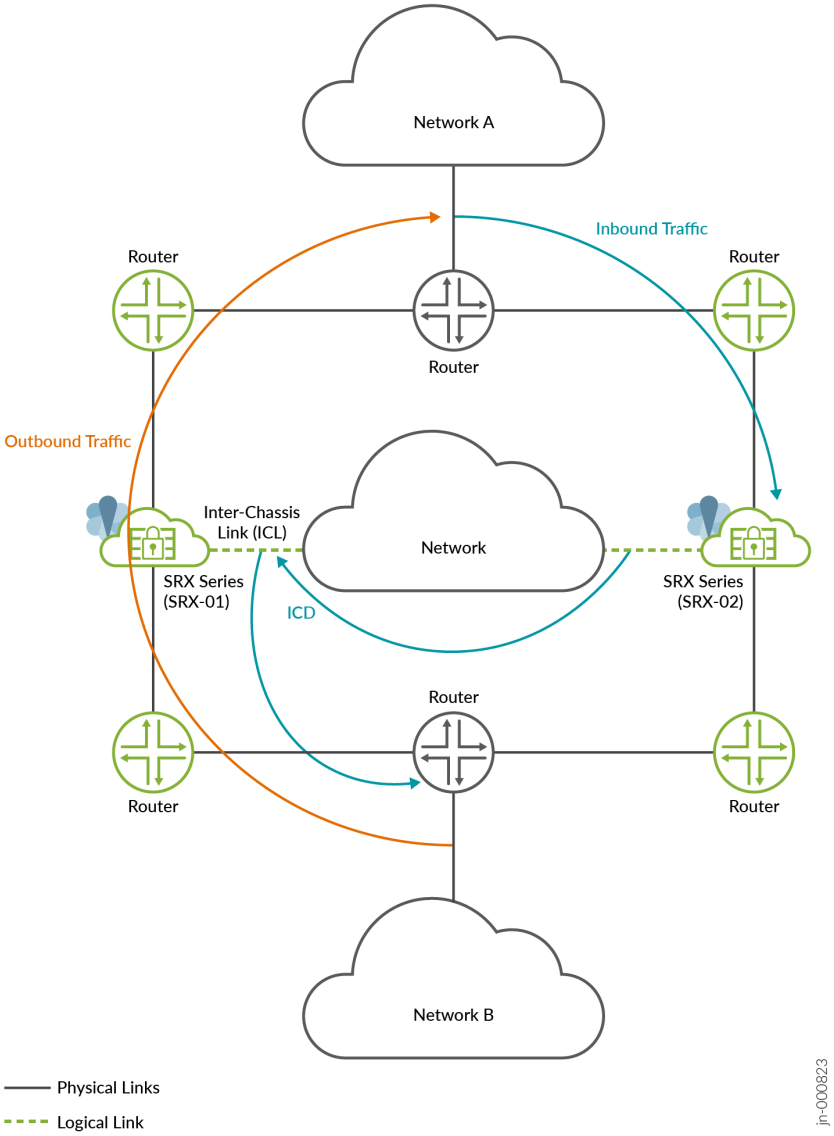
Outbound traffic from network B to network A passes through node 1 (SRX-01) and return traffic (inbound traffic) flows from network A to network B through node 2 (SRX-02).

In this case of asymmetric traffic flow, due to lack of complete state information about bidirectional traffic of the same flow, SRX Series Firewall (in this example SRX-02) drops packets.

With Asymmetric Traffic Flow Support

To support asymmetric traffic flow, Multinode High Availability uses Inter Chassis Datapath (ICD). The ICD forwards packets of asymmetric traffic flows between two SRX Series devices in a high availability setup.

Figure 63: Packet Flow with Asymmetric Traffic Flow Support



In this case, the Multinode High Availability system creates a new routable link between the nodes. This routable link enables the nodes to forward asymmetrical flows to the original node that can perform security inspection for the flow. That is, the node 2 (SRX-02) forwards the inbound traffic to the node 1 (SRX-01) instead to a next-hop router. The SRX Series Firewall performs security inspection for the packets of the bidirectional flow.

How Inter Chassis Datapath (ICD) Works?

Multinode High Availability ICD carries data traffic and forwards the data flows to the peer node. This link does not forward interchassis link (ICL) packets.

The workflow includes the following steps:

1. When a Multinode High Availability node receives a data packet, security services running on the node determines whether to forward the packet to the peer node or process it locally. The decision to forward the packet depends on:
   - Packet's flow session states or service type
   - State of SRG associated with the flow of the packet
2. If the peer node is reachable over the ICD, security services on a node can send and receive packets between the nodes.
3. Once the peer node receives a forwarded data packet through the ICD, it performs security inspection based on the configured policies.

To use ICD for packet forwarding between nodes, you must:

- Assign ICD to the loopback interface with a routable path to the other node.
- Ensure that the ICD has path diversity for the highest reliability by assigning multiple physical interfaces to the ICD.

Planning Interfaces for ICL and ICD

In a Multinode High Availability configuration, the ICL and ICD physical interfaces must be active and operational to accommodate asymmetric traffic flows. The ICL and ICD interfaces facilitate communication between nodes in the high availability setup, and their status will influence the packet processing. If either interface is non-functional, it will affect support for asymmetric traffic flows. Therefore, it is crucial to ensure the proper functioning of these interfaces for optimal network performance.

When you have multiple physical interfaces connected to the ICL, and one of these interfaces that's being actively used to process packets fails, the data flow switches to use another available physical interface associated with ICL. If all physical interfaces associated with ICL are down, SRX Series Firewalls lose the ICL connection. In this case, the SRX Series nodes can not exchange RTO messages and can not support asymmetric traffic flows.

Use different loopback interfaces for ICL and for ICD in a Multinode High Availability setup.

Nodes learn the route to reach IP address of the peer node's ICD through static or dynamic routing protocols (Example: BGP). Multinode High Availability setup leverages existing routing functionality on each SRX Series Firewall to route the packets.

ICL and ICD States Affecting Asymmetric Traffic

Table 42 on page 656 shows how the states of BFD between the nodes are dependent on assigned physical interfaces of both ICL and ICD.

Table 42: ICL and ICD States Affecting Asymmetric Traffic Flow Support

ICL		ICD		Service for Asymmetric Traffic Flow
Physical Interface	**BFD State**	**Physical interface**	**BFD State**	
Up	Up	Up	Up	Up
Up	Up	Down	Down	Down
Down	Down	Up	Up	Down
Up	Down	Up	Down	Down
Down	Down	Down	Down	Down

Configure Asymmetric Traffic Flow Support in Multinode High Availability

SUMMARY

Read this topic to understand how to configure asymmetric traffic flow support for SRX Series Firewalls deployed in the Multinode High Availability solution. The example covers configuration in active/backup mode when SRX Series Firewalls are connected to routers on both sides (Layer 3 deployment).

IN THIS SECTION

Junos OS Release 23.4R1 brings in a new feature that supports asymmetric traffic flow. Asymmetric routing is a scenario where the path of packets in one direction is different from the origin path.

In a typical high availability deployment, you have multiple routers and switches on the both sides of the network. The routers use a next-hop path to forward each packet flow; but routers might not use the same path for the return traffic. In a Multinode High Availability setup, routers send packets to the firewall based on current routing path, which can result in asymmetric traffic flows

To handle asymmetric traffic flows, the Multinode High Availability infrastructure employs a new link known as Inter Chassis Datapath (ICD). ICD has the ability to forward the traffic between two nodes. It enables the nodes to redirect asymmetric traffic flows to the peer node that is originally in charge of providing stateful services for these flows.

Follow this configuration example to set up Multinode High Availability to support asymmetric routing and to validate the configuration on your device.

TIP:

Table 43: Time Estimates

Reading Time	Less than 15 minutes.
Configuration Time	Less than an hour.

Example Prerequisites

Table 44 on page 657 lists the hardware and software components that support the configuration.

Table 44: Requirements

Supported Hardware	• SRX5800, SRX5600, SRX5400 with SPC3, IOC3, IOC4, SCB3, SCB4, and RE3 • SRX4600, SRX4300, SRX4200, SRX4120, SRX4100, SRX2300, SRX1600, and SRX1500
Supported Software	Junos OS Release 23.4R1
Licensing requirements	No separate license is required to configure Multinode High Availability. 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.

In this example, we've used two supported SRX Series Firewalls with Junos OS Release 23.4R1 and two Juniper Networks(R) MX960 Universal Routing Platform as upstream and downstream routers.

Before You Begin

Benefits	The SRX Series Firewall in a Multinode High Availability handles asymmetrically routed packets efficiently. This process ensures reliable and consistent handling of stateful services for these packets, improving overall performance and minimizing packet loss and inconsistencies in the network.
Know more	Multinode High Availability

Functional Overview

Table 45 on page 658 provides a quick summary of the configuration components deployed in this example.

Table 45: Configuration Components

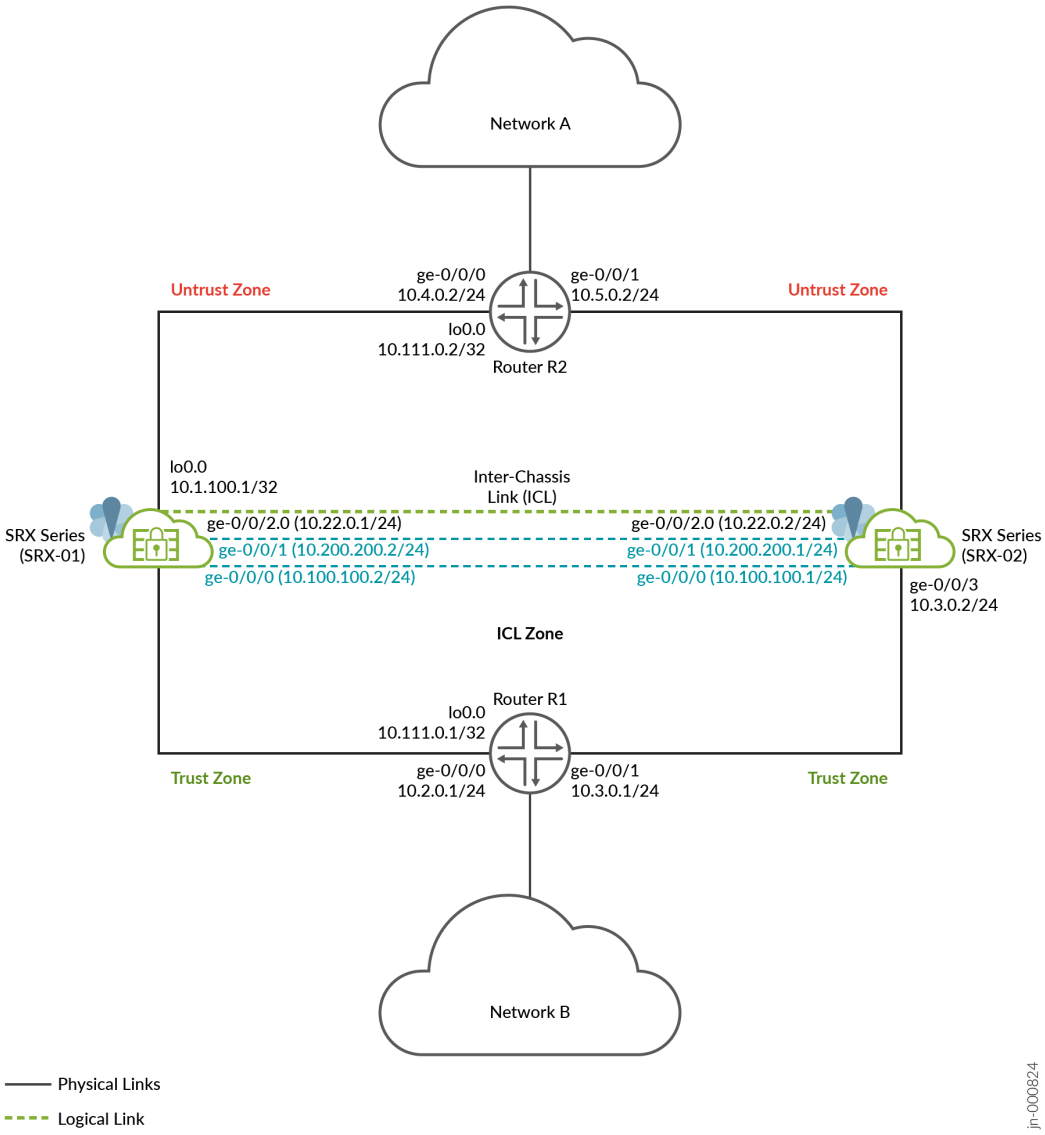
Technologies used	• High availability • Routing policy • Routing options

Primary verification tasks	1. Verify the high availability on both the nodes in the setup. 2. Verify the Multinode High Availability data plane statistics.

Topology Illustration

Figure 64 on page 659 shows the topology used in this example.

Figure 64: Multinode High Availability in Layer 3 Network with Interchassis Datapath (ICD)



As shown in the topology, two SRX Series Firewalls are connected to adjacent routers on trust and untrust side forming a BGP neighborship.

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. In general, you can use aggregated Ethernet (AE) or a revenue Ethernet port on the SRX Series Firewalls 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.

Two physical links (ICD) connect two SRX Series Firewalls. The physical interfaces on both the nodes are forming the MNHA ICD connections. In this example, use two dedicated revenue interfaces to configure ICD.

Loopback interfaces are used to host the IP addresses on SRX Series and routers.

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 Firewalls.

Topology Overview

In this example, you'll establish high availability between the SRX Series Firewalls and establish ICD (interchassis datapath) for providing support to handle asymmetric routing support.

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 Firewalls.

Table 46 on page 660 and Table 47 on page 661 show the details on interfaces configuration used in this example.

Table 46: Interfaces and IP Address Configuration on Security Devices

Device	Interface	Zone	IP Address	Configured For
SRX-01	lo0	Trust	10.1.100.1/32	Local forwarding address used to forward data packet over ICD link.
	ge-0/0/2	ICL-Zone	10.22.0.1/24	Interchassis link (ICL)
	ge-0/0/1 and ge-0/0/0	Trust	• 10.200.200.2/24 • 10.100.100.2/24	Interchassis Datalink connecting two SRX Series Firewalls

Table 46: Interfaces and IP Address Configuration on Security Devices *(Continued)*

Device	Interface	Zone	IP Address	Configured For
	ge-0/0/4	Untrust	10.4.0.1/24	Connects to R2 router
	ge-0/0/3	Trust	10.2.0.2/24	Connects to R1 router
SRX-02	lo0	Trust	10.1.200.1/32	Local forwarding address used to forward data packet over ICD link.
	ge-0/0/2	ICL-zone	10.22.0.2/24	Interchassis link (ICL)
	• ge-0/0/0 • ge-0/0/1	Trust	• 10.100.100.1/24 • 10.200.200.1/24	Interchassis Data link (ICD)
	ge-0/0/3	Trust	10.3.0.2/24	Connects to R1 router
	ge-0/0/4	Untrust	10.5.0.1/24	Connects to R2 router

Interfaces and IP Address Configuration on Routing Devices

Table 47: Interfaces and IP Address Configuration on Routing Devices

Device	Interface	IP Address	Configured for
R2	lo0	10.111.0.2/32	Loopback interface address of R2
	ge-0/0/0	10.4.0.2/24	Connects to SRX-02
	ge-0/0/1	10.5.0.2/24	Connects to SRX-01
	ge-0/0/2	10.6.0.1/24	Connects to external network
R1	lo0	10.111.0.1/32	Loopback interface address of R1
	ge-0/0/0	10.2.0.1/24	Connects to SRX-01
	ge-0/0/1	10.3.0.1/24	Connects to SRX-02
	ge-0/0/2	10.1.0.1/24	Connects to internal network

Configuration

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

- "Set Commands on All Devices" on page 679
- "Show Configuration Output " on page 686

Junos IKE package is required on your SRX Series Firewalls for Multinode High Availability configuration. This package is available as a default package or as an optional package on SRX Series Firewalls. See Support for Junos IKE Package for details.

If the package is not installed by default on your SRX Series firewall, use the **request system software add optional://junos-ike.tgz** to install it. You require this step for ICL encryption.

1. Configure interfaces.

   a. Configure interface used to connect trust network.

      SRX-01

```
[edit]
user@srx-01# set interfaces ge-0/0/3 description trust
user@srx-01# set interfaces ge-0/0/3 unit 0 family inet address 10.2.0.2/24
```

      SRX-02

```
[edit]
user@srx-02# set interfaces ge-0/0/3 description trust
user@srx-02# set interfaces ge-0/0/3 unit 0 family inet address 10.3.0.2/24
```

   b. Configure interface used to connect untrust network.

      SRX-01

```
[edit]
user@srx-01# set interfaces ge-0/0/4 description untrust
user@srx-01# set interfaces ge-0/0/4 unit 0 family inet address 10.4.0.1/24
```

SRX-02

```
[edit]
user@srx-02# set interfaces ge-0/0/4 description untrust
user@srx-02# set interfaces ge-0/0/4 unit 0 family inet address 10.5.0.1/24
```

c. Configure interface for ICL.

SRX-01

```
[edit]
user@srx-01# set interfaces ge-0/0/2 description interchassis_link
user@srx-01# set interfaces ge-0/0/2 unit 0 family inet address 10.22.0.1/24
```

SRX-02

```
[edit]
user@srx-02# set interfaces ge-0/0/2 description interchassis_link
user@srx-02# set interfaces ge-0/0/2 unit 0 family inet address 10.22.0.2/24
```

d. Configure interface for ICD.

SRX-01

```
[edit]
user@srx-01# set interfaces ge-0/0/0 description icd-1
user@srx-01# set interfaces ge-0/0/0 unit 0 family inet address 10.100.100.2/24
user@srx-01# set interfaces ge-0/0/1 description icd-2
user@srx-01# set interfaces ge-0/0/1 unit 0 family inet address 10.200.200.2/24
```

SRX-02

```
[edit]
user@srx-02# set interfaces ge-0/0/0 description icd-1
user@srx-02# set interfaces ge-0/0/0 unit 0 family inet address 10.100.100.1/24
user@srx-02# set interfaces ge-0/0/1 description icd-2
user@srx-02# set interfaces ge-0/0/1 unit 0 family inet address 10.200.200.1/24
```

e. Configure loopback interface.

SRX-01

```
[edit]
user@srx-01# set interfaces lo0 description trust
user@srx-01# set interfaces lo0 unit 0 family inet address 10.1.100.1/32
```

SRX-02

```
[edit]
user@srx-02# set interfaces lo0 description trust
user@srx-02# set interfaces lo0 unit 0 family inet address 10.1.200.1/32
```

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

   a. Configure trust zone.

   SRX-01

```
[edit]
user@srx-01# set security zones security-zone trust host-inbound-traffic system-services
all
user@srx-01# set security zones security-zone trust host-inbound-traffic protocols all
user@srx-01# set security zones security-zone trust interfaces ge-0/0/3.0
user@srx-01# set security zones security-zone trust interfaces ge-0/0/1.0
user@srx-01# set security zones security-zone trust interfaces ge-0/0/0.0
user@srx-01# set security zones security-zone trust interfaces lo0.0
```

   SRX-02

```
[edit]
user@srx-02# set security zones security-zone trust host-inbound-traffic system-services
all
user@srx-02# set security zones security-zone trust host-inbound-traffic protocols all
user@srx-02# set security zones security-zone trust interfaces ge-0/0/3.0
user@srx-02# set security zones security-zone trust interfaces lo0.0
user@srx-02# set security zones security-zone trust interfaces ge-0/0/1.0
user@srx-02# set security zones security-zone trust interfaces ge-0/0/0.0
```

   b. Configure untrust zone.

SRX-01

```
[edit]
user@srx-01# set security zones security-zone untrust host-inbound-traffic system-services
ike
user@srx-01# set security zones security-zone untrust host-inbound-traffic system-services
ping
user@srx-01# set security zones security-zone untrust host-inbound-traffic protocols bfd
user@srx-01# set security zones security-zone untrust host-inbound-traffic protocols bgp
user@srx-01# set security zones security-zone untrust interfaces ge-0/0/4.0
```

SRX-02

```
[edit]
user@srx-02# set security zones security-zone untrust host-inbound-traffic system-services
ike
user@srx-02# set security zones security-zone untrust host-inbound-traffic system-services
ping
user@srx-02# set security zones security-zone untrust host-inbound-traffic protocols bfd
user@srx-02# set security zones security-zone untrust host-inbound-traffic protocols bgp
user@srx-02# set security zones security-zone untrust interfaces ge-0/0/4.0
```

c. Configure ICL zone.

SRX-01

```
[edit]
user@srx-01# set security zones security-zone icl-zone host-inbound-traffic system-
services ike
user@srx-01# set security zones security-zone icl-zone host-inbound-traffic system-
services ping
user@srx-01# set security zones security-zone icl-zone host-inbound-traffic system-
services high-availability
user@srx-01# set security zones security-zone icl-zone host-inbound-traffic system-
services ssh
user@srx-01# set security zones security-zone icl-zone host-inbound-traffic protocols bfd
user@srx-01# set security zones security-zone icl-zone host-inbound-traffic protocols bgp
user@srx-01# set security zones security-zone icl-zone interfaces ge-0/0/2.0
```

SRX-02

```
[edit]
user@srx-02# set security zones security-zone icl-zone host-inbound-traffic system-
services ike
user@srx-02# set security zones security-zone icl-zone host-inbound-traffic system-
services ping
user@srx-02# set security zones security-zone icl-zone host-inbound-traffic system-
services high-availability
user@srx-02# set security zones security-zone icl-zone host-inbound-traffic system-
services ssh
user@srx-02# set security zones security-zone icl-zone host-inbound-traffic protocols bfd
user@srx-02# set security zones security-zone icl-zone host-inbound-traffic protocols bgp
user@srx-02# set security zones security-zone icl-zone interfaces ge-0/0/2.0
```

3. Configure Multinode High Availability local node.

   a. Configure local node.

      SRX-01

      ```
      [edit]
      user@srx-01# set chassis high-availability local-id 1
      user@srx-01# set chassis high-availability local-id local-ip 10.22.0.1
      user@srx-01# set chassis high-availability local-id local-forwarding-ip 10.1.100.1
      ```

      Use IP address 10.1.100.1 as local forwarding IP. This IP address is the IP address of loopback interface.

      SRX-02

      ```
      [edit]
      user@srx-02# set chassis high-availability local-id 2
      user@srx-02# set chassis high-availability local-id local-ip 10.22.0.2
      user@srx-02# set chassis high-availability local-id local-forwarding-ip 10.1.200.1
      ```

   b. Configure peer node.

      SRX-01

      ```
      [edit]
      user@srx-01# set chassis high-availability peer-id 2 peer-ip 10.22.0.2
      ```

```
user@srx-01# set chassis high-availability peer-id 2 interface ge-0/0/2.0
user@srx-01# set chassis high-availability peer-id 2 vpn-profile IPSEC_VPN_ICL
user@srx-01# set chassis high-availability peer-id 2 peer-forwarding-ip 10.1.200.1
user@srx-01# set chassis high-availability peer-id 2 peer-forwarding-ip interface lo0.0
user@srx-01# set chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection
minimum-interval 1000
user@srx-01# set chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection
multiplier 5
user@srx-01# set chassis high-availability peer-id 2 liveness-detection minimum-interval
400
user@srx-01# set chassis high-availability peer-id 2 liveness-detection multiplier 5
```

You'll use the ge-0/0/2 interface for communicating with the peer node using the ICL. You are using IP address 10.1.200.1 as peer forwarding IP. This IP address is the IP address of loopback interface on the peer node.

SRX-02

```
[edit]
user@srx-02# set chassis high-availability peer-id 1 peer-ip 10.22.0.1
user@srx-02# set chassis high-availability peer-id 1 interface ge-0/0/2.0
user@srx-02# set chassis high-availability peer-id 1 vpn-profile IPSEC_VPN_ICL
user@srx-02# set chassis high-availability peer-id 1 peer-forwarding-ip 10.1.100.1
user@srx-02# set chassis high-availability peer-id 1 peer-forwarding-ip interface lo0.0
user@srx-02# set chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection
minimum-interval 1000
user@srx-02# set chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection
multiplier 5
user@srx-02# set chassis high-availability peer-id 1 liveness-detection minimum-interval
400
user@srx-02# set chassis high-availability peer-id 1 liveness-detection multiplier 5
```

c. Configure SRG0.

SRX-01

```
[edit]
user@srx-01# set chassis high-availability services-redundancy-group 0 peer-id 2
```

SRX-02

```
[edit]
user@srx-02# set chassis high-availability services-redundancy-group 0 peer-id 1
```

d. Configure SRG 1.

SRX-01

```
[edit]
user@srx-01# set chassis high-availability services-redundancy-group 1 deployment-type
routing
user@srx-01# set chassis high-availability services-redundancy-group 1 peer-id 2
user@srx-01# set chassis high-availability services-redundancy-group 1 activeness-probe
dest-ip 10.111.0.1
user@srx-01# set chassis high-availability services-redundancy-group 1 activeness-probe
dest-ip src-ip 10.11.0.1
user@srx-01# set chassis high-availability services-redundancy-group 1 monitor ip
10.10.10.1
user@srx-01# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.4.0.2 src-ip 10.4.0.1
user@srx-01# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.4.0.2 session-type singlehop
user@srx-01# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.4.0.2 interface ge-0/0/4.0
user@srx-01# set chassis high-availability services-redundancy-group 1 monitor interface
ge-0/0/1
user@srx-01# set chassis high-availability services-redundancy-group 1 active-signal-route
10.39.1.1
user@srx-01# set chassis high-availability services-redundancy-group 1 backup-signal-route
10.39.1.2
user@srx-01# set chassis high-availability services-redundancy-group 1 preemption
user@srx-01# set chassis high-availability services-redundancy-group 1 activeness-priority
200
```

SRX-02

```
[edit]
user@srx-02# set chassis high-availability services-redundancy-group 1 deployment-type
routing
user@srx-02# set chassis high-availability services-redundancy-group 1 peer-id 1
```

```
user@srx-02# set chassis high-availability services-redundancy-group 1 activeness-probe
dest-ip 10.111.0.1
user@srx-02# set chassis high-availability services-redundancy-group 1 activeness-probe
dest-ip src-ip 10.11.0.1
user@srx-02# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.5.0.2 src-ip 10.5.0.1
user@srx-02# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.5.0.2 session-type singlehop
user@srx-02# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.5.0.2 interface ge-0/0/4.0
user@srx-02# set chassis high-availability services-redundancy-group 1 active-signal-route
10.39.1.1
user@srx-02# set chassis high-availability services-redundancy-group 1 backup-signal-route
10.39.1.2
user@srx-02# set chassis high-availability services-redundancy-group 1 activeness-priority
1
```

NOTE: You must specify the active signal route along with the route-exists policy in the policy-options statement. When you configure the `active-signal-route` with `if-route-exists` condition, the HA module adds this route to the routing table.

4. Configure routing-options

   SRX-01

```
[edit]
user@srx-01# set routing-options autonomous-system 65000
user@srx-01# set routing-options static route 10.1.0.0/24 next-hop 10.2.0.1
user@srx-01# set routing-options static route 10.6.0.0/24 next-hop 10.4.0.2
user@srx-01# set routing-options static route 10.111.0.1/32 next-hop 10.2.0.1
user@srx-01# set routing-options static route 10.111.0.2/32 next-hop 10.4.0.2
user@srx-01# set routing-options static route 10.1.200.1/32 next-hop 10.200.200.1
user@srx-02# set routing-options static route 10.1.200.1/32 next-hop 10.100.100.1
```

   SRX-02

```
[edit]
user@srx-02# set routing-options autonomous-system 65000
user@srx-02# set routing-options static route 10.1.0.0/24 next-hop 10.3.0.1
user@srx-02# set routing-options static route 10.6.0.0/24 next-hop 10.5.0.2
user@srx-02# set routing-options static route 10.111.0.1/32 next-hop 10.3.0.1
```

```
user@srx-02# set routing-options static route 10.111.0.2/32 next-hop 10.5.0.2
user@srx-02# set routing-options static route 10.1.100.1/32 next-hop 10.200.200.2
user@srx-02# set routing-options static route 10.1.100.1/32 next-hop 10.100.100.2
```

5. Configure policy options.

   SRX-01

```
[edit]
user@srx-01# set policy-options policy-statement mnha-route-policy term 1 from protocol static
user@srx-01# set policy-options policy-statement mnha-route-policy term 1 from protocol direct
user@srx-01# set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists
user@srx-01# set policy-options policy-statement mnha-route-policy term 1 then metric 10
user@srx-01# set policy-options policy-statement mnha-route-policy term 1 then accept
user@srx-01# set policy-options policy-statement mnha-route-policy term 2 from protocol static
user@srx-01# set policy-options policy-statement mnha-route-policy term 2 from protocol direct
user@srx-01# set policy-options policy-statement mnha-route-policy term 2 from condition
backup_route_exists
user@srx-01# set policy-options policy-statement mnha-route-policy term 2 then metric 20
user@srx-01# set policy-options policy-statement mnha-route-policy term 2 then accept
user@srx-01# set policy-options policy-statement mnha-route-policy term 3 from protocol static
user@srx-01# set policy-options policy-statement mnha-route-policy term 3 from protocol direct
user@srx-01# set policy-options policy-statement mnha-route-policy term 3 then metric 30
user@srx-01# set policy-options policy-statement mnha-route-policy term 3 then accept
user@srx-01# set policy-options policy-statement mnha-route-policy term default then reject
user@srx-01# set policy-options condition active_route_exists if-route-exists address-family
inet 10.39.1.1/32
user@srx-01# set policy-options condition active_route_exists if-route-exists address-family
inet table inet.0
user@srx-01# set policy-options condition backup_route_exists if-route-exists address-family
inet 10.39.1.2/32
user@srx-01# set policy-options condition backup_route_exists if-route-exists address-family
inet table inet.0
```

   SRX-02

```
[edit]
user@srx-02# set policy-options route-filter-list ipsec 10.6.0.0/16 orlonger
user@srx-02# set policy-options route-filter-list loopback 10.11.0.0/24 orlonger
user@srx-02# set policy-options policy-statement mnha-route-policy term 1 from protocol static
user@srx-02# set policy-options policy-statement mnha-route-policy term 1 from protocol direct
```

```
user@srx-02# set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists
user@srx-02# set policy-options policy-statement mnha-route-policy term 1 then metric 10
user@srx-02# set policy-options policy-statement mnha-route-policy term 1 then accept
user@srx-02# set policy-options policy-statement mnha-route-policy term 2 from protocol static
user@srx-02# set policy-options policy-statement mnha-route-policy term 2 from protocol direct
user@srx-02# set policy-options policy-statement mnha-route-policy term 2 from condition
backup_route_exists
user@srx-02# set policy-options policy-statement mnha-route-policy term 2 then metric 20
user@srx-02# set policy-options policy-statement mnha-route-policy term 2 then accept
user@srx-02# set policy-options policy-statement mnha-route-policy term 3 from protocol static
user@srx-02# set policy-options policy-statement mnha-route-policy term 3 from protocol direct
user@srx-02# set policy-options policy-statement mnha-route-policy term 3 then metric 35
user@srx-02# set policy-options policy-statement mnha-route-policy term 3 then accept
user@srx-02# set policy-options policy-statement mnha-route-policy term default then reject
user@srx-02# set policy-options condition active_route_exists if-route-exists address-family
inet 10.39.1.1/32
user@srx-02# set policy-options condition active_route_exists if-route-exists address-family
inet table inet.0
user@srx-02# set policy-options condition backup_route_exists if-route-exists address-family
inet 10.39.1.2/32
user@srx-02# set policy-options condition backup_route_exists if-route-exists address-family
inet table inet.0
```

6. Specify options to secure high availability traffic flow between the nodes through ICL.

   SRX-01 and SRX-02

   a. Configure PKI options.

```
[edit]
user@srx-01# set security pki ca-profile Root-CA ca-identity Root-CA
user@srx-01# set security pki ca-profile Root-CA enrollment url http://10.157.69.204/
certsrv/mscep/mscep.dll
user@srx-01# set security pki ca-profile Root-CA revocation-check disable
```

   b. Define Internet Key Exchange (IKE) configuration. An IKE configuration defines the algorithms and keys used to establish a secure connection.

```
[edit]
user@srx-01# set security ike proposal MNHA_IKE_PROP description mnha_link_encr_tunnel
user@srx-01# set security ike proposal MNHA_IKE_PROP authentication-method pre-shared-keys
```

```
user@srx-01# set security ike proposal MNHA_IKE_PROP dh-group group14
user@srx-01# set security ike proposal MNHA_IKE_PROP authentication-algorithm sha-256
user@srx-01# set security ike proposal MNHA_IKE_PROP encryption-algorithm aes-256-cbc
user@srx-01# set security ike proposal MNHA_IKE_PROP lifetime-seconds 3600
user@srx-01# set security ike policy MNHA_IKE_POL description mnha_link_encr_tunnel
user@srx-01# set security ike policy MNHA_IKE_POL proposals MNHA_IKE_PROP
user@srx-01# set security ike policy MNHA_IKE_POL pre-shared-key ascii-text "$ABC123"
user@srx-01# set security ike gateway MNHA_IKE_GW ike-policy MNHA_IKE_POL
user@srx-01# set security ike gateway MNHA_IKE_GW version v2-only
```

c. IPsec proposal protocol and encryption algorithm

```
[edit]
user@srx-01# set security ipsec proposal MNHA_IPSEC_PROP description mnha_link_encr_tunnel
user@srx-01# set security ipsec proposal MNHA_IPSEC_PROP protocol esp
user@srx-01# set security ipsec proposal MNHA_IPSEC_PROP encryption-algorithm aes-256-gcm
user@srx-01# set security ipsec proposal MNHA_IPSEC_PROP lifetime-seconds 3600
user@srx-01# set security ipsec policy MNHA_IPSEC_POL description mnha_link_encr_tunnel
user@srx-01# set security ipsec policy MNHA_IPSEC_POL proposals MNHA_IPSEC_PROP
user@srx-01# set security ipsec vpn IPSEC_VPN_ICL ha-link-encryption
user@srx-01# set security ipsec vpn IPSEC_VPN_ICL ike gateway MNHA_IKE_GW
user@srx-01# set security ipsec vpn IPSEC_VPN_ICL ike ipsec-policy MNHA_IPSEC_POL
```

7. Configure the security policy .

   SRX-01 and SRX-02

```
[edit]
user@srx-01# set security policies default-policy permit-all
```

TIP: The security policy shown in this example is only for demonstration. You should configure security policies as per your network needs. Ensure that your security policies allow only the applications, users, and devices that you trust.

8. Configure BFD peering sessions options and specify liveness detection timers.

   (SRX-01)

```
[edit]
user@srx-01# set protocols bgp group trust type internal
user@srx-01# set protocols bgp group trust local-address 10.2.0.2
```

```
user@srx-01# set protocols bgp group trust export mnha-route-policy
user@srx-01# set protocols bgp group trust local-as 65000
user@srx-01# set protocols bgp group trust bfd-liveness-detection minimum-interval 500
user@srx-01# set protocols bgp group trust bfd-liveness-detection minimum-receive-interval 500
user@srx-01# set protocols bgp group trust bfd-liveness-detection multiplier 3
user@srx-01# set protocols bgp group trust neighbor 10.2.0.1
user@srx-01# set protocols bgp group untrust type internal
user@srx-01# set protocols bgp group untrust local-address 10.4.0.1
user@srx-01# set protocols bgp group untrust export mnha-route-policy
user@srx-01# set protocols bgp group untrust local-as 65000
user@srx-01# set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
user@srx-01# set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval
500
user@srx-01# set protocols bgp group untrust bfd-liveness-detection multiplier 3
user@srx-01# set protocols bgp group untrust neighbor 10.4.0.2
```

SRX-02

```
[edit]
user@srx-02# set protocols bgp group trust type internal
user@srx-02# set protocols bgp group trust local-address 10.3.0.2
user@srx-02# set protocols bgp group trust export mnha-route-policy
user@srx-02# set protocols bgp group trust local-as 65000
user@srx-02# set protocols bgp group trust bfd-liveness-detection minimum-interval 500
user@srx-02# set protocols bgp group trust bfd-liveness-detection minimum-receive-interval 500
user@srx-02# set protocols bgp group trust bfd-liveness-detection multiplier 3
user@srx-02# set protocols bgp group trust neighbor 10.3.0.1
user@srx-02# set protocols bgp group untrust type internal
user@srx-02# set protocols bgp group untrust local-address 10.5.0.1
user@srx-02# set protocols bgp group untrust export mnha-route-policy
user@srx-02# set protocols bgp group untrust local-as 65000
user@srx-02# set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
user@srx-02# set protocols bgp group untrust bfd-liveness-detection minimum-receive-interval
500
user@srx-02# set protocols bgp group untrust bfd-liveness-detection multiplier 3
user@srx-02# set protocols bgp group untrust neighbor 10.5.0.2
```

Verification

IN THIS SECTION

Use the following show commands to verify the feature in this example.

Table 48: Verification Tasks

Commands	Verification Task
show chassis high availability information	Displays Multinode High Availability details including status.
show chassis high-availability data-plane statistics	Displays ICD data packets statistics.

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:

SRX-01

```
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
Local Forwarding IP: 10.1.100.1
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
    Configured BFD Detection Time: 5 * 400ms
    Cold Sync Status: COMPLETE
    Peer Forwarding IP: 10.1.200.1               Interface: lo0.0
    Peer ICD Conn State: UP

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: [ IP ]
        Peer Information:
          Peer Id: 2
          Status : ACTIVE
          Health Status: HEALTHY
          Failover Readiness: N/A
```

SRX-02

```
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
Local Forwarding IP: 10.1.200.1
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
    Configured BFD Detection Time: 5 * 400ms
    Cold Sync Status: COMPLETE
    Peer Forwarding IP: 10.1.100.1               Interface: lo0.0
    Peer ICD Conn State: UP

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
```

Meaning

Verify these details from the command output:

- Local node and peer node details such as IP address and ID.
- The field `Peer ICD Conn State: UP` indicates that the ICD link is established and operational.

Verify ICD Data Packets Statistics

Purpose

Check if the ICD is operational and facilitating the transfer of data packets between the nodes.

Action

From operational mode, run the following command:

```
user@srx-01> show chassis high-availability data-plane statistics
Services Synchronized:
    Service name                              RTOs sent    RTOs 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 RTO 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

Packet stats                                Pkts sent    Pkts received
    ICD Data                                   1035          1286
```

Meaning

The field `ICD Data` indicates ICD is routing asymmetric traffic flow in Multinode High Availability setup.

Set Commands on All Devices

Set command output on all devices.

SRX-01 (Node 1)

```
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 10.22.0.1
set chassis high-availability local-id local-forwarding-ip 10.1.100.1
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 peer-forwarding-ip 10.1.200.1
set chassis high-availability peer-id 2 peer-forwarding-ip interface lo0.0
set chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection minimum-interval
1000
set chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection multiplier 5
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 ip 10.10.10.1
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 monitor interface ge-0/0/1
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 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 trust interfaces ge-0/0/1.0
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust interfaces lo0.0
set security zones security-zone icl-zone host-inbound-traffic system-services ike
set security zones security-zone icl-zone host-inbound-traffic system-services ping
set security zones security-zone icl-zone host-inbound-traffic system-services high-availability
set security zones security-zone icl-zone host-inbound-traffic system-services ssh
set security zones security-zone icl-zone host-inbound-traffic protocols bfd
set security zones security-zone icl-zone host-inbound-traffic protocols bgp
set security zones security-zone icl-zone interfaces ge-0/0/2.0
set interfaces ge-0/0/0 description icd-1
set interfaces ge-0/0/0 unit 0 family inet address 10.100.100.2/24
```

```
set interfaces ge-0/0/1 description icd-2
set interfaces ge-0/0/1 unit 0 family inet address 10.200.200.2/24
set interfaces ge-0/0/2 description interchassis_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 10.2.0.2/24
set interfaces ge-0/0/4 description untrust
set interfaces ge-0/0/4 unit 0 family inet address 10.4.0.1/24
set interfaces lo0 description trust
set interfaces lo0 unit 0 family inet address 10.1.100.1/32
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 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 local-as 65000
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 neighbor 10.2.0.1
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 local-as 65000
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 10.4.0.2
set routing-options autonomous-system 65000
set routing-options static route 10.1.0.0/24 next-hop 10.2.0.1
set routing-options static route 10.6.0.0/24 next-hop 10.4.0.2
set routing-options static route 10.111.0.1/32 next-hop 10.2.0.1
set routing-options static route 10.111.0.2/32 next-hop 10.4.0.2
set routing-options static route 10.1.200.1/32 next-hop 10.200.200.1
set routing-options static route 10.1.200.1/32 next-hop 10.100.100.1
```

SRX-02 (Node 2)

```
set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 10.22.0.2
set chassis high-availability local-id local-forwarding-ip 200.1.1.1
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 peer-forwarding-ip 100.1.1.1
set chassis high-availability peer-id 1 peer-forwarding-ip interface lo0.0
set chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection minimum-interval
1000
set chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection multiplier 5
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 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 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 trust interfaces lo0.0
set security zones security-zone trust interfaces ge-0/0/1.0
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone icl-zone host-inbound-traffic system-services ike
set security zones security-zone icl-zone host-inbound-traffic system-services ping
set security zones security-zone icl-zone host-inbound-traffic system-services high-availability
set security zones security-zone icl-zone host-inbound-traffic system-services ssh
set security zones security-zone icl-zone host-inbound-traffic protocols bfd
set security zones security-zone icl-zone host-inbound-traffic protocols bgp
set security zones security-zone icl-zone interfaces ge-0/0/2.0
set interfaces ge-0/0/0 description icd-1
set interfaces ge-0/0/0 unit 0 family inet address 10.100.100.1/24
```

```
set interfaces ge-0/0/1 description icd-2
set interfaces ge-0/0/1 unit 0 family inet address 10.200.200.1/24
set interfaces ge-0/0/2 description interchassis_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 10.3.0.2/24
set interfaces ge-0/0/4 description untrust
set interfaces ge-0/0/4 unit 0 family inet address 10.5.0.1/24
set interfaces lo0 description trust
set interfaces lo0 unit 0 family inet address 10.1.200.1/32
set policy-options route-filter-list ipsec 10.6.0.0/16 orlonger
set policy-options route-filter-list loopback 10.11.0.0/24 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 35
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 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 local-as 65000
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 neighbor 10.3.0.1
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 local-as 65000
```

```
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 10.5.0.2
set routing-options autonomous-system 65000
set routing-options static route 10.1.0.0/24 next-hop 10.3.0.1
set routing-options static route 10.6.0.0/24 next-hop 10.5.0.2
set routing-options static route 10.111.0.1/32 next-hop 10.3.0.1
set routing-options static route 10.111.0.2/32 next-hop 10.5.0.2
set routing-options static route 10.1.100.1/32 next-hop 10.200.200.2
set routing-options static route 10.1.100.1/32 next-hop 10.100.100.2
```

Router -1

```
set interfaces ge-0/0/0 description ha
set interfaces ge-0/0/0 unit 0 family inet address 10.2.0.1/24
set interfaces ge-0/0/1 description ha
set interfaces ge-0/0/1 unit 0 family inet address 10.3.0.1/24
set interfaces ge-0/0/2 description lan
set interfaces ge-0/0/2 unit 0 family inet address 10.1.0.1/24
set interfaces lo0 description loopback
set interfaces lo0 unit 0 family inet address 10.111.0.1/32 primary
set interfaces lo0 unit 0 family inet address 10.111.0.1/32 preferred
set routing-options autonomous-system 65000
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 local-as 65000
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 10.2.0.2
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 local-as 65000
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 10.3.0.2
```

Router-2

```
set interfaces ge-0/0/0 description HA
set interfaces ge-0/0/0 unit 0 family inet address 10.4.0.2/24
set interfaces ge-0/0/1 description HA
set interfaces ge-0/0/1 unit 0 family inet address 10.5.0.2/24
set interfaces ge-0/0/2 description trust
set interfaces ge-0/0/2 unit 0 family inet address 10.6.0.1/24
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 65000
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 local-as 65000
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 10.4.0.1
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 local-as 65000
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 10.5.0.1
```

Show Configuration Output

IN THIS SECTION

From configuration mode, confirm your configuration by entering the `show high availability`, , `show security zones`, and `show interfaces`. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

SRX-01 (Node 1)

```
user@srx-01# show chassis high-availability
local-id {
    1;
    local-ip 10.22.0.1;
    local-forwarding-ip 100.1.1.1;
}
peer-id 2 {
    peer-ip 10.22.0.2;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    peer-forwarding-ip {
        200.1.1.1;
        interface lo0.0;
        liveness-detection {
            minimum-interval 1000;
            multiplier 5;
        }
    }
    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 {
        ip 10.10.10.1;
        bfd-liveliness 10.4.0.2 {
            src-ip 10.4.0.1;
            session-type singlehop;
            interface ge-0/0/4.0;
        }
        interface {
            ge-0/0/1;
        }
    }
    active-signal-route {
        10.39.1.1;
    }
    backup-signal-route {
        10.39.1.2;
    }
    preemption;
    activeness-priority 200;
}
```

```
user@srx-01# show interfaces
ge-0/0/0 {
    description icd-1;
    unit 0 {
        family inet {
            address 10.100.100.2/24;
        }
    }
}
ge-0/0/1 {
    description icd-2;
    unit 0 {
        family inet {
            address 10.200.200.2/24;
        }
    }
}
ge-0/0/2 {
    description interchassis_link;
    unit 0 {
```

```
        family inet {
            address 10.22.0.1/24;
        }
    }
}
ge-0/0/3 {
    description trust;
    unit 0 {
        family inet {
            address 10.2.0.2/24;
        }
    }
}
ge-0/0/4 {
    description untrust;
    unit 0 {
        family inet {
            address 10.4.0.1/24;
        }
    }
}
lo0 {
    description trust;
    unit 0 {
        family inet {
            address 10.1.100.1/32;
        }
    }
}
```

```
user@srx-01# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
```

```
    }
    interfaces {
        ge-0/0/4.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
        ge-0/0/1.0;
        ge-0/0/0.0;
        lo0.0;
    }
}
security-zone icl-zone {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
            high-availability;
            ssh;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
```

```
user@srx-01# 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@srx-01# show routing-options
autonomous-system 65000;
static {
    route 10.1.0.0/24 next-hop 10.2.0.1;
    route 10.6.0.0/24 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;
    route 10.1.200.1/32 next-hop [ 10.200.200.1 10.100.100.1 ];
}
```

SRX-02 (Node 2)

```
user@srx-02# show chassis high-availability
local-id {
    2;
    local-ip 10.22.0.2;
    local-forwarding-ip 200.1.1.1;
}
peer-id 1 {
    peer-ip 10.22.0.1;
    interface ge-0/0/2.0;
    vpn-profile IPSEC_VPN_ICL;
    peer-forwarding-ip {
        100.1.1.1;
        interface lo0.0;
        liveness-detection {
            minimum-interval 1000;
            multiplier 5;
        }
    }
```

```
    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;
}
```

```
user@srx-02# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
```

```
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/4.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/3.0;
        lo0.0;
        ge-0/0/1.0;
        ge-0/0/0.0;
    }
}
security-zone icl-zone {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
            high-availability;
            ssh;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/2.0;
```

```
    }
}
```

```
user@srx-02# show interfaces
ge-0/0/0 {
    description icd-1;
    unit 0 {
        family inet {
            address 10.100.100.1/24;
        }
    }
}
ge-0/0/1 {
    description icd-2;
    unit 0 {
        family inet {
            address 10.200.200.1/24;
        }
    }
}
ge-0/0/2 {
    description interchassis_link;
    unit 0 {
        family inet {
            address 10.22.0.2/24;
        }
    }
}
ge-0/0/3 {
    description trust;
    unit 0 {
        family inet {
            address 10.3.0.2/24;
        }
    }
}
ge-0/0/4 {
    description untrust;
    unit 0 {
        family inet {
```

```
            address 10.5.0.1/24;
        }
    }
}
lo0 {
    description trust;
    unit 0 {
        family inet {
            address 10.1.200.1/32;
        }
    }
}
```

```
user@srx-02# show policy-options

route-filter-list ipsec {
    10.6.0.0/16 orlonger;
}
route-filter-list loopback {
    10.11.0.0/24 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 35;
            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@srx-02# show routing-options
autonomous-system 65000;
static {
    route 10.1.0.0/24 next-hop 10.3.0.1;
    route 10.6.0.0/24 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;
```

```
    route 10.1.100.1/32 next-hop [ 10.200.200.2 10.100.100.2 ];
}
```

Change History Table

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

Release	Description
23.4	Multinode High Availability support for asymmetric traffic flows.

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 Firewalls. The example covers configuration in active/backup mode when SRX Series Firewalls are connected to routers on both sides.

IN THIS SECTION

Overview

In Multi-Node High Availability, participating SRX Series Firewalls 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 Firewall, 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: We support a two-node configuration in the Multinode High Availability solution.

Requirements

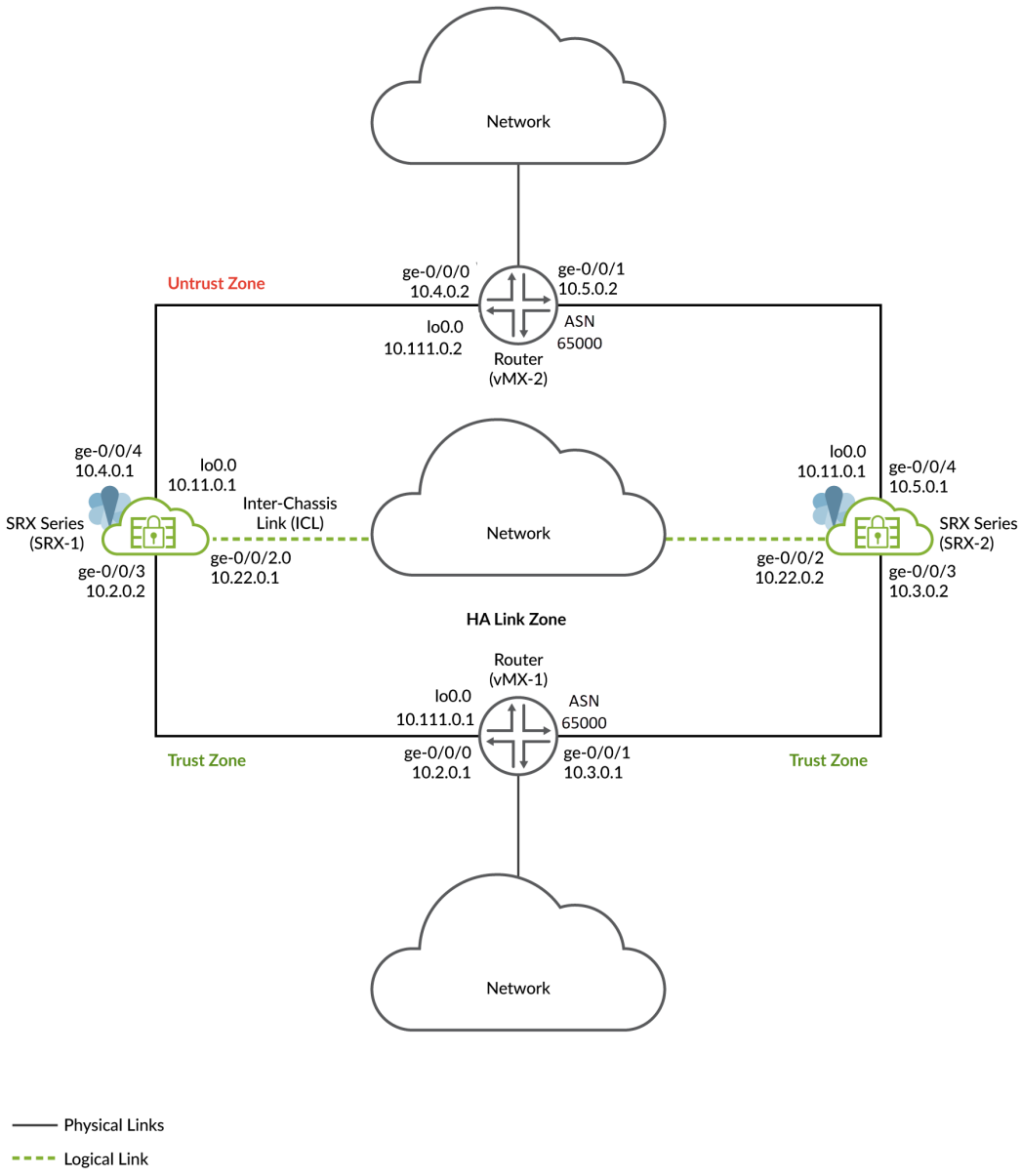
This example uses the following hardware and software components:

- Two SRX Series Firewalls or vSRX Virtual Firewall instances
- Two Juniper Networks(R) MX960 Universal Routing Platform
- Junos OS Release 22.3R1

Topology

Figure 65 on page 700 shows the topology used in this example.

Figure 65: Multinode High Availability in Layer 3 Network



As shown in the topology, two SRX Series Firewalls are connected to adjacent routers on trust and untrust side forming a BGP neighborship. 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 Firewalls 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 Firewalls.

In this example, you'll establish high availability between the SRX Series Firewalls 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 Firewalls 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.

In this example, you use static routes on SRX-1 and SRX-2 and advertise these routes into BGP to add the metric to determine which SRX Series Firewall is in the preferred path. Alternatively you can use route reflectors on the SRX Series Firewalls to advertise the routes learned via BGP and accordingly configure the routing policy to match on BGP.

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.

  When you configure monitoring (BFD or IP or Interface) options under SRG1, we recommend not to configure the shutdown-on-failure option under SRG0.

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. Do not to use the dedicated HA ports (control and fabric ports) if available on your SRX Series Firewall).
- 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.

A secure tunnel interface (st0) from st0.16000 to st0.16385 is reserved for Multinode High Availability. These interfaces are not user configurable interfaces. You can only use interfaces from st0.0 to st0.15999.

Configuration

IN THIS SECTION

Before You Begin

Junos IKE package is required on your SRX Series Firewalls for Multinode High Availability configuration. This package is available as a default package or as an optional package on SRX Series Firewalls. See Support for Junos IKE Package for details.

If the package is not installed by default on your SRX Series firewall, use the following command to install it. 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 routing-options autonomous-system 65000
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 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 65000
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 65000
```

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 routing-options autonomous-system 65000
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 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 35
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 65000
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 65000
```

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 65000
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 65000
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 65000
```

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 65000
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 65000
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 65000
```

Configuration

Step-by-Step Procedure

We're showing the configuration of SRX-1 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
```

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 65000
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
```

In this step, the active SRX Series Firewall creates the route with IP address 10.39.1.1 and the backup SRX Series Firewall creates the route with IP address 10.39.1.2 depending on the configuration. In this example, the policy on the SRX-1 matches on 10.39.1.1 (since its active) and advertises static/direct routes with a metric 10 making it preferred. The policy on SRX-2 matches on 10.39.1.2 (since its backup) and advertises static/direct routes with a metric 20 making it less preferred.

The active signal route IP address you assign is used for route preference advertisement.

NOTE: You must specify the active signal route along with the route-exists policy in the policy-options statement. When you configure the `active-signal-route` with `if-route-exists` condition, the HA module adds this route to the routing table.

13. 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`).

14. 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.

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 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 65000
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 1051 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 SRG0.

```
user@host# set chassis high-availability services-redundancy-group 0 install-on-failure-route
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 SRG0 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 65000;
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;
        }
    }
}
```

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 35;
            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 65000;
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;
        }
    }
}
```

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

- Verify Interchassis Link Active Peers | 742

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

CAUTION: The IP range (180.100.1.x) shown in the command output serves as the ICL IPsec traffic selector. The system dynamically assigns this IP range, and it is essential not to alter or modify it. Additionally, BFD (Bidirectional Forwarding Detection) will be automatically enabled for the broader 180.x.x.x IP range.

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

Two-Node Multinode High Availability | 573

Multinode High Availability Services | 624

Prepare Your Environment for Multinode High Availability Deployment | 620

Example: Configure Multinode High Availability in a Default Gateway Deployment | 743

Example: Configure Multinode High Availability in a Hybrid Deployment | 778

Example: Configure Multinode High Availability in a Default Gateway Deployment

SUMMARY

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

IN THIS SECTION

Overview

In Multi-Node High Availability, participating SRX Series Firewalls 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 Firewall, 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:

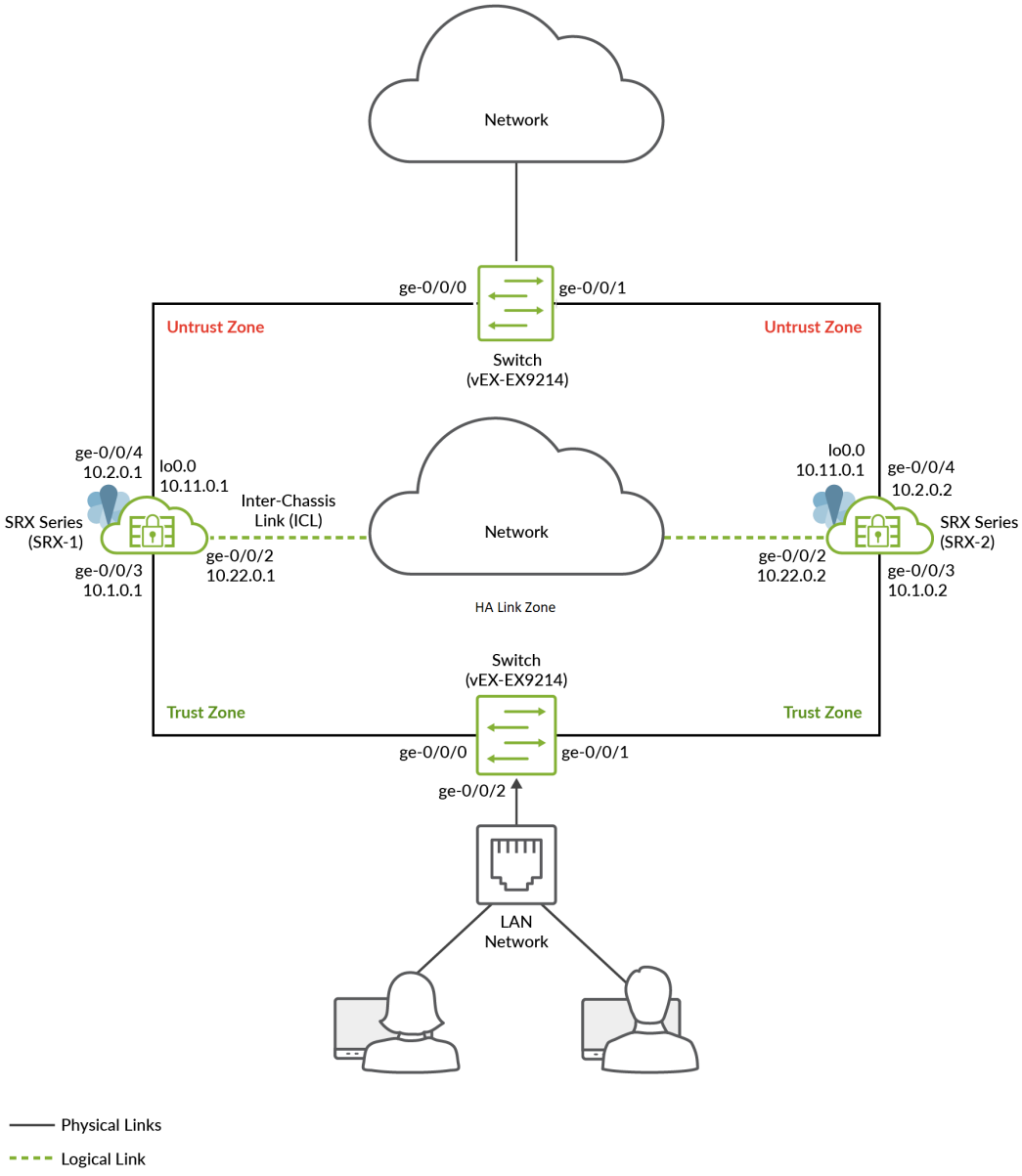
- Two SRX Series Firewalls or vSRX Virtual Firewall instances

- Two Juniper Networks EX9214 Ethernet Switches
- Junos OS Release 22.3R1

Topology

Figure 66 on page 745 shows the topology used in this example.

Figure 66: Multinode High Availability in Default Gateway Deployment



As shown in the topology, two SRX Series Firewalls 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 Firewalls.

In general, you can use Aggregated Ethernet (AE) or a revenue Ethernet port on the SRX Series Firewalls 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 Firewalls.

In this example, you use static routes on SRX-1 and SRX-2 and advertise these routes into BGP to add the metric to determine which SRX Series Firewall is in the preferred path. Alternatively you can use route reflectors on the SRX Series Firewall to advertise the routes learned via BGP and accordingly configure the routing policy to match on BGP.

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

- Configure a pair of SRX Series Firewalls as local and peer nodes by assigning IDs.
- Configure services redundancy groups (SRGs).
- Configure virtual IP addresses for activeness determination and enforcement.
- Configure a VPN profile for the high availability (ICL) traffic using IKEv2.
- 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.

  When you configure monitoring (BFD or IP or Interface) options under SRG1, we recommend not to configure the shutdown-on-failure option under SRG0.

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. Do not to use the dedicated HA ports (control and fabric ports) if available on your SRX Series Firewall).
- 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

Junos IKE package is required on your SRX Series Firewalls for Multinode High Availability configuration. This package is available as a default package or as an optional package on SRX Series Firewalls. See Support for Junos IKE Package for details.

If the package is not installed by default on your SRX Series firewall, use the following command to install it. 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 65000
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 system services netconf ssh
```

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 routing-options autonomous-system 65000
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 system services netconf ssh
```

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 65000
```

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.

> **NOTE**: Configuring the `use-virtual-mac` option is the recommended option in most cases, except where the surrounding infrastructure would not support a moving virtual MAC address active on a port in addition to the local MAC address.

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. 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`

14. 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.

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 1051 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 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 routing-options
autonomous-system 65000;
```

```
[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 routing-options
autonomous-system 65000;
```

```
[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;
        }
    }
}
```

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

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

CAUTION: The IP range (180.100.1.x) shown in the command output serves as the ICL IPsec traffic selector. The system dynamically assigns this IP range, and it is essential not to alter or modify it. Additionally, BFD (Bidirectional Forwarding Detection) will be automatically enabled for the broader 180.x.x.x IP range.

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

Two-Node Multinode High Availability | 573

Prepare Your Environment for Multinode High Availability Deployment | 620

Multinode High Availability Services | 624

Example: Configure Multinode High Availability in a Layer 3 Network | 698

Example: Configure Multinode High Availability in a Hybrid Deployment | 778

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 Firewalls. The example covers configuration in active/backup mode when SRX Series Firewalls are connected to a router on one side and switch on the other side.

IN THIS SECTION

Overview

In a hybrid deployments, participating SRX Series Firewalls 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 Firewall, 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 Firewalls and secure the tunnel traffic by enabling HA link encryption.

Requirements

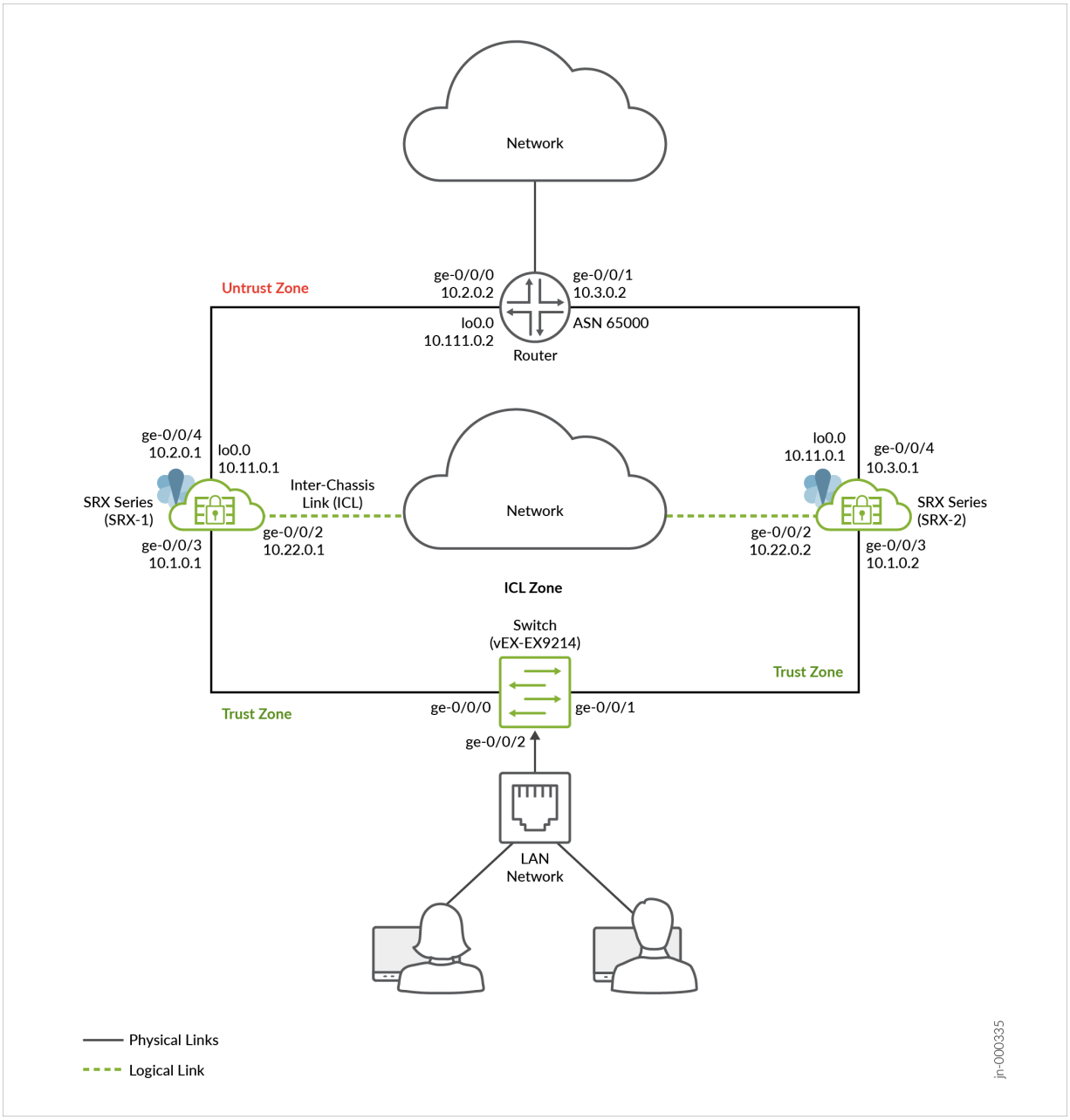
This example uses the following hardware and software components:

- Two SRX Series Firewalls or vSRX Virtual Firewall 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 67: Multinode High Availability In Hybrid Network



As shown in the topology, two SRX Series Firewalls 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 Firewalls 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 router and one switch.

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

- Configure a pair of SRX Series Firewalls 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.

In this example, you use static routes on SRX-1 and SRX-2 and advertise these routes into BGP to add the metric to determine which SRX Series Firewall is in the preferred path. Alternatively you can use route reflectors on the SRX Series Firewalls to advertise the routes learned via BGP and accordingly configure the routing policy to match on BGP.

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.

  When you configure monitoring (BFD or IP or Interface) options under SRG1, we recommend not to configure the shutdown-on-failure option under SRG0.

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. Do not to use the dedicated HA ports (control and fabric ports) if available on your SRX Series Firewall).
- 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

Junos IKE package is required on your SRX Series Firewalls for Multinode High Availability configuration. This package is available as a default package or as an optional package on SRX Series Firewalls. See Support for Junos IKE Package for details.

If the package is not installed by default on your SRX Series firewall, use the following command to install it. 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 10.2.0.2 src-ip
10.2.0.1
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.2.0.2
session-type singlehop
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.2.0.2
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 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 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 65000
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 65000
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 10.3.0.2 src-ip
10.3.0.1
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.3.0.2
session-type singlehop
set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness 10.3.0.2
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 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 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 35
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 65000
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 65000
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 65000
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 65000
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 65000
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 65000
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.

NOTE: Configuring the `use-virtual-mac` option is the recommended option in most cases, except where the surrounding infrastructure would not support a moving virtual MAC address active on a port in addition to the local MAC address.

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
10.2.0.2 src-ip 10.2.0.1
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.2.0.2 session-type singlehop
```

```
user@host# set chassis high-availability services-redundancy-group 1 monitor bfd-liveliness
10.2.0.2 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
```

In this step, the active SRX Series Firewall creates the route with IP address 10.39.1.1 and the backup SRX Series Firewall creates the route with IP address 10.39.1.2 depending on the configuration. In this example, the policy on the SRX-1 matches on 10.39.1.1 (since its active) and advertises static/direct routes with a metric 10 making it preferred. The policy on SRX-2 matches on 10.39.1.2 (since its backup) and advertises static/direct routes with a metric 20 making it less preferred.

The active signal route IP address you assign is used for route preference advertisement.

NOTE: You must specify the active signal route along with the route-exists policy in the policy-options statement. When you configure the `active-signal-route` with `if-route-exists` condition, the HA module adds this route to the routing table.

13. 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
```

14. 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 65000
```

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
```

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.

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 1051 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 10.2.0.2 {
            src-ip 10.2.0.1;
```

```
            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 65000;
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 10.3.0.2 {
            src-ip 10.3.0.1;
            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 35;
            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 65000;
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

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

CAUTION: The IP range (180.100.1.x) shown in the command output serves as the ICL IPsec traffic selector. The system dynamically assigns this IP range, and it is essential not to alter or modify it. Additionally, BFD (Bidirectional Forwarding Detection) will be automatically enabled for the broader 180.x.x.x IP range.

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

- Two-Node Multinode High Availability | 573
- Multinode High Availability Services | 624
- Prepare Your Environment for Multinode High Availability Deployment | 620
- Software Upgrade in Multinode High Availability | 1051
- Example: Configure Multinode High Availability in a Default Gateway Deployment | 743
- Example: Configure Multinode High Availability in a Layer 3 Network | 698

Example: Configure IPSec VPN in Active-Active Multinode High Availability in a Layer 3 Network

SUMMARY

This example shows how to configure and verify IPsec VPN for active-active Multinode High Availability setup.

IN THIS SECTION

Overview

In Multi-Node High Availability, participating SRX Series Firewalls 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.

You can operate Multinode High Availability in active-active mode with support of multiple services redundancy groups (SRGs). In this mode, some SRGs remain active on one node and some SRGs remain active on another node.

Multinode High Availability supports IPsec VPN in active-active mode with multiple SRGs (SRG1+). In this mode, you can establish multiple active tunnels from both the nodes, based on SRG activeness. Multinode High Availability establishes IPsec tunnel and performs key exchanges by associating termination IP address (which also identifies the tunnels ending on it) to the SRG. Since different SRG1+ can be in active state or in backup state on each of the devices, Multinode High Availability steers the matching traffic effectively to the corresponding active SRG1. Since different SRGs can be active on different nodes, tunnels belonging to these SRGs come up on both nodes independently.

NOTE: We support a two-node configuration in the Multinode High Availability solution.

Requirements

IN THIS SECTION

This example uses the following hardware and software components:

- Two SRX Series Firewalls (Supported devices are SRX5400, SRX5600, and SRX5800 with SPC3, IOC3, SCB3, SCB4, and RE3)
- Junos OS Release 22.4R1

We've used two Juniper Networks MX Series Routing Platform as upstream/downstream routers in this example.

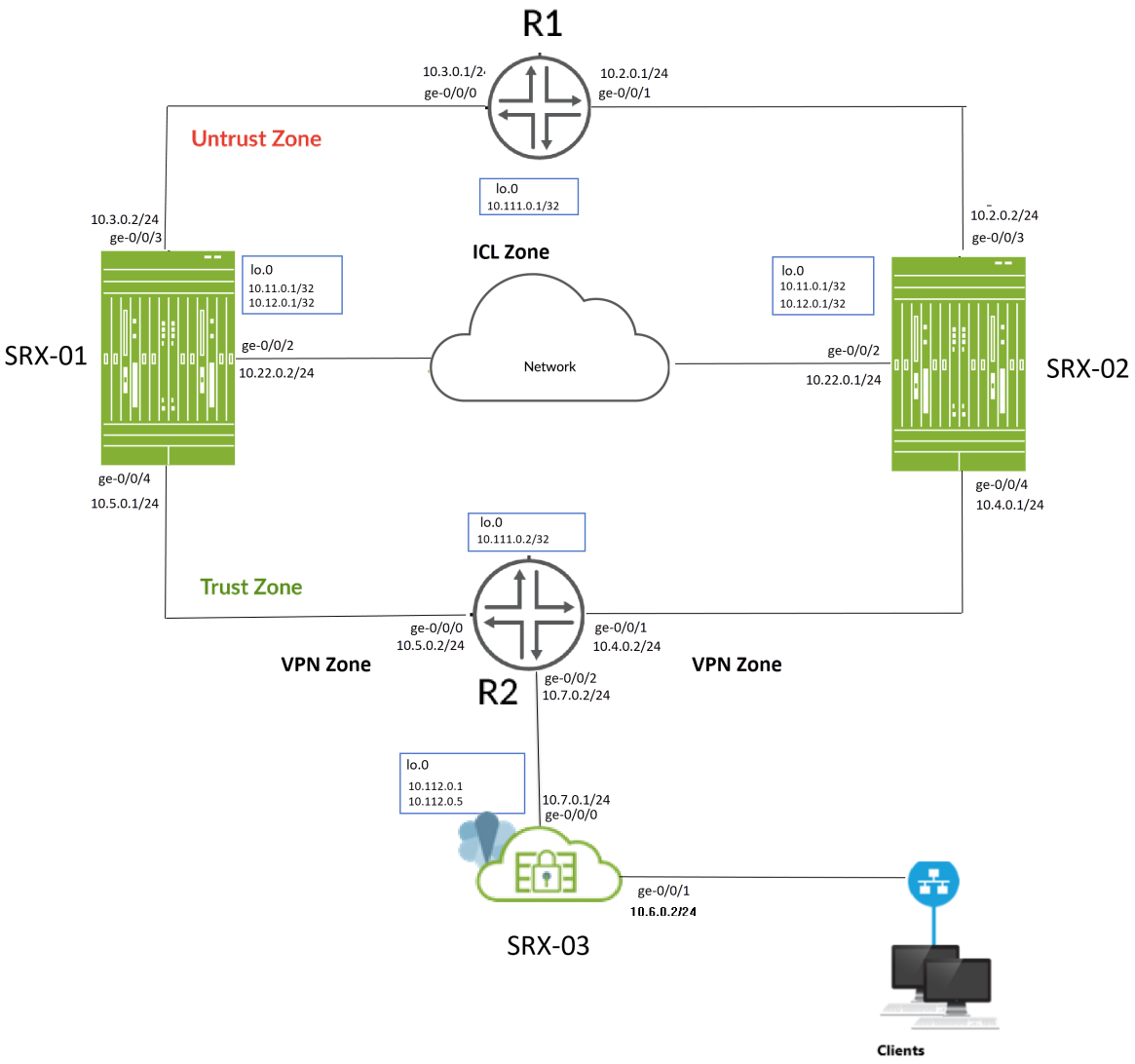
Before You Begin

- Configure stateless firewall filtering and quality of service (QoS) as per your network requirements and have appropriate security policies to manage traffic in your network.
- 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 Firewalls. Ensure that you've configured upstream and downstream routers as per your network requirements.
- Install the Junos IKE package on your SRX Series Firewalls using the `request system software add optional://junos-ike.tgz` command. The `junos-ike` package is included in your Junos software packages (Junos OS Release 20.4R1 onwards).

Topology

Figure 68 on page 824 shows the topology used in this example.

Figure 68: Multinode High Availability in Layer 3 Network



As shown in the topology, two SRX Series Firewalls (SRX-1 and SRX-2) are connected to adjacent routers on trust and untrust side forming a BGP neighborship. 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.

The SRX-03 device acts as a peer device to the Multinode High Availability setup and it establishes IPsec VPN tunnels with SRX-01 and SRX-02 devices.

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

- Configure a pair of SRX Series Firewalls as local and peer nodes by assigning IDs.
- Configure services redundancy groups (SRG1 and SRG2).

- Configure a loopback interface (lo0.0) to host the floating IP address and to reach the peer gateway. 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).
- 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 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.
- Create a group configuration for IPsec VPN on SRX-01 and SRX-02 devices to set up a tunnel with VPN peer device (SRX-03). Configuration groups enable you to apply common elements that are reused within the same configuration.
- Configure IPsec VPN options to establish tunnels with SRX-03 device and enable IPsec VPN configuration synchronization on both the devices (SRX-01 and SRX-02) by using [groups] option.
- Configure VPN peer device with IPsec VPN options.

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

- In general, you can use Aggregated Ethernet (AE) or a revenue Ethernet port on the SRX Series Firewalls 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.
- Do not to use the dedicated HA ports (control and fabric ports) if available on your SRX Series Firewall).
- 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.

You can configure the following options on SRG0 and SRG1+:

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.

  When you configure monitoring (BFD or IP or Interface) options under SRG1, we recommend not to configure the shutdown-on-failure option under SRG0.

- 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.

  When you configure monitoring (BFD or IP or Interface) options under SRG1, we recommend not to configure the shutdown-on-failure option under SRG0.

Table 49 on page 826 shows the details on interfaces configuration used in this example.

Table 49: Interfaces and IP Address Configuration on Security Devices

Device	Interface	Zone	IP Address	Configured For
SRX-01	lo0	Untrust	10.11.0.1/32	Floating IP address IKE Gateway address
			10.12.0.1/32	IKE Gateway address
	ge-0/0/2	ICL	10.22.0.2/24	Connecting ICL
	ge-0/0/4	Trust	10.5.0.1/24	Connects to R2 router
	ge-0/0/3	Untrust	10.3.0.2/24	Connects to R1 router

Table 49: Interfaces and IP Address Configuration on Security Devices *(Continued)*

Device	Interface	Zone	IP Address	Configured For
SRX-02	lo0	Untrust	10.12.0.1/32	Floating IP address IKE Gateway address
			10.11.0.1/32	IKE Gateway address
	ge-0/0/2	ICL	10.22.0.1/24	Connecting ICL
	ge-0/0/3	Untrust	10.2.0.2/24	Connects to R1 router
	ge-0/0/4	Trust	10.4.0.1/24	Connects to R2 router
SRX-03	lo0	Untrust	10.112.0.1/32	IKE Gateway address
			10.112.0.5/32	IKE Gateway address
	ge-0/0/0	Untrust	10.7.0.1/24	Connects to R2 router
	ge-0/0/2	Trust	10.6.0.2/24	Connects to client device

Table 50: Interfaces and IP Address Configuration on Routing Devices

Device	Interface	IP Address	Configured for
R2	lo0	10.111.0.2/32	Loopback interface address of R2

Table 50: Interfaces and IP Address Configuration on Routing Devices *(Continued)*

Device	Interface	IP Address	Configured for
	ge-0/0/1	10.4.0.2/24	Connects to SRX-02
	ge-0/0/0	10.5.0.2/24	Connects to SRX-01
	ge-0/0/2	10.7.0.2/24	Connects to SRX-03 (VPN peer device)
R1	lo0	10.111.0.1/32	Loopback interface address of R1
	ge-0/0/0	10.3.0.1/24	Connects to SRX-01
	ge-0/0/1	10.2.0.1/24	Connects to SRX-02

Configuration

IN THIS SECTION

Before You Begin

Junos IKE package is required on your SRX Series Firewalls for Multinode High Availability configuration. This package is available as a default package or as an optional package on SRX Series Firewalls. See Support for Junos IKE Package for details.

If the package is not installed by default on your SRX Series firewall, use the following command to install it. 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.

SRX-01 Device

```
set groups vpn_config when peers SRX-01
set groups vpn_config when peers SRX-02
set groups vpn_config security ike proposal SRG1_IKE_PROP authentication-method pre-shared-keys
set groups vpn_config security ike proposal SRG1_IKE_PROP dh-group group14
set groups vpn_config security ike proposal SRG1_IKE_PROP authentication-algorithm sha-256
set groups vpn_config security ike proposal SRG1_IKE_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ike proposal SRG1_IKE_PROP lifetime-seconds 3600
set groups vpn_config security ike proposal SRG2_IKE_PROP authentication-method pre-shared-keys
set groups vpn_config security ike proposal SRG2_IKE_PROP dh-group group14
set groups vpn_config security ike proposal SRG2_IKE_PROP authentication-algorithm sha-256
set groups vpn_config security ike proposal SRG2_IKE_PROP encryption-algorithm aes-256-cbc
```

```
set groups vpn_config security ike proposal SRG2_IKE_PROP lifetime-seconds 3600
set groups vpn_config security ike policy SRG1_IKE_POL1 proposals SRG1_IKE_PROP
set groups vpn_config security ike policy SRG1_IKE_POL1 pre-shared-key ascii-text "$ABC123"
set groups vpn_config security ike policy SRG2_IKE_POL500 proposals SRG2_IKE_PROP
set groups vpn_config security ike policy SRG2_IKE_POL500 pre-shared-key ascii-text "$ABC123"
set groups vpn_config security ike gateway SRG1_IKE_GW1 ike-policy SRG1_IKE_POL1
set groups vpn_config security ike gateway SRG1_IKE_GW1 address 10.112.0.1
set groups vpn_config security ike gateway SRG1_IKE_GW1 external-interface lo0
set groups vpn_config security ike gateway SRG1_IKE_GW1 local-address 10.11.0.1
set groups vpn_config security ike gateway SRG1_IKE_GW1 version v2-only
set groups vpn_config security ike gateway SRG2_IKE_GW500 ike-policy SRG2_IKE_POL500
set groups vpn_config security ike gateway SRG2_IKE_GW500 address 10.112.0.5
set groups vpn_config security ike gateway SRG2_IKE_GW500 external-interface lo0
set groups vpn_config security ike gateway SRG2_IKE_GW500 local-address 10.12.0.1
set groups vpn_config security ike gateway SRG2_IKE_GW500 version v2-only
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP protocol esp
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP lifetime-seconds 1800
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP protocol esp
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP lifetime-seconds 1800
set groups vpn_config security ipsec policy SRG1_IPSEC_POL1 proposals SRG1_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL501 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL500 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL502 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL503 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 bind-interface st0.1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 ike gateway SRG1_IKE_GW1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 ike ipsec-policy SRG1_IPSEC_POL1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 local-ip
10.1.0.2/32
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 remote-ip
10.7.0.2/32
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 establish-tunnels on-traffic
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 bind-interface st0.500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 ike gateway SRG2_IKE_GW500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 ike ipsec-policy SRG2_IPSEC_POL500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts500 local-ip
10.8.0.2/32
```

```
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts500 remote-ip
10.9.0.2/32
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 establish-tunnels on-traffic
set groups vpn_config security zones security-zone vpn host-inbound-traffic system-services ike
set groups vpn_config security zones security-zone vpn host-inbound-traffic protocols all
set groups vpn_config security zones security-zone vpn interfaces st0.1
set groups vpn_config security zones security-zone vpn interfaces st0.500
set groups vpn_config interfaces st0 unit 1 family inet
set groups vpn_config interfaces st0 unit 1 family inet6
set groups vpn_config interfaces st0 unit 500 family inet
set groups vpn_config interfaces st0 unit 500 family inet6
set apply-groups vpn_config
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 10.22.0.2
set chassis high-availability peer-id 2 peer-ip 10.22.0.1
set chassis high-availability peer-id 2 interface ge-0/0/2.0
set chassis high-availability peer-id 2 vpn-profile ICL_IPSEC_VPN
set chassis high-availability peer-id 2 liveness-detection minimum-interval 200
set chassis high-availability peer-id 2 liveness-detection multiplier 3
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 10.5.0.2 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-0/0/3.0
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 prefix-list SRG1_PFX
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 preemption
set chassis high-availability services-redundancy-group 1 activeness-priority 1
set chassis high-availability services-redundancy-group 2 peer-id 2
set chassis high-availability services-redundancy-group 2 activeness-probe dest-ip 10.111.0.1
set chassis high-availability services-redundancy-group 2 activeness-probe dest-ip src-ip
10.12.0.1
set chassis high-availability services-redundancy-group 2 monitor bfd-liveliness 10.5.0.2 src-ip
```

```
10.5.0.1
set chassis high-availability services-redundancy-group 2 monitor bfd-liveliness 10.5.0.2
session-type singlehop
set chassis high-availability services-redundancy-group 2 monitor bfd-liveliness 10.5.0.2
interface ge-0/0/3.0
set chassis high-availability services-redundancy-group 2 monitor interface ge-0/0/3
set chassis high-availability services-redundancy-group 2 monitor interface ge-0/0/4
set chassis high-availability services-redundancy-group 2 active-signal-route 10.49.1.1
set chassis high-availability services-redundancy-group 2 backup-signal-route 10.49.1.2
set chassis high-availability services-redundancy-group 2 prefix-list SRG2_PFX
set chassis high-availability services-redundancy-group 2 managed-services ipsec
set chassis high-availability services-redundancy-group 2 preemption
set chassis high-availability services-redundancy-group 2 activeness-priority 200
set security ike proposal ICL_IKE_PROP description interchassis_link_encr_tunnel
set security ike proposal ICL_IKE_PROP authentication-method pre-shared-keys
set security ike proposal ICL_IKE_PROP dh-group group14
set security ike proposal ICL_IKE_PROP authentication-algorithm sha-256
set security ike proposal ICL_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal ICL_IKE_PROP lifetime-seconds 300
set security ike policy ICL_IKE_POL description interchassis_link_encr_tunnel
set security ike policy ICL_IKE_POL proposals ICL_IKE_PROP
set security ike policy ICL_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway ICL_IKE_GW ike-policy ICL_IKE_POL
set security ike gateway ICL_IKE_GW version v2-only
set security ipsec proposal ICL_IPSEC_PROP description interchassis_link_encr_tunnel
set security ipsec proposal ICL_IPSEC_PROP protocol esp
set security ipsec proposal ICL_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal ICL_IPSEC_PROP lifetime-seconds 300
set security ipsec policy ICL_IPSEC_POL description interchassis_link_encr_tunnel
set security ipsec policy ICL_IPSEC_POL proposals ICL_IPSEC_PROP
set security ipsec vpn ICL_IPSEC_VPN ha-link-encryption
set security ipsec vpn ICL_IPSEC_VPN ike gateway ICL_IKE_GW
set security ipsec vpn ICL_IPSEC_VPN ike ipsec-policy ICL_IPSEC_POL
set security policies default-policy permit-all
set security zones security-zone vpn host-inbound-traffic system-services ike
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.1
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 lo0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
```

```
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-0/0/4.0
set security zones security-zone icl_zone host-inbound-traffic system-services ike
set security zones security-zone icl_zone host-inbound-traffic system-services ping
set security zones security-zone icl_zone host-inbound-traffic system-services high-availability
set security zones security-zone icl_zone host-inbound-traffic system-services ssh
set security zones security-zone icl_zone host-inbound-traffic protocols bfd
set security zones security-zone icl_zone host-inbound-traffic protocols bgp
set security zones security-zone icl_zone interfaces ge-0/0/2.0
set interfaces ge-0/0/1 unit 0 family inet
set interfaces ge-0/0/2 description inter_chassis_link
set interfaces ge-0/0/2 unit 0 family inet address 10.22.0.2/24
set interfaces ge-0/0/3 description untrust
set interfaces ge-0/0/3 unit 0 family inet address 10.3.0.2/24
set interfaces ge-0/0/4 description trust
set interfaces ge-0/0/4 unit 0 family inet address 10.5.0.1/24
set interfaces lo0 apply-groups-except global
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 10.12.0.1/32
set interfaces st0 unit 1
set policy-options prefix-list SRG1_PFX 10.11.0.0/24
set policy-options prefix-list SRG2_PFX 10.12.0.0/24
set policy-options route-filter-list srg1_rf_list 10.11.0.0/24 orlonger
set policy-options route-filter-list srg1_rf_list 10.7.0.0/16 orlonger
set policy-options route-filter-list srg1_rf_list 10.1.0.0/16 orlonger
set policy-options route-filter-list srg2_rf_list 10.12.0.0/24 orlonger
set policy-options route-filter-list srg2_rf_list 10.9.0.0/16 orlonger
set policy-options route-filter-list srg2_rf_list 10.8.0.0/16 orlonger
set policy-options policy-statement mnha-route-policy term 1 from route-filter-list srg1_rf_list
set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists_srg1
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 route-filter-list srg1_rf_list
set policy-options policy-statement mnha-route-policy term 2 from condition
backup_route_exists_srg1
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 route-filter-list srg2_rf_list
set policy-options policy-statement mnha-route-policy term 3 from condition
active_route_exists_srg2
```

```
set policy-options policy-statement mnha-route-policy term 3 then metric 10
set policy-options policy-statement mnha-route-policy term 3 then accept
set policy-options policy-statement mnha-route-policy term 4 from route-filter-list srg2_rf_list
set policy-options policy-statement mnha-route-policy term 4 from condition
backup_route_exists_srg2
set policy-options policy-statement mnha-route-policy term 4 then metric 20
set policy-options policy-statement mnha-route-policy term 4 then accept
set policy-options policy-statement mnha-route-policy term default then reject
set policy-options condition active_route_exists_srg1 if-route-exists address-family inet
10.39.1.1/32
set policy-options condition active_route_exists_srg1 if-route-exists address-family inet table
inet.0
set policy-options condition active_route_exists_srg2 if-route-exists address-family inet
10.49.1.1/32
set policy-options condition active_route_exists_srg2 if-route-exists address-family inet table
inet.0
set policy-options condition backup_route_exists_srg1 if-route-exists address-family inet
10.39.1.2/32
set policy-options condition backup_route_exists_srg1 if-route-exists address-family inet table
inet.0
set policy-options condition backup_route_exists_srg2 if-route-exists address-family inet
10.49.1.2/32
set policy-options condition backup_route_exists_srg2 if-route-exists address-family inet table
inet.0
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 local-as 100
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 neighbor 10.3.0.1
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 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 10.5.0.2
set routing-options autonomous-system 100
```

```
set routing-options static route 10.7.0.0/16 next-hop 10.5.0.2
set routing-options static route 10.112.0.0/24 next-hop 10.5.0.2
```

SRX-02 Device

```
set groups vpn_config when peers SRX-01
set groups vpn_config when peers SRX-02
set groups vpn_config security ike proposal SRG1_IKE_PROP authentication-method pre-shared-keys
set groups vpn_config security ike proposal SRG1_IKE_PROP dh-group group14
set groups vpn_config security ike proposal SRG1_IKE_PROP authentication-algorithm sha-256
set groups vpn_config security ike proposal SRG1_IKE_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ike proposal SRG1_IKE_PROP lifetime-seconds 3600
set groups vpn_config security ike proposal SRG2_IKE_PROP authentication-method pre-shared-keys
set groups vpn_config security ike proposal SRG2_IKE_PROP dh-group group14
set groups vpn_config security ike proposal SRG2_IKE_PROP authentication-algorithm sha-256
set groups vpn_config security ike proposal SRG2_IKE_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ike proposal SRG2_IKE_PROP lifetime-seconds 3600
set groups vpn_config security ike policy SRG1_IKE_POL1 proposals SRG1_IKE_PROP
set groups vpn_config security ike policy SRG1_IKE_POL1 pre-shared-key ascii-text "$ABC123"
set groups vpn_config security ike policy SRG2_IKE_POL500 proposals SRG2_IKE_PROP
set groups vpn_config security ike policy SRG2_IKE_POL500 pre-shared-key ascii-text "$ABC123"
set groups vpn_config security ike gateway SRG1_IKE_GW1 ike-policy SRG1_IKE_POL1
set groups vpn_config security ike gateway SRG1_IKE_GW1 address 10.112.0.1
set groups vpn_config security ike gateway SRG1_IKE_GW1 external-interface lo0
set groups vpn_config security ike gateway SRG1_IKE_GW1 local-address 10.11.0.1
set groups vpn_config security ike gateway SRG1_IKE_GW1 version v2-only
set groups vpn_config security ike gateway SRG2_IKE_GW500 ike-policy SRG2_IKE_POL500
set groups vpn_config security ike gateway SRG2_IKE_GW500 address 10.112.0.5
set groups vpn_config security ike gateway SRG2_IKE_GW500 external-interface lo0
set groups vpn_config security ike gateway SRG2_IKE_GW500 local-address 10.12.0.1
set groups vpn_config security ike gateway SRG2_IKE_GW500 version v2-only
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP protocol esp
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP lifetime-seconds 1800
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP protocol esp
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP lifetime-seconds 1800
set groups vpn_config security ipsec policy SRG1_IPSEC_POL1 proposals SRG1_IPSEC_PROP
```

```
set groups vpn_config security ipsec policy SRG2_IPSEC_POL501 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL500 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL502 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL503 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 bind-interface st0.1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 ike gateway SRG1_IKE_GW1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 ike ipsec-policy SRG1_IPSEC_POL1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 local-ip
10.1.0.2/32
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 remote-ip
10.7.0.2/32
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 establish-tunnels on-traffic
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 bind-interface st0.500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 ike gateway SRG2_IKE_GW500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 ike ipsec-policy SRG2_IPSEC_POL500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts500 local-ip
10.8.0.2/32
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts500 remote-ip
10.9.0.2/32
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 establish-tunnels on-traffic
set groups vpn_config security zones security-zone vpn host-inbound-traffic system-services ike
set groups vpn_config security zones security-zone vpn host-inbound-traffic protocols all
set groups vpn_config security zones security-zone vpn interfaces st0.1
set groups vpn_config security zones security-zone vpn interfaces st0.500
set groups vpn_config interfaces st0 unit 1 family inet
set groups vpn_config interfaces st0 unit 1 family inet6
set groups vpn_config interfaces st0 unit 500 family inet
set groups vpn_config interfaces st0 unit 500 family inet6
set apply-groups vpn_config
set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 10.22.0.1
set chassis high-availability peer-id 1 peer-ip 10.22.0.2
set chassis high-availability peer-id 1 interface ge-0/0/2.0
set chassis high-availability peer-id 1 vpn-profile ICL_IPSEC_VPN
set chassis high-availability peer-id 1 liveness-detection minimum-interval 200
set chassis high-availability peer-id 1 liveness-detection multiplier 3
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 10.4.0.2 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-0/0/3.0
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 prefix-list SRG1_PFX
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 preemption
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set chassis high-availability services-redundancy-group 2 peer-id 1
set chassis high-availability services-redundancy-group 2 activeness-probe dest-ip 10.111.0.1
set chassis high-availability services-redundancy-group 2 activeness-probe dest-ip src-ip
10.12.0.1
set chassis high-availability services-redundancy-group 2 monitor bfd-liveliness 10.4.0.2 src-ip
10.4.0.1
set chassis high-availability services-redundancy-group 2 monitor bfd-liveliness 10.4.0.2
session-type singlehop
set chassis high-availability services-redundancy-group 2 monitor bfd-liveliness 10.4.0.2
interface ge-0/0/3.0
set chassis high-availability services-redundancy-group 2 monitor interface ge-0/0/3
set chassis high-availability services-redundancy-group 2 monitor interface ge-0/0/4
set chassis high-availability services-redundancy-group 2 active-signal-route 10.49.1.1
set chassis high-availability services-redundancy-group 2 backup-signal-route 10.49.1.2
set chassis high-availability services-redundancy-group 2 prefix-list SRG2_PFX
set chassis high-availability services-redundancy-group 2 managed-services ipsec
set chassis high-availability services-redundancy-group 2 preemption
set chassis high-availability services-redundancy-group 2 activeness-priority 1
set security ike proposal ICL_IKE_PROP description interchassisa_link_encr_tunnel
set security ike proposal ICL_IKE_PROP authentication-method pre-shared-keys
set security ike proposal ICL_IKE_PROP dh-group group14
set security ike proposal ICL_IKE_PROP authentication-algorithm sha-256
set security ike proposal ICL_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal ICL_IKE_PROP lifetime-seconds 300
set security ike policy ICL_IKE_POL description interchassis_link_encr_tunnel
set security ike policy ICL_IKE_POL proposals ICL_IKE_PROP
set security ike policy ICL_IKE_POL pre-shared-key ascii-text "$ABC123"
set security ike gateway ICL_IKE_GW ike-policy ICL_IKE_POL
set security ike gateway ICL_IKE_GW version v2-only
set security ipsec proposal ICL_IPSEC_PROP description interchassis_link_encr_tunnel
set security ipsec proposal ICL_IPSEC_PROP protocol esp
```

```
set security ipsec proposal ICL_IPSEC_PROP encryption-algorithm aes-256-gcm
set security ipsec proposal ICL_IPSEC_PROP lifetime-seconds 300
set security ipsec policy ICL_IPSEC_POL description interchassis_link_encr_tunnel
set security ipsec policy ICL_IPSEC_POL proposals ICL_IPSEC_PROP
set security ipsec vpn ICL_IPSEC_VPN ha-link-encryption
set security ipsec vpn ICL_IPSEC_VPN ike gateway ICL_IKE_GW
set security ipsec vpn ICL_IPSEC_VPN ike ipsec-policy ICL_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 lo0.0
set security zones security-zone untrust interfaces ge-0/0/3.0
set security zones security-zone vpn host-inbound-traffic system-services ike
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.1
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-0/0/4.0
set security zones security-zone icl_zone host-inbound-traffic system-services ike
set security zones security-zone icl_zone host-inbound-traffic system-services ping
set security zones security-zone icl_zone host-inbound-traffic system-services high-availability
set security zones security-zone icl_zone host-inbound-traffic system-services ssh
set security zones security-zone icl_zone host-inbound-traffic protocols bfd
set security zones security-zone icl_zone host-inbound-traffic protocols bgp
set security zones security-zone icl_zone interfaces ge-0/0/2.0
set interfaces ge-0/0/1 unit 0 family inet
set interfaces ge-0/0/2 description inter_chassis_link
set interfaces ge-0/0/2 unit 0 family inet address 10.22.0.1/24
set interfaces ge-0/0/3 description untrust
set interfaces ge-0/0/3 unit 0 family inet address 10.2.0.2/24
set interfaces ge-0/0/4 description trust
set interfaces ge-0/0/4 unit 0 family inet address 10.4.0.1/24
set interfaces lo0 apply-groups-except global
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 10.12.0.1/32
set interfaces st0 unit 1 family inet
set interfaces st0 unit 1 family inet6
set policy-options prefix-list SRG1_PFX 10.11.0.0/24
set policy-options prefix-list SRG2_PFX 10.12.0.0/24
set policy-options route-filter-list srg1_rf_list 10.11.0.0/24 orlonger
```

```
set policy-options route-filter-list srg1_rf_list 10.7.0.0/24 orlonger
set policy-options route-filter-list srg1_rf_list 10.1.0.0/24 orlonger
set policy-options route-filter-list srg2_rf_list 10.12.0.0/24 orlonger
set policy-options route-filter-list srg2_rf_list 10.9.0.0/24 orlonger
set policy-options route-filter-list srg2_rf_list 10.8.0.0/24 orlonger
set policy-options policy-statement mnha-route-policy term 1 from route-filter-list srg1_rf_list
set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists_srg1
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 route-filter-list srg1_rf_list
set policy-options policy-statement mnha-route-policy term 2 from condition
backup_route_exists_srg1
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 route-filter-list srg2_rf_list
set policy-options policy-statement mnha-route-policy term 3 from condition
active_route_exists_srg2
set policy-options policy-statement mnha-route-policy term 3 then metric 10
set policy-options policy-statement mnha-route-policy term 3 then accept
set policy-options policy-statement mnha-route-policy term 4 from route-filter-list srg2_rf_list
set policy-options policy-statement mnha-route-policy term 4 from condition
backup_route_exists_srg2
set policy-options policy-statement mnha-route-policy term 4 then metric 20
set policy-options policy-statement mnha-route-policy term 4 then accept
set policy-options policy-statement mnha-route-policy term default then reject
set policy-options condition active_route_exists_srg1 if-route-exists address-family inet
10.39.1.1/32
set policy-options condition active_route_exists_srg1 if-route-exists address-family inet table
inet.0
set policy-options condition active_route_exists_srg2 if-route-exists address-family inet
10.49.1.1/32
set policy-options condition active_route_exists_srg2 if-route-exists address-family inet table
inet.0
set policy-options condition backup_route_exists_srg1 if-route-exists address-family inet
10.39.1.2/32
set policy-options condition backup_route_exists_srg1 if-route-exists address-family inet table
inet.0
set policy-options condition backup_route_exists_srg2 if-route-exists address-family inet
10.49.1.2/32
set policy-options condition backup_route_exists_srg2 if-route-exists address-family inet table
inet.0
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 local-as 100
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 neighbor 10.2.0.1
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 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 10.4.0.2
set routing-options autonomous-system 100
set routing-options static route 10.7.0.0/24 next-hop 10.4.0.2
set routing-options static route 10.112.0.0/24 next-hop 10.4.0.2
```

SRX-3 Device

```
set security ike proposal SRG1_IKE_PROP authentication-method pre-shared-keys
set security ike proposal SRG1_IKE_PROP dh-group group14
set security ike proposal SRG1_IKE_PROP authentication-algorithm sha-256
set security ike proposal SRG1_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SRG1_IKE_PROP lifetime-seconds 3600
set security ike proposal SRG2_IKE_PROP authentication-method pre-shared-keys
set security ike proposal SRG2_IKE_PROP dh-group group14
set security ike proposal SRG2_IKE_PROP authentication-algorithm sha-256
set security ike proposal SRG2_IKE_PROP encryption-algorithm aes-256-cbc
set security ike proposal SRG2_IKE_PROP lifetime-seconds 3600
set security ike policy SRG1_IKE_POL1 proposals SRG1_IKE_PROP
set security ike policy SRG1_IKE_POL1 pre-shared-key ascii-text "$ABC123"
set security ike policy SRG2_IKE_POL500 proposals SRG2_IKE_PROP
set security ike policy SRG2_IKE_POL500 pre-shared-key ascii-text "$ABC123"
set security ike gateway SRG1_IKE_GW1 ike-policy SRG1_IKE_POL1
set security ike gateway SRG1_IKE_GW1 address 10.11.0.1
set security ike gateway SRG1_IKE_GW1 external-interface lo0
set security ike gateway SRG1_IKE_GW1 local-address 10.112.0.1
set security ike gateway SRG1_IKE_GW1 version v2-only
set security ike gateway SRG2_IKE_GW500 ike-policy SRG2_IKE_POL500
set security ike gateway SRG2_IKE_GW500 address 10.12.0.1
```

```
set security ike gateway SRG2_IKE_GW500 external-interface lo0
set security ike gateway SRG2_IKE_GW500 local-address 10.112.0.5
set security ike gateway SRG2_IKE_GW500 version v2-only
set security ipsec proposal SRG1_IPSEC_PROP protocol esp
set security ipsec proposal SRG1_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SRG1_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal SRG1_IPSEC_PROP lifetime-seconds 1800
set security ipsec proposal SRG2_IPSEC_PROP protocol esp
set security ipsec proposal SRG2_IPSEC_PROP authentication-algorithm hmac-sha-256-128
set security ipsec proposal SRG2_IPSEC_PROP encryption-algorithm aes-256-cbc
set security ipsec proposal SRG2_IPSEC_PROP lifetime-seconds 1800
set security ipsec policy SRG1_IPSEC_POL1 proposals SRG1_IPSEC_PROP
set security ipsec policy SRG2_IPSEC_POL500 proposals SRG2_IPSEC_PROP
set security ipsec vpn SRG1_IPSEC_VPN1 bind-interface st0.1
set security ipsec vpn SRG1_IPSEC_VPN1 ike gateway SRG1_IKE_GW1
set security ipsec vpn SRG1_IPSEC_VPN1 ike ipsec-policy SRG1_IPSEC_POL1
set security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 local-ip 10.7.0.2/32
set security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 remote-ip 10.1.0.2/32
set security ipsec vpn SRG1_IPSEC_VPN1 establish-tunnels immediately
set security ipsec vpn SRG2_IPSEC_VPN500 bind-interface st0.500
set security ipsec vpn SRG2_IPSEC_VPN500 ike gateway SRG2_IKE_GW500
set security ipsec vpn SRG2_IPSEC_VPN500 ike ipsec-policy SRG2_IPSEC_POL500
set security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts1 local-ip 10.9.0.2/32
set security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts1 remote-ip 10.8.0.2/32
set security ipsec vpn SRG2_IPSEC_VPN500 establish-tunnels immediately
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 protocols all
set security zones security-zone untrust interfaces st0.1
set security zones security-zone untrust interfaces lo0.0
set security zones security-zone untrust interfaces st0.500
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/0.0
set security zones security-zone trust host-inbound-traffic system-services ike
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces ge-0/0/2.0
set interfaces ge-0/0/0 description trust
set interfaces ge-0/0/0 unit 0 family inet address 10.7.0.1/24
set interfaces ge-0/0/1 description untrust
set interfaces ge-0/0/1 unit 0 family inet address 10.6.0.2/24
set interfaces ge-0/0/2 description trust
set interfaces ge-0/0/2 unit 0 family inet address 10.9.0.1/24
set interfaces lo0 description untrust
```

```
set interfaces lo0 unit 0 family inet address 10.112.0.1/32
set interfaces lo0 unit 0 family inet address 10.112.0.5/32
set interfaces st0 unit 1 family inet
set interfaces st0 unit 500 family inet
set routing-options autonomous-system 100
set routing-options static route 10.4.0.0/24 next-hop 10.7.0.2
set routing-options static route 10.5.0.0/24 next-hop 10.7.0.2
set routing-options static route 10.11.0.0/24 next-hop 10.7.0.2
set routing-options static route 10.12.0.0/24 next-hop 10.7.0.2
set routing-options static route 10.111.0.1/32 next-hop 10.7.0.2
set routing-options static route 10.111.0.2/32 next-hop 10.7.0.2
```

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

R1 Router

```
set interfaces ge-0/0/0 description srx_1
set interfaces ge-0/0/0 unit 0 family inet address 10.3.0.1/24
set interfaces ge-0/0/1 description srx_2
set interfaces ge-0/0/1 unit 0 family inet address 10.2.0.1/24
set interfaces lo0 description loopback
set interfaces lo0 unit 0 family inet address 10.111.0.1/32 primary
set interfaces lo0 unit 0 family inet address 10.111.0.1/32 preferred
set routing-options autonomous-system 100
set protocols bgp group srx2_group type internal
set protocols bgp group srx2_group local-address 10.2.0.1
set protocols bgp group srx2_group local-as 100
set protocols bgp group srx2_group bfd-liveness-detection minimum-interval 500
set protocols bgp group srx2_group bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group srx2_group bfd-liveness-detection multiplier 3
set protocols bgp group srx2_group neighbor 10.2.0.2
set protocols bgp group srx1_group type internal
set protocols bgp group srx1_group local-address 10.3.0.1
set protocols bgp group srx1_group local-as 100
set protocols bgp group srx1_group bfd-liveness-detection minimum-interval 500
set protocols bgp group srx1_group bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group srx1_group bfd-liveness-detection multiplier 3
set protocols bgp group srx1_group neighbor 10.3.0.2
```

R2 Router

```
set interfaces ge-0/0/0 description srx_1
set interfaces ge-0/0/0 unit 0 family inet address 10.5.0.2/24
set interfaces ge-0/0/1 description srx_2
set interfaces ge-0/0/1 unit 0 family inet address 10.4.0.2/24
set interfaces ge-0/0/2 description srx-3
set interfaces ge-0/0/2 unit 0 family inet address 10.7.0.2/24
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 routing-options static route 10.112.0.0/24 next-hop 10.7.0.1
set protocols bgp group srx2_group type internal
set protocols bgp group srx2_group local-address 10.4.0.2
set protocols bgp group srx2_group local-as 100
set protocols bgp group srx2_group bfd-liveness-detection minimum-interval 500
set protocols bgp group srx2_group bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group srx2_group bfd-liveness-detection multiplier 3
set protocols bgp group srx2_group neighbor 10.4.0.1
set protocols bgp group srx1_group type internal
set protocols bgp group srx1_group local-address 10.5.0.2
set protocols bgp group srx1_group local-as 100
set protocols bgp group srx1_group bfd-liveness-detection minimum-interval 500
set protocols bgp group srx1_group bfd-liveness-detection minimum-receive-interval 500
set protocols bgp group srx1_group bfd-liveness-detection multiplier 3
set protocols bgp group srx1_group neighbor 10.5.0.1
```

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@srx-1# set interfaces ge-0/0/2 description inter_chassis_link
```

```
 user@srx-1# set interfaces ge-0/0/2 unit 0 family inet address 10.22.0.2/24
 user@srx-1# set interfaces ge-0/0/3 description untrust
 user@srx-1# set interfaces ge-0/0/3 unit 0 family inet address 10.3.0.2/24
 user@srx-1# set interfaces ge-0/0/4 description trust
 user@srx-1# set interfaces ge-0/0/4 unit 0 family inet address 10.5.0.1/24
```

Use ge-0/0/3 and ge-0/0/4 interfaces to connect to the upstream and downstream routers and use ge-0/0/2 interface to set up the ICL.

2. Configure the loopback interfaces.

```
[edit]
 user@srx-1# set interfaces lo0 apply-groups-except global
 user@srx-1# set interfaces lo0 description untrust
 user@srx-1# set interfaces lo0 unit 0 family inet address 10.11.0.1/32
 user@srx-1# set interfaces lo0 unit 0 family inet address 10.12.0.1/32
 user@srx-1# set interfaces st0 unit 1
```

Assign IP address 10.11.0.1 and 10.12.0.1 to the loopback interface. We'll use 10.11.0.1 as the floating IP address and 10.12.0.1 as IKE gateway address.

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

```
[edit]
 user@srx-1# set security zones security-zone vpn host-inbound-traffic system-services ike
 user@srx-1# set security zones security-zone vpn host-inbound-traffic protocols all
 user@srx-1# set security zones security-zone vpn interfaces st0.1
 user@srx-1# set security zones security-zone untrust host-inbound-traffic system-services
ike
 user@srx-1# set security zones security-zone untrust host-inbound-traffic system-services
ping
 user@srx-1# set security zones security-zone untrust host-inbound-traffic protocols bfd
 user@srx-1# set security zones security-zone untrust host-inbound-traffic protocols bgp
 user@srx-1# set security zones security-zone untrust interfaces lo0.0
 user@srx-1# set security zones security-zone untrust interfaces ge-0/0/3.0
 user@srx-1# set security zones security-zone trust host-inbound-traffic system-services all
 user@srx-1# set security zones security-zone trust host-inbound-traffic protocols all
 user@srx-1# set security zones security-zone trust interfaces ge-0/0/4.0
 user@srx-1# set security zones security-zone icl_zone host-inbound-traffic system-services
ike
```

```
 user@srx-1# set security zones security-zone icl_zone host-inbound-traffic system-services
ping
 user@srx-1# set security zones security-zone icl_zone host-inbound-traffic system-services
high-availability
 user@srx-1# set security zones security-zone icl_zone host-inbound-traffic system-services
ssh
 user@srx-1# set security zones security-zone icl_zone host-inbound-traffic protocols bfd
 user@srx-1# set security zones security-zone icl_zone host-inbound-traffic protocols bgp
 user@srx-1# set security zones security-zone icl_zone interfaces ge-0/0/2.0
```

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 IP network. Assign the interface ge-0/0/2 to the ICL zone. You use this zone to set up the ICL. Assign the secure tunnel interface to the VPN security zone.

4. 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@srx-1# set chassis high-availability local-id 1
user@srx-1# set chassis high-availability local-id local-ip 10.22.0.2
user@srx-1# set chassis high-availability peer-id 2 peer-ip 10.22.0.1
user@srx-1# 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.

5. Attach the IPsec VPN profile IPSEC_VPN_ICL to the peer node.

```
[edit]
user@srx-1# set chassis high-availability peer-id 2 vpn-profile ICL_IPSEC_VPN
```

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

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

```
[edit]
user@srx-1# set chassis high-availability peer-id 2 liveness-detection minimum-interval 200
user@srx-1# set chassis high-availability peer-id 2 liveness-detection multiplier 3
```

7. Configure the services redundancy groups SRG1 and SRG2.

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

8. Setup activeness determination parameters both SRG1 and SRG2.

SRG1

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

SRG2

```
[edit]
user@srx-1# set chassis high-availability services-redundancy-group 2 activeness-probe dest-
ip 10.111.0.1
user@srx-1# set chassis high-availability services-redundancy-group 2 activeness-probe dest-
ip src-ip 10.11.0.1
```

Use the floating IP address as source IP address (10.11.0.1 for SRG1 and 10.12.0.1 for SRG2) 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)

9. Configure BFD monitoring parameters for the SRG1 and SRG2 to detect failures in network.

SRG1

```
[edit]
user@srx-1# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.5.0.2 src-ip 10.5.0.1
user@srx-1# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.5.0.2 session-type singlehop
user@srx-1# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.5.0.2 interface ge-0/0/3.0
user@srx-1# set chassis high-availability services-redundancy-group 1 monitor interface
ge-0/0/3
user@srx-1# set chassis high-availability services-redundancy-group 1 monitor interface
ge-0/0/4
```

SRG2

```
[edit]
user@srx-1# set chassis high-availability services-redundancy-group 2 monitor bfd-
liveliness 10.5.0.2 src-ip 10.5.0.1
user@srx-1# set chassis high-availability services-redundancy-group 2 monitor bfd-
liveliness 10.5.0.2 session-type singlehop
user@srx-1# set chassis high-availability services-redundancy-group 2 monitor bfd-
liveliness 10.5.0.2 interface ge-0/0/3.0
user@srx-1# set chassis high-availability services-redundancy-group 2 monitor interface
ge-0/0/3
user@srx-1# set chassis high-availability services-redundancy-group 2 monitor interface
ge-0/0/4
```

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

SRG1

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

SRG2

```
[edit]
user@srx-1# set chassis high-availability services-redundancy-group 2 active-signal-route
10.49.1.1
user@srx-1# set chassis high-availability services-redundancy-group 2 backup-signal-route
10.49.1.2
user@srx-1# set chassis high-availability services-redundancy-group 2 preemption
user@srx-1# set chassis high-availability services-redundancy-group 2 activeness-priority
200
```

NOTE: You must specify the active signal route along with the route-exists policy in the policy-options statement. When you configure the `active-signal-route` with `if-route-exists` condition, the HA module adds this route to the routing table.

11. Create an IP prefix list by including the local address of IKE gateway and associate the IP prefix list to SRG1 and SRG2:

    SRG1

    ```
    [edit]
    user@srx-1# set policy-options prefix-list SRG1_PFX 10.11.0.0/24
    user@srx-1# set chassis high-availability services-redundancy-group 1 prefix-list SRG1_PFX
    ```

    SRG2

    ```
    [edit]
    user@srx-1# set policy-options prefix-list SRG2_PFX 10.12.0.0/24
    user@srx-1# set chassis high-availability services-redundancy-group 2 prefix-list SRG2_PFX
    ```

    This configuration anchors a certain IPsec VPN tunnel to a particular security device.

12. Enable IPsec VPN service on both SRG1 and SRG2.

    ```
    [edit]
    user@srx-1# set chassis high-availability services-redundancy-group 1 managed-services ipsec
    user@srx-1# set chassis high-availability services-redundancy-group 2 managed-services ipsec
    ```

13. Configure IPSec VPN options for the ICL.

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

```
[edit]
 user@srx-1# set security ike proposal ICL_IKE_PROP description
interchassis_link_encr_tunnel
 user@srx-1# set security ike proposal ICL_IKE_PROP authentication-method pre-shared-keys
 user@srx-1# set security ike proposal ICL_IKE_PROP dh-group group14
 user@srx-1# set security ike proposal ICL_IKE_PROP authentication-algorithm sha-256
 user@srx-1# set security ike proposal ICL_IKE_PROP encryption-algorithm aes-256-cbc
 user@srx-1# set security ike proposal ICL_IKE_PROP lifetime-seconds 300
 user@srx-1# set security ike policy ICL_IKE_POL description
interchassis_link_encr_tunnel
 user@srx-1# set security ike policy ICL_IKE_POL proposals ICL_IKE_PROP
 user@srx-1# set security ike policy ICL_IKE_POL pre-shared-key ascii-text "$ABC123"
 user@srx-1# set security ike gateway ICL_IKE_GW ike-policy ICL_IKE_POL
 user@srx-1# set security ike gateway ICL_IKE_GW version v2-only
```

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

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

```
[edit]
user@srx-1# set security ipsec proposal ICL_IPSEC_PROP description
interchassis_link_encr_tunnel
 user@srx-1# set security ipsec proposal ICL_IPSEC_PROP protocol esp
 user@srx-1# set security ipsec proposal ICL_IPSEC_PROP encryption-algorithm aes-256-gcm
 user@srx-1# set security ipsec proposal ICL_IPSEC_PROP lifetime-seconds 300
 user@srx-1# set security ipsec policy ICL_IPSEC_POL description
interchassis_link_encr_tunnel
 user@srx-1# set security ipsec policy ICL_IPSEC_POL proposals ICL_IPSEC_PROP
 user@srx-1# set security ipsec vpn ICL_IPSEC_VPN ha-link-encryption
 user@srx-1# set security ipsec vpn ICL_IPSEC_VPN ike gateway ICL_IKE_GW
 user@srx-1# set security ipsec vpn ICL_IPSEC_VPN ike ipsec-policy ICL_IPSEC_POL
```

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

The same VPN name ICL_IPSEC_VPN must be mentioned for *vpn_profile* in the `set chassis high-availability peer-id <id> vpn-profile` *vpn_profile* configuration.

14. Configure the security policy.

```
[edit]
user@srx-1# set security policies default-policy permit-all
```

For this example, we've configured a policy to permit all traffic. We strongly recommend you to create security policies as per your network requirements to permit traffic that is allowed by your organizational policy and deny all other traffic. We've used the default policy for the demo purpose only in this example.

15. Configure routing options.

```
[edit]
 user@srx-1# set routing-options autonomous-system 100
 user@srx-1# set routing-options static route 10.7.0.0/16 next-hop 10.5.0.2
 user@srx-1# set routing-options static route 10.112.0.0/24 next-hop 10.5.0.2
```

16. Configure policy options.

```
[edit]
 user@srx-1# set policy-options route-filter-list srg1_rf_list 10.11.0.0/24 orlonger
 user@srx-1# set policy-options route-filter-list srg1_rf_list 10.7.0.0/16 orlonger
 user@srx-1# set policy-options route-filter-list srg1_rf_list 10.1.0.0/16 orlonger
 user@srx-1# set policy-options route-filter-list srg2_rf_list 10.12.0.0/24 orlonger
 user@srx-1# set policy-options route-filter-list srg2_rf_list 10.9.0.0/16 orlonger
 user@srx-1# set policy-options route-filter-list srg2_rf_list 10.8.0.0/16 orlonger
 user@srx-1# set policy-options policy-statement mnha-route-policy term 1 from route-filter-
list srg1_rf_list
 user@srx-1# set policy-options policy-statement mnha-route-policy term 1 from condition
active_route_exists_srg1
 user@srx-1# set policy-options policy-statement mnha-route-policy term 1 then metric 10
 user@srx-1# set policy-options policy-statement mnha-route-policy term 1 then accept
 user@srx-1# set policy-options policy-statement mnha-route-policy term 2 from route-filter-
list srg1_rf_list
 user@srx-1# set policy-options policy-statement mnha-route-policy term 2 from condition
backup_route_exists_srg1
 user@srx-1# set policy-options policy-statement mnha-route-policy term 2 then metric 20
 user@srx-1# set policy-options policy-statement mnha-route-policy term 2 then accept
 user@srx-1# set policy-options policy-statement mnha-route-policy term 3 from route-filter-
list srg2_rf_list
 user@srx-1# set policy-options policy-statement mnha-route-policy term 3 from condition
```

```
active_route_exists_srg2
 user@srx-1# set policy-options policy-statement mnha-route-policy term 3 then metric 10
 user@srx-1# set policy-options policy-statement mnha-route-policy term 3 then accept
 user@srx-1# set policy-options policy-statement mnha-route-policy term 4 from route-filter-
list srg2_rf_list
 user@srx-1# set policy-options policy-statement mnha-route-policy term 4 from condition
backup_route_exists_srg2
 user@srx-1# set policy-options policy-statement mnha-route-policy term 4 then metric 20
 user@srx-1# set policy-options policy-statement mnha-route-policy term 4 then accept
 user@srx-1# set policy-options policy-statement mnha-route-policy term default then reject
 user@srx-1# set policy-options condition active_route_exists_srg1 if-route-exists address-
family inet 10.39.1.1/32
 user@srx-1# set policy-options condition active_route_exists_srg1 if-route-exists address-
family inet table inet.0
 user@srx-1# set policy-options condition active_route_exists_srg2 if-route-exists address-
family inet 10.49.1.1/32
 user@srx-1# set policy-options condition active_route_exists_srg2 if-route-exists address-
family inet table inet.0
 user@srx-1# set policy-options condition backup_route_exists_srg1 if-route-exists address-
family inet 10.39.1.2/32
 user@srx-1# set policy-options condition backup_route_exists_srg1 if-route-exists address-
family inet table inet.0
 user@srx-1# set policy-options condition backup_route_exists_srg2 if-route-exists address-
family inet 10.49.1.2/32
 user@srx-1# set policy-options condition backup_route_exists_srg2 if-route-exists address-
family inet table inet.0
```

Configure the active signal route 10.39.1.1 (SRG1) and 10.49.1.1 (SRG2) 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 backup signal route (10.39.1.2 and 10.49.1.2) 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@srx-1# set protocols bgp group trust type internal
 user@srx-1# set protocols bgp group trust local-address 10.3.0.2
 user@srx-1# set protocols bgp group trust export mnha-route-policy
```

```
 user@srx-1# set protocols bgp group trust local-as 100
 user@srx-1# set protocols bgp group trust bfd-liveness-detection minimum-interval 500
 user@srx-1# set protocols bgp group trust bfd-liveness-detection minimum-receive-interval
500
 user@srx-1# set protocols bgp group trust bfd-liveness-detection multiplier 3
 user@srx-1# set protocols bgp group trust neighbor 10.3.0.1
 user@srx-1# set protocols bgp group untrust type internal
 user@srx-1# set protocols bgp group untrust local-address 10.5.0.1
 user@srx-1# set protocols bgp group untrust export mnha-route-policy
 user@srx-1# set protocols bgp group untrust local-as 100
 user@srx-1# set protocols bgp group untrust bfd-liveness-detection minimum-interval 500
 user@srx-1# set protocols bgp group untrust bfd-liveness-detection minimum-receive-
interval 500
 user@srx-1# set protocols bgp group untrust bfd-liveness-detection multiplier 3
 user@srx-1# set protocols bgp group untrust neighbor 10.5.0.2
```

IPsec VPN Configuration (SRX-1 and SRX-2)

Use the following steps to setup IPsec VPN connection with the peer SRX Series firewall. In this example, you'll be placing all of your IPsec VPN configuration statements inside a JUNOS configuration group named vpn_config.

1. Create a configuration group `vpn_config` at the top of the configuration and configure IPsec VPN specific details.

```
[edit]
set groups vpn_config when peers SRX-01
set groups vpn_config when peers SRX-02
set groups vpn_config security ike proposal SRG1_IKE_PROP authentication-method pre-shared-
keys
set groups vpn_config security ike proposal SRG1_IKE_PROP dh-group group14
set groups vpn_config security ike proposal SRG1_IKE_PROP authentication-algorithm sha-256
set groups vpn_config security ike proposal SRG1_IKE_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ike proposal SRG1_IKE_PROP lifetime-seconds 3600
set groups vpn_config security ike proposal SRG2_IKE_PROP authentication-method pre-shared-
keys
set groups vpn_config security ike proposal SRG2_IKE_PROP dh-group group14
set groups vpn_config security ike proposal SRG2_IKE_PROP authentication-algorithm sha-256
set groups vpn_config security ike proposal SRG2_IKE_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ike proposal SRG2_IKE_PROP lifetime-seconds 3600
set groups vpn_config security ike policy SRG1_IKE_POL1 proposals SRG1_IKE_PROP
set groups vpn_config security ike policy SRG1_IKE_POL1 pre-shared-key ascii-text "$ABC123"
```

```
set groups vpn_config security ike policy SRG2_IKE_POL500 proposals SRG2_IKE_PROP
set groups vpn_config security ike policy SRG2_IKE_POL500 pre-shared-key ascii-text "$ABC123"
set groups vpn_config security ike gateway SRG1_IKE_GW1 ike-policy SRG1_IKE_POL1
set groups vpn_config security ike gateway SRG1_IKE_GW1 address 10.112.0.1
set groups vpn_config security ike gateway SRG1_IKE_GW1 external-interface lo0
set groups vpn_config security ike gateway SRG1_IKE_GW1 local-address 10.11.0.1
set groups vpn_config security ike gateway SRG1_IKE_GW1 version v2-only
set groups vpn_config security ike gateway SRG2_IKE_GW500 ike-policy SRG2_IKE_POL500
set groups vpn_config security ike gateway SRG2_IKE_GW500 address 10.112.0.5
set groups vpn_config security ike gateway SRG2_IKE_GW500 external-interface lo0
set groups vpn_config security ike gateway SRG2_IKE_GW500 local-address 10.12.0.1
set groups vpn_config security ike gateway SRG2_IKE_GW500 version v2-only
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP protocol esp
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ipsec proposal SRG1_IPSEC_PROP lifetime-seconds 1800
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP protocol esp
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP encryption-algorithm aes-256-cbc
set groups vpn_config security ipsec proposal SRG2_IPSEC_PROP lifetime-seconds 1800
set groups vpn_config security ipsec policy SRG1_IPSEC_POL1 proposals SRG1_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL501 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL500 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL502 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec policy SRG2_IPSEC_POL503 proposals SRG2_IPSEC_PROP
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 bind-interface st0.1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 ike gateway SRG1_IKE_GW1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 ike ipsec-policy SRG1_IPSEC_POL1
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 local-ip
10.1.0.2/32
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 remote-ip
10.7.0.2/32
set groups vpn_config security ipsec vpn SRG1_IPSEC_VPN1 establish-tunnels on-traffic
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 bind-interface st0.500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 ike gateway SRG2_IKE_GW500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 ike ipsec-policy SRG2_IPSEC_POL500
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts500 local-ip
10.8.0.2/32
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts500 remote-ip
10.9.0.2/32
set groups vpn_config security ipsec vpn SRG2_IPSEC_VPN500 establish-tunnels on-traffic
```

```
set groups vpn_config security zones security-zone vpn host-inbound-traffic system-services
ike
set groups vpn_config security zones security-zone vpn host-inbound-traffic protocols all
set groups vpn_config security zones security-zone vpn interfaces st0.1
set groups vpn_config security zones security-zone vpn interfaces st0.500
set groups vpn_config interfaces st0 unit 1 family inet
set groups vpn_config interfaces st0 unit 500 family inet
```

2. Include the apply-groups statement in the configuration to inherit the statements from the vpn_config configuration group,

```
[edit]
user@srx-1# set apply-groups vpn_config
```

Configuration (SRX-03) (VPN Peer Device)

Step-By-Step Procedure

1. Create the IKE proposal.

```
[edit]
user@srx-3# set security ike proposal SRG1_IKE_PROP authentication-method pre-shared-keys
user@srx-3# set security ike proposal SRG1_IKE_PROP dh-group group14
user@srx-3# set security ike proposal SRG1_IKE_PROP authentication-algorithm sha-256
user@srx-3# set security ike proposal SRG1_IKE_PROP encryption-algorithm aes-256-cbc
user@srx-3# set security ike proposal SRG1_IKE_PROP lifetime-seconds 3600
user@srx-3# set security ike proposal SRG2_IKE_PROP authentication-method pre-shared-keys
user@srx-3# set security ike proposal SRG2_IKE_PROP dh-group group14
user@srx-3# set security ike proposal SRG2_IKE_PROP authentication-algorithm sha-256
user@srx-3# set security ike proposal SRG2_IKE_PROP encryption-algorithm aes-256-cbc
user@srx-3# set security ike proposal SRG2_IKE_PROP lifetime-seconds 3600
```

2. Define IKE policies.

```
[edit]
user@srx-3# set security ike policy SRG1_IKE_POL1 proposals SRG1_IKE_PROP
user@srx-3# set security ike policy SRG1_IKE_POL1 pre-shared-key ascii-text "$ABC123"
```

```
user@srx-3# set security ike policy SRG2_IKE_POL500 proposals SRG2_IKE_PROP
user@srx-3# set security ike policy SRG2_IKE_POL500 pre-shared-key ascii-text "$ABC123"
```

3. Create an IKE gateway, define address, specify external interfaces and version.

```
[edit]
user@srx-3# set security ike gateway SRG1_IKE_GW1 ike-policy SRG1_IKE_POL1
user@srx-3# set security ike gateway SRG1_IKE_GW1 address 10.11.0.1
user@srx-3# set security ike gateway SRG1_IKE_GW1 external-interface lo0
user@srx-3# set security ike gateway SRG1_IKE_GW1 local-address 10.112.0.1
user@srx-3# set security ike gateway SRG1_IKE_GW1 version v2-only
user@srx-3# set security ike gateway SRG2_IKE_GW500 ike-policy SRG2_IKE_POL500
user@srx-3# set security ike gateway SRG2_IKE_GW500 address 10.12.0.1
user@srx-3# set security ike gateway SRG2_IKE_GW500 external-interface lo0
user@srx-3# set security ike gateway SRG2_IKE_GW500 local-address 10.112.0.5
user@srx-3# set security ike gateway SRG2_IKE_GW500 version v2-only
```

4. Create IPsec proposals.

```
[edit]
user@srx-3# set security ipsec proposal SRG1_IPSEC_PROP protocol esp
user@srx-3# set security ipsec proposal SRG1_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
user@srx-3# set security ipsec proposal SRG1_IPSEC_PROP encryption-algorithm aes-256-cbc
user@srx-3# set security ipsec proposal SRG1_IPSEC_PROP lifetime-seconds 1800
user@srx-3# set security ipsec proposal SRG2_IPSEC_PROP protocol esp
user@srx-3# set security ipsec proposal SRG2_IPSEC_PROP authentication-algorithm hmac-
sha-256-128
user@srx-3# set security ipsec proposal SRG2_IPSEC_PROP encryption-algorithm aes-256-cbc
user@srx-3# set security ipsec proposal SRG2_IPSEC_PROP lifetime-seconds 1800
```

5. Create IPsec policies.

```
[edit]
user@srx-3# set security ipsec policy SRG1_IPSEC_POL1 proposals SRG1_IPSEC_PROP
user@srx-3# set security ipsec policy SRG2_IPSEC_POL500 proposals SRG2_IPSEC_PROP
```

6. Specify the IPsec proposal references (IKE gateway, IPsec policy, interface to bind, and traffic selectors).

```
[edit]
user@srx-3# set security ipsec vpn SRG1_IPSEC_VPN1 bind-interface st0.1
user@srx-3# set security ipsec vpn SRG1_IPSEC_VPN1 ike gateway SRG1_IKE_GW1
user@srx-3# set security ipsec vpn SRG1_IPSEC_VPN1 ike ipsec-policy SRG1_IPSEC_POL1
user@srx-3# set security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 local-ip 10.7.0.2/32
user@srx-3# set security ipsec vpn SRG1_IPSEC_VPN1 traffic-selector ts1 remote-ip
10.1.0.2/32
user@srx-3# set security ipsec vpn SRG1_IPSEC_VPN1 establish-tunnels immediately
user@srx-3# set security ipsec vpn SRG2_IPSEC_VPN500 bind-interface st0.500
user@srx-3# set security ipsec vpn SRG2_IPSEC_VPN500 ike gateway SRG2_IKE_GW500
user@srx-3# set security ipsec vpn SRG2_IPSEC_VPN500 ike ipsec-policy SRG2_IPSEC_POL500
user@srx-3# set security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts1 local-ip
10.9.0.2/32
user@srx-3# set security ipsec vpn SRG2_IPSEC_VPN500 traffic-selector ts1 remote-ip
10.8.0.2/32
user@srx-3# set security ipsec vpn SRG2_IPSEC_VPN500 establish-tunnels immediately
```

7. Create a security policy.

```
[edit]
user@srx-3# set security policies default-policy permit-all
```

For this example, we've configured a policy to permit all traffic. We strongly recommend you to create security policies as per your network requirements to permit traffic that is allowed by your organizational policy and deny all other traffic. We've used the default policy for the demo purpose only in this example.

8. Configure the interfaces.

```
[edit]
user@srx-3# set interfaces ge-0/0/0 description trust
user@srx-3# set interfaces ge-0/0/0 unit 0 family inet address 10.7.0.1/24
user@srx-3# set interfaces ge-0/0/1 description untrust
user@srx-3# set interfaces ge-0/0/1 unit 0 family inet address 10.6.0.2/24
user@srx-3# set interfaces ge-0/0/2 description trust
user@srx-3# set interfaces ge-0/0/2 unit 0 family inet address 10.9.0.1/24
user@srx-3# set interfaces lo0 description untrust
user@srx-3# set interfaces lo0 unit 0 family inet address 10.112.0.1/32
```

```
user@srx-3# set interfaces lo0 unit 0 family inet address 10.112.0.5/32
user@srx-3# set interfaces st0 unit 1 family inet
user@srx-3# set interfaces st0 unit 500 family inet
```

9. Define security zones and add interfaces.

```
[edit]
user@srx-3# set security zones security-zone untrust host-inbound-traffic system-services
all
user@srx-3# set security zones security-zone untrust host-inbound-traffic protocols all
user@srx-3# set security zones security-zone untrust interfaces st0.1
user@srx-3# set security zones security-zone untrust interfaces lo0.0
user@srx-3# set security zones security-zone untrust interfaces st0.500
user@srx-3# set security zones security-zone untrust interfaces ge-0/0/1.0
user@srx-3# set security zones security-zone untrust interfaces ge-0/0/0.0
user@srx-3# set security zones security-zone trust host-inbound-traffic system-services all
user@srx-3# set security zones security-zone trust host-inbound-traffic protocols all
user@srx-3# set security zones security-zone trust interfaces ge-0/0/2.0
```

10. Configure the static routes.

```
[edit]
user@srx-3# set routing-options autonomous-system 100
user@srx-3# set routing-options static route 10.4.0.0/16 next-hop 10.7.0.2
user@srx-3# set routing-options static route 10.5.0.0/16 next-hop 10.7.0.2
user@srx-3# set routing-options static route 10.11.0.0/24 next-hop 10.7.0.2
user@srx-3# set routing-options static route 10.12.0.0/24 next-hop 10.7.0.2
user@srx-3# set routing-options static route 10.111.0.1/32 next-hop 10.7.0.2
user@srx-3# set routing-options static route 10.111.0.2/32 next-hop 10.7.0.2
```

Results (SRX-01)

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@srx-1# show groups vpn_config
```

```
when {
    peers [ SRX-01 SRX-02 ];
}
security {
    ike {
        proposal SRG1_IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 3600;
        }
        proposal SRG2_IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 3600;
        }
        policy SRG1_IKE_POL1 {
            proposals SRG1_IKE_PROP;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        policy SRG2_IKE_POL500 {
            proposals SRG2_IKE_PROP;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        gateway SRG1_IKE_GW1 {
            ike-policy SRG1_IKE_POL1;
            address 10.112.0.1;
            external-interface lo0;
            local-address 10.11.0.1;
            version v2-only;
        }
        gateway SRG2_IKE_GW500 {
            ike-policy SRG2_IKE_POL500;
            address 10.112.0.5;
            external-interface lo0;
            local-address 10.12.0.1;
            version v2-only;
        }
    }
    ipsec {
```

```
        proposal SRG1_IPSEC_PROP {
            protocol esp;
            authentication-algorithm hmac-sha-256-128;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 1800;
        }
        proposal SRG2_IPSEC_PROP {
            protocol esp;
            authentication-algorithm hmac-sha-256-128;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 1800;
        }
        policy SRG1_IPSEC_POL1 {
            proposals SRG1_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL501 {
            proposals SRG2_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL500 {
            proposals SRG2_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL502 {
            proposals SRG2_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL503 {
            proposals SRG2_IPSEC_PROP;
        }
        vpn SRG1_IPSEC_VPN1 {
            bind-interface st0.1;
            ike {
                gateway SRG1_IKE_GW1;
                ipsec-policy SRG1_IPSEC_POL1;
            }
            traffic-selector ts1 {
                local-ip 10.1.0.2/32;
                remote-ip 10.7.0.2/32;
            }
            establish-tunnels on-traffic;
        }
        vpn SRG2_IPSEC_VPN500 {
            bind-interface st0.500;
            ike {
                gateway SRG2_IKE_GW500;
```

```
                ipsec-policy SRG2_IPSEC_POL500;
            }
            traffic-selector ts500 {
                local-ip 10.8.0.2/32;
                remote-ip 10.9.0.2/32;
            }
            establish-tunnels on-traffic;
        }
    }
    zones {
        security-zone vpn {
            host-inbound-traffic {
                system-services {
                    ike;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                st0.1;
                st0.500;
            }
        }
    }
}
interfaces {
    st0 {
        unit 1 {
            family inet;
            family inet6;
        }
        unit 500 {
            family inet;
            family inet6;
        }
    }
}
```

```
[edit]
user@srx-1# show chassis high-availability
```

```
local-id 1 local-ip 10.22.0.2;
peer-id 2 {
    peer-ip 10.22.0.1;
    interface ge-0/0/2.0;
    vpn-profile ICL_IPSEC_VPN;
    liveness-detection {
        minimum-interval 200;
        multiplier 3;
    }
}
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 10.5.0.2 {
            src-ip 10.5.0.1;
            session-type singlehop;
            interface ge-0/0/3.0;
        }
        interface {
            ge-0/0/3;
            ge-0/0/4;
        }
    }
    active-signal-route {
        10.39.1.1;
    }
    backup-signal-route {
        10.39.1.2;
    }
    prefix-list SRG1_PFX;
    managed-services ipsec;
    preemption;
    activeness-priority 1;
}
```

```
services-redundancy-group 2 {
    peer-id {
        2;
    }
    activeness-probe {
        dest-ip {
            10.111.0.1;
            src-ip 10.12.0.1;
        }
    }
    monitor {
        bfd-liveliness 10.5.0.2 {
            src-ip 10.5.0.1;
            session-type singlehop;
            interface ge-0/0/3.0;
        }
        interface {
            ge-0/0/3;
            ge-0/0/4;
        }
    }
    active-signal-route {
        10.49.1.1;
    }
    backup-signal-route {
        10.49.1.2;
    }
    prefix-list SRG2_PFX;
    managed-services ipsec;
    preemption;
    activeness-priority 200;
}
```

```
[edit]
user@srx-1# show security ike
proposal ICL_IKE_PROP {
    description interchassis_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
```

```
    lifetime-seconds 300;
}
policy ICL_IKE_POL {
    description interchassis_link_encr_tunnel;
    proposals ICL_IKE_PROP;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway ICL_IKE_GW {
    ike-policy ICL_IKE_POL;
    version v2-only;
}
```

```
[edit]
user@srx-1# show security ipsec
proposal ICL_IPSEC_PROP {
    description interchassis_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 300;
}
policy ICL_IPSEC_POL {
    description interchassis_link_encr_tunnel;
    proposals ICL_IPSEC_PROP;
}
vpn ICL_IPSEC_VPN {
    ha-link-encryption;
    ike {
        gateway ICL_IKE_GW;
        ipsec-policy ICL_IPSEC_POL;
    }
}
```

```
[edit]
user@srx-1# show policy-options

prefix-list SRG1_PFX {
    10.11.0.0/24;
}
prefix-list SRG2_PFX {
    10.12.0.0/24;
```

```
}
route-filter-list srg1_rf_list {
    10.11.0.0/24 orlonger;
    10.7.0.0/16 orlonger;
    10.1.0.0/16 orlonger;
}
route-filter-list srg2_rf_list {
    10.12.0.0/24 orlonger;
    10.9.0.0/16 orlonger;
    10.8.0.0/16 orlonger;
}
policy-statement mnha-route-policy {
    term 1 {
        from {
            route-filter-list srg1_rf_list;
            condition active_route_exists_srg1;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 2 {
        from {
            route-filter-list srg1_rf_list;
            condition backup_route_exists_srg1;
        }
        then {
            metric 20;
            accept;
        }
    }
    term 3 {
        from {
            route-filter-list srg2_rf_list;
            condition active_route_exists_srg2;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 4 {
```

```
        from {
            route-filter-list srg2_rf_list;
            condition backup_route_exists_srg2;
        }
        then {
            metric 20;
            accept;
        }
    }
    term default {
        then reject;
    }
}
condition active_route_exists_srg1 {
    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition active_route_exists_srg2 {
    if-route-exists {
        address-family {
            inet {
                10.49.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists_srg1 {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}
```

```
condition backup_route_exists_srg2 {
    if-route-exists {
        address-family {
            inet {
                10.49.1.2/32;
                table inet.0;
            }
        }
    }
}
```

```
[edit]
user@srx-1# show routing-options
autonomous-system 100;
static {
    route 10.7.0.0/16 next-hop 10.5.0.2;
    route 10.112.0.0/24 next-hop 10.5.0.2;
}
```

```
[edit]
user@srx-1# show security zones
security-zone vpn {
    host-inbound-traffic {
        system-services {
            ike;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
```

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

```
[edit]
user@srx-1# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet;
    }
}
ge-0/0/2 {
    description inter_chassis_link;
    unit 0 {
        family inet {
            address 10.22.0.2/24;
        }
    }
}
ge-0/0/3 {
    description untrust;
    unit 0 {
        family inet {
            address 10.3.0.2/24;
        }
    }
}
ge-0/0/4 {
    description trust;
    unit 0 {
        family inet {
            address 10.5.0.1/24;
        }
    }
}
lo0 {
    apply-groups-except global;
    description untrust;
    unit 0 {
        family inet {
            address 10.11.0.1/32;
            address 10.12.0.1/32;
```

```
        }
    }
}
st0 {
    unit 1;
}
```

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

Results (SRX-02)

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@srx-2# show groups vpn_config
when {
    peers [ SRX-01 SRX-02 ];
}
security {
    ike {
        proposal SRG1_IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 3600;
        }
        proposal SRG2_IKE_PROP {
            authentication-method pre-shared-keys;
            dh-group group14;
            authentication-algorithm sha-256;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 3600;
        }
        policy SRG1_IKE_POL1 {
            proposals SRG1_IKE_PROP;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        policy SRG2_IKE_POL500 {
            proposals SRG2_IKE_PROP;
```

```
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        gateway SRG1_IKE_GW1 {
            ike-policy SRG1_IKE_POL1;
            address 10.112.0.1;
            external-interface lo0;
            local-address 10.11.0.1;
            version v2-only;
        }
        gateway SRG2_IKE_GW500 {
            ike-policy SRG2_IKE_POL500;
            address 10.112.0.5;
            external-interface lo0;
            local-address 10.12.0.1;
            version v2-only;
        }
    }
    ipsec {
        proposal SRG1_IPSEC_PROP {
            protocol esp;
            authentication-algorithm hmac-sha-256-128;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 1800;
        }
        proposal SRG2_IPSEC_PROP {
            protocol esp;
            authentication-algorithm hmac-sha-256-128;
            encryption-algorithm aes-256-cbc;
            lifetime-seconds 1800;
        }
        policy SRG1_IPSEC_POL1 {
            proposals SRG1_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL501 {
            proposals SRG2_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL500 {
            proposals SRG2_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL502 {
            proposals SRG2_IPSEC_PROP;
        }
        policy SRG2_IPSEC_POL503 {
```

```
            proposals SRG2_IPSEC_PROP;
        }
        vpn SRG1_IPSEC_VPN1 {
            bind-interface st0.1;
            ike {
                gateway SRG1_IKE_GW1;
                ipsec-policy SRG1_IPSEC_POL1;
            }
            traffic-selector ts1 {
                local-ip 10.1.0.2/32;
                remote-ip 10.7.0.2/32;
            }
            establish-tunnels on-traffic;
        }
        vpn SRG2_IPSEC_VPN500 {
            bind-interface st0.500;
            ike {
                gateway SRG2_IKE_GW500;
                ipsec-policy SRG2_IPSEC_POL500;
            }
            traffic-selector ts500 {
                local-ip 10.8.0.2/32;
                remote-ip 10.9.0.2/32;
            }
            establish-tunnels on-traffic;
        }
    }
    zones {
        security-zone vpn {
            host-inbound-traffic {
                system-services {
                    ike;
                }
                protocols {
                    all;
                }
            }
            interfaces {
                st0.1;
                st0.500;
            }
        }
    }
```

```
}
interfaces {
    st0 {
        unit 1 {
            family inet;
            family inet6;
        }
        unit 500 {
            family inet;
            family inet6;
        }
    }
}
```

```
[edit]
user@srx-2# show chassis high-availability
local-id 2 local-ip 10.22.0.1;
peer-id 1 {
    peer-ip 10.22.0.2;
    interface ge-0/0/2.0;
    vpn-profile ICL_IPSEC_VPN;
    liveness-detection {
        minimum-interval 200;
        multiplier 3;
    }
}
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 10.4.0.2 {
            src-ip 10.4.0.1;
            session-type singlehop;
```

```
            interface ge-0/0/3.0;
        }
        interface {
            ge-0/0/3;
            ge-0/0/4;
        }
    }
    active-signal-route {
        10.39.1.1;
    }
    backup-signal-route {
        10.39.1.2;
    }
    prefix-list SRG1_PFX;
    managed-services ipsec;
    preemption;
    activeness-priority 200;
}
services-redundancy-group 2 {
    peer-id {
        1;
    }
    activeness-probe {
        dest-ip {
            10.111.0.1;
            src-ip 10.12.0.1;
        }
    }
    monitor {
        bfd-liveliness 10.4.0.2 {
            src-ip 10.4.0.1;
            session-type singlehop;
            interface ge-0/0/3.0;
        }
        interface {
            ge-0/0/3;
            ge-0/0/4;
        }
    }
    active-signal-route {
        10.49.1.1;
    }
    backup-signal-route {
```

```
        10.49.1.2;
    }
    prefix-list SRG2_PFX;
    managed-services ipsec;
    preemption;
    activeness-priority 1;
}
```

```
[edit]
user@srx-2# show security ike
proposal ICL_IKE_PROP {
    description interchassisa_link_encr_tunnel;
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 300;
}
policy ICL_IKE_POL {
    description interchassis_link_encr_tunnel;
    proposals ICL_IKE_PROP;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway ICL_IKE_GW {
    ike-policy ICL_IKE_POL;
    version v2-only;
}
```

```
[edit]
user@srx-2# show security ipsec
proposal ICL_IPSEC_PROP {
    description interchassis_link_encr_tunnel;
    protocol esp;
    encryption-algorithm aes-256-gcm;
    lifetime-seconds 300;
}
policy ICL_IPSEC_POL {
    description interchassis_link_encr_tunnel;
    proposals ICL_IPSEC_PROP;
}
```

```
vpn ICL_IPSEC_VPN {
    ha-link-encryption;
    ike {
        gateway ICL_IKE_GW;
        ipsec-policy ICL_IPSEC_POL;
    }
}
```

```
[edit]
user@srx-2# show policy-options
prefix-list SRG1_PFX {
    10.11.0.0/24;
}
prefix-list SRG2_PFX {
    10.12.0.0/24;
}
route-filter-list srg1_rf_list {
    10.11.0.0/24 orlonger;
    10.7.0.0/24 orlonger;
    10.1.0.0/24 orlonger;
}
route-filter-list srg2_rf_list {
    10.12.0.0/24 orlonger;
    10.9.0.0/24 orlonger;
    10.8.0.0/24 orlonger;
}
policy-statement mnha-route-policy {
    term 1 {
        from {
            route-filter-list srg1_rf_list;
            condition active_route_exists_srg1;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 2 {
        from {
            route-filter-list srg1_rf_list;
            condition backup_route_exists_srg1;
```

```
        }
        then {
            metric 20;
            accept;
        }
    }
    term 3 {
        from {
            route-filter-list srg2_rf_list;
            condition active_route_exists_srg2;
        }
        then {
            metric 10;
            accept;
        }
    }
    term 4 {
        from {
            route-filter-list srg2_rf_list;
            condition backup_route_exists_srg2;
        }
        then {
            metric 20;
            accept;
        }
    }
    term default {
        then reject;
    }
}
condition active_route_exists_srg1 {
    if-route-exists {
        address-family {
            inet {
                10.39.1.1/32;
                table inet.0;
            }
        }
    }
}
condition active_route_exists_srg2 {
    if-route-exists {
        address-family {
```

```
            inet {
                10.49.1.1/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists_srg1 {
    if-route-exists {
        address-family {
            inet {
                10.39.1.2/32;
                table inet.0;
            }
        }
    }
}
condition backup_route_exists_srg2 {
    if-route-exists {
        address-family {
            inet {
                10.49.1.2/32;
                table inet.0;
            }
        }
    }
}
```

```
[edit]
user@srx-2# show routing-options
autonomous-system 100;
static {
    route 10.7.0.0/24 next-hop 10.4.0.2;
    route 10.112.0.0/24 next-hop 10.4.0.2;
}
```

```
[edit]
user@srx-2# show security zones
    security-zone untrust {
    host-inbound-traffic {
```

```
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        lo0.0;
        ge-0/0/3.0;
    }
}
security-zone vpn {
    host-inbound-traffic {
        system-services {
            ike;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/4.0;
    }
}
security-zone icl_zone {
    host-inbound-traffic {
        system-services {
```

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

```
[edit]
user@srx-2# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet;
    }
}
ge-0/0/2 {
    description inter_chassis_link;
    unit 0 {
        family inet {
            address 10.22.0.1/24;
        }
    }
}
ge-0/0/3 {
    description untrust;
    unit 0 {
        family inet {
            address 10.2.0.2/24;
        }
    }
}
ge-0/0/4 {
    description trust;
    unit 0 {
```

```
        family inet {
            address 10.4.0.1/24;
        }
    }
}
lo0 {
    apply-groups-except global;
    description untrust;
    unit 0 {
        family inet {
            address 10.11.0.1/32;
            address 10.12.0.1/32;
        }
    }
}
st0 {
    unit 1 {
        family inet;
        }
}
```

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
```

Results (SRX-3) (VPN Peer Device)

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@srx-3# show security ike
proposal SRG1_IKE_PROP {
    authentication-method pre-shared-keys;
```

```
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
proposal SRG2_IKE_PROP {
    authentication-method pre-shared-keys;
    dh-group group14;
    authentication-algorithm sha-256;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 3600;
}
policy SRG1_IKE_POL1 {
    proposals SRG1_IKE_PROP;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
policy SRG2_IKE_POL500 {
    proposals SRG2_IKE_PROP;
    pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
}
gateway SRG1_IKE_GW1 {
    ike-policy SRG1_IKE_POL1;
    address 10.11.0.1;
    external-interface lo0;
    local-address 10.112.0.1;
    version v2-only;
}
gateway SRG2_IKE_GW500 {
    ike-policy SRG2_IKE_POL500;
    address 10.12.0.1;
    external-interface lo0;
    local-address 10.112.0.5;
    version v2-only;
}
```

```
[edit]
user@srx-3# show security ipsec
proposal SRG1_IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-256-cbc;
```

```
    lifetime-seconds 1800;
}
proposal SRG2_IPSEC_PROP {
    protocol esp;
    authentication-algorithm hmac-sha-256-128;
    encryption-algorithm aes-256-cbc;
    lifetime-seconds 1800;
}
policy SRG1_IPSEC_POL1 {
    proposals SRG1_IPSEC_PROP;
}
policy SRG2_IPSEC_POL500 {
    proposals SRG2_IPSEC_PROP;
}
vpn SRG1_IPSEC_VPN1 {
    bind-interface st0.1;
    ike {
        gateway SRG1_IKE_GW1;
        ipsec-policy SRG1_IPSEC_POL1;
    }
    traffic-selector ts1 {
        local-ip 10.7.0.2/32;
        remote-ip 10.1.0.2/32;
    }
    establish-tunnels immediately;
}
vpn SRG2_IPSEC_VPN500 {
    bind-interface st0.500;
    ike {
        gateway SRG2_IKE_GW500;
        ipsec-policy SRG2_IPSEC_POL500;
    }
    traffic-selector ts1 {
        local-ip 10.9.0.2/32;
        remote-ip 10.8.0.2/32;
    }
    establish-tunnels immediately;
}
```

```
[edit]
user@srx-3# show routing-options
```

```
autonomous-system 100;
static {
    route 10.4.0.0/24 next-hop 10.7.0.2;
    route 10.5.0.0/24 next-hop 10.7.0.2;
    route 10.11.0.0/24 next-hop 10.7.0.2;
    route 10.12.0.0/24 next-hop 10.7.0.2;
    route 10.111.0.1/32 next-hop 10.7.0.2;
    route 10.111.0.2/32 next-hop 10.7.0.2;
}
```

```
[edit]
user@srx-3# show security zones
    security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
        }
        protocols {
            all;
        }
    }
    interfaces {
        st0.1;
        lo0.0;
        st0.500;
        ge-0/0/1.0;
        ge-0/0/0.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            ike;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
```

```
}
```

```
[edit]
user@srx-3# show interfaces
ge-0/0/0 {
    description trust;
    unit 0 {
        family inet {
            address 10.7.0.1/24;
        }
    }
}
ge-0/0/1 {
    description untrust;
    unit 0 {
        family inet {
            address 10.6.0.2/24;
        }
    }
}
ge-0/0/2 {
    description trust;
    unit 0 {
        family inet {
            address 10.9.0.1/24;
        }
    }
}
lo0 {
    description untrust;
    unit 0 {
        family inet {
            address 10.112.0.1/32;
            address 10.112.0.5/32;
        }
    }
}
st0 {
    unit 1 {
```

```
        family inet;
    }
    unit 500 {
        family inet;
    }
}
```

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

Verification

IN THIS SECTION

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@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: 10.22.0.2
HA Peer Information:

    Peer Id: 2        IP address: 10.22.0.1     Interface: ge-0/0/2.0
    Routing Instance: default
    Encrypted: YES    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: ROUTING
        Status: BACKUP
        Activeness Priority: 1
        Preemption: ENABLED
        Process Packet In Backup State: NO
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
          Peer Id: 2
          Status : ACTIVE
          Health Status: HEALTHY
          Failover Readiness: N/A
```

```
Services Redundancy Group: 2
        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: NOT READY
```

On SRX-2

```
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: 10.22.0.1
HA Peer Information:

    Peer Id: 1        IP address: 10.22.0.2     Interface: ge-0/0/2.0
    Routing Instance: default
    Encrypted: YES    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: 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: 1
          Status : BACKUP
          Health Status: HEALTHY
          Failover Readiness: NOT READY


Services Redundancy Group: 2
        Deployment Type: ROUTING
        Status: BACKUP
        Activeness Priority: 1
        Preemption: ENABLED
        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` and `Services Redundancy Group: 2` indicate the status of the SRG1 and SRG2 (active or backup) on that node.

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-01

```
user@srx-01> show chassis high-availability information detail
Node level Information:
        Node Status: ONLINE
        Local-id: 1
        Local-IP: 10.22.0.2
HA Peer Information:

   Peer-ID: 2        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              6
        SRG Status Ack            4              4
        Attribute Msg             1              1
        Attribute Ack             1              1

HA Peer Conn events:
        Jan 31 00:55:19.249 : HA Peer 180.100.1.1 BFD conn came up

Cold Synchronization:
    Status:
        Cold synchronization completed for: N/A
```

```
        Cold synchronization failed for: N/A
        Cold synchronization not known for: N/A
        Current Monitoring Weight: 0

    Progress:
        CS Prereq               1 of 1 SPUs completed
           1. if_state sync          1 SPUs completed
           2. ha peer conn           1 SPUs completed
           3. policy data sync       1 SPUs completed
           4. cp ready               1 SPUs completed
           5. VPN data sync          1 SPUs completed
           6. IPID data sync         1 SPUs completed
           7. All SPU ready          1 SPUs completed
           8. AppID ready            1 SPUs completed
           9. Tunnel Sess ready      1 SPUs completed
        CS RTO sync             1 of 1 SPUs completed
        CS Postreq              1 of 1 SPUs completed

    Statistics:
        Number of cold synchronization completed: 0
        Number of cold synchronization failed: 0

    Events:
        Jan 31 00:55:24.616 : Cold sync for PFE  is Post-req check in process
        Jan 31 00:55:25.615 : Cold sync for PFE  is Completed

SPU monitoring:
    Status: Enabled
        Current monitoring weight: 0


    Statistics:
        SPU up count: 1
        NPC up count: 0
        SPU down count: 0
        NPC down count: 0
        Chassis info processing error count: 0

Loopback Information:

    PIC Name        Loopback        Nexthop     Mbuf
    ------------------------------------------------
                    Success         Success     Success
```

```
Hardware monitoring:
    Status:
        Activation status: Enabled
        Ctrl Plane Hardware errors: 0
        Data Plane Hardware errors: 0

SRGS Information:

Services Redundancy Group: 1
        Deployment Type: ROUTING
        Status: BACKUP
        Activeness Priority: 1
        Hold Timer: 1
        Services: [ IPSEC ]
        Process Packet In Backup State: NO
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Peer Information:
        Failure Events: NONE
          Peer Id: 2
          Last Advertised HA Status: ACTIVE
          Last Advertised Health Status: HEALTHY
          Failover Readiness: N/A

        Signal Route Info:
          Active Signal Route:
          IP: 10.39.1.1
          Routing Instance: default
          Status: NOT INSTALLED

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

```
SRG State Change Events:
        Jan 31 00:52:14.347 : SRG[1] state UNKNOWN -> HOLD, Reason: State machine start
        Jan 31 00:56:33.046 : SRG[1] state HOLD -> BACKUP, Reason: Peer state Active received

        BFD Monitoring:
          Status: UNKNOWN

          SRC-IP: 10.5.0.1     DST-IP: 10.5.0.2
          Routing Instance: default
          Type: SINGLE-HOP
              IFL Name: ge-0/0/3.0
          State: INSTALLED



        Interface Monitoring:
          Status: UP

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

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


Probe status events:
        Jan 31 00:54:12.695 : SRG[1] HA probe dst 10.111.0.1 became unreachable, Reason: UNKNOWN

SRGS Information:

Services Redundancy Group: 2
        Deployment Type: ROUTING
        Status: ACTIVE
        Activeness Priority: 200
        Hold Timer: 1
        Services: [ IPSEC ]
        Process Packet In Backup State: NO
        Control Plane State: READY
        System Integrity Check: N/A
        Peer Information:
        Failure Events: NONE
          Peer Id: 2
          Last Advertised HA Status: BACKUP
          Last Advertised Health Status: HEALTHY
          Failover Readiness: NOT READY
```

```
        Signal Route Info:
          Active Signal Route:
          IP: 10.49.1.1
          Routing Instance: default
          Status: INSTALLED

          Backup Signal Route:
          IP: 10.49.1.2
          Routing Instance: default
          Status: NOT INSTALLED

        Split-brain Prevention Probe Info:
          DST-IP: 10.111.0.1
          SRC-IP: 10.12.0.1
          Routing Instance: default
          Status: NOT RUNNING
          Result: N/A           Reason: N/A


SRG State Change Events:
        Jan 31 00:52:14.439 : SRG[2] state UNKNOWN -> HOLD, Reason: State machine start
        Jan 31 00:55:24.263 : SRG[2] state HOLD -> ACTIVE, Reason: Local Priority Higher

        BFD Monitoring:
          Status: UNKNOWN

          SRC-IP: 10.5.0.1     DST-IP: 10.5.0.2
          Routing Instance: default
          Type: SINGLE-HOP
              IFL Name: ge-0/0/3.0
          State: INSTALLED


        Interface Monitoring:
          Status: UP

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

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

```
Probe status events:
        Jan 31 00:54:13.698 : SRG[2] HA probe dst 10.111.0.1 became unreachable, Reason: UNKNOWN
```

Meaning

Verify these details from the command output:

- The field `Services: [ IPSEC ]` indicates the associated IPSec VPN for each SRG.
- The fields `BFD Monitoring`, `Interface Monitoring`, `Split-brain Prevention Probe Info` display monitoring details.
- The fields `Cold Synchronization`, `SRG State Change Events` provide details on current status and recent changes.
- The field `Services Redundancy Group: 1` and `Services Redundancy Group: 2` indicate the status of the SRG1 and SRG2 (active or backup) on that node.

In the command output, the IP addresses such as IP 180.100.1.2 are generated internally by Junos OS and these addresses do not interfere with routing tables.

Check Multinode High Availability Peer Node Status

Purpose

View and verify the peer node details.

Action

From operational mode, run the following command on SRX-01 and SRX-02:

SRX-01

```
user@srx-01> show chassis high-availability peer-info
HA Peer Information:

   Peer-ID: 2        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              6
        SRG Status Ack            4              4
        Attribute Msg             1              1
        Attribute Ack             1              1
```

SRX-02

```
user@srx-02> show chassis high-availability peer-info
HA Peer Information:

   Peer-ID: 1        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            6              4
        SRG Status Ack            4              4
        Attribute Msg             2              1
        Attribute Ack             1              1
```

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 on both security devices:

SRG1 on SRX-02

```
user@srx-02> 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: 1
          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

        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: UNKNOWN

          SRC-IP: 10.4.0.1     DST-IP: 10.4.0.2
          Routing Instance: default
          Type: SINGLE-HOP
              IFL Name: ge-0/0/3.0
          State: INSTALLED


        Interface Monitoring:
          Status: UP

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

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


IP SRGID Table:
        SRGID    IP Prefix                                  Routing Table
        1        10.11.0.0/24                               default
```

SRG2 on SRX-02

```
user@srx-02> show chassis high-availability services-redundancy-group 2
SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
    IF  Interface monitoring
    CP  Control Plane monitoring
```

```
Services Redundancy Group: 2
        Deployment Type: ROUTING
        Status: BACKUP
        Activeness Priority: 1
        Preemption: ENABLED
        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

        Signal Route Info:
          Active Signal Route:
          IP: 10.49.1.1
          Routing Instance: default
          Status: NOT INSTALLED

          Backup Signal Route:
          IP: 10.49.1.2
          Routing Instance: default
          Status: INSTALLED

        Split-brain Prevention Probe Info:
          DST-IP: 10.111.0.1
          SRC-IP: 10.12.0.1
          Routing Instance: default
          Status: NOT RUNNING
          Result: N/A           Reason: N/A


        BFD Monitoring:
          Status: UNKNOWN

          SRC-IP: 10.4.0.1     DST-IP: 10.4.0.2
          Routing Instance: default
          Type: SINGLE-HOP
              IFL Name: ge-0/0/3.0
          State: INSTALLED
```

```
        Interface Monitoring:
          Status: UP

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

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


IP SRGID Table:
        SRGID    IP Prefix                                  Routing Table
        2        10.12.0.0/24                               default
```

SRG1 on SRX-01

```
user@srx-01> 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: BACKUP
        Activeness Priority: 1
        Preemption: ENABLED
        Process Packet In Backup State: NO
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
          Peer Id: 2
          Status : ACTIVE
          Health Status: HEALTHY
          Failover Readiness: N/A

        Signal Route Info:
          Active Signal Route:
          IP: 10.39.1.1
          Routing Instance: default
          Status: NOT INSTALLED
```

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

          SRC-IP: 10.5.0.1     DST-IP: 10.5.0.2
          Routing Instance: default
          Type: SINGLE-HOP
              IFL Name: ge-0/0/3.0
          State: INSTALLED


        Interface Monitoring:
          Status: UP

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

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


IP SRGID Table:
        SRGID    IP Prefix                                  Routing Table
        1        10.11.0.0/24                               default
```

SRG2 on SRX-01

```
user@srx-01> show chassis high-availability services-redundancy-group 2
SRG failure event codes:
    BF  BFD monitoring
    IP  IP monitoring
```

```
    IF  Interface monitoring
    CP  Control Plane monitoring

Services Redundancy Group: 2
        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: NOT READY

        Signal Route Info:
          Active Signal Route:
          IP: 10.49.1.1
          Routing Instance: default
          Status: INSTALLED

          Backup Signal Route:
          IP: 10.49.1.2
          Routing Instance: default
          Status: NOT INSTALLED

        Split-brain Prevention Probe Info:
          DST-IP: 10.111.0.1
          SRC-IP: 10.12.0.1
          Routing Instance: default
          Status: NOT RUNNING
          Result: N/A           Reason: N/A


        BFD Monitoring:
          Status: UNKNOWN

          SRC-IP: 10.5.0.1     DST-IP: 10.5.0.2
          Routing Instance: default
          Type: SINGLE-HOP
```

```
              IFL Name: ge-0/0/3.0
            State: INSTALLED


         Interface Monitoring:
           Status: UP

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

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


IP SRGID Table:
         SRGID    IP Prefix                                  Routing Table
         2        10.12.0.0/24                               default
```

Meaning

Verify these details from the command output:

- Peer node details such as deployment type, status, active and back up signal routes.
- Split-brain prevention probe, IP monitoring and BFD monitoring status.
- Associated IP prefix table.

Verify Interchassis Link (ICL) Encryption Status

Purpose

Verify the interchassis link (ICL) status.

Action

Run the following command on SRX-01:

```
user@srx-01> show security ipsec security-associations ha-link-encryption
Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm                SPI       Life:sec/kb  Mon   lsys   Port      Gateway
```

```
  <495002 ESP:aes-gcm-256/aes256-gcm 0x0008d9c7 236/ unlim  -     root    500      10.22.0.1
  >495002 ESP:aes-gcm-256/aes256-gcm 0x0001a573 236/ unlim  -     root    500      10.22.0.1
```

```
user@srx-01> show security ike security-associations ha-link-encryption
Index    State  Initiator cookie   Responder cookie  Mode       Remote Address
16776938 UP     9f8fe46ce3be92f8  44e6b3fd74cc9294  IKEv2      10.22.0.1
```

```
user@srx-01> show security ipsec security-associations ha-link-encryption detail

ID: 495002 Virtual-system: root, VPN Name: ICL_IPSEC_VPN
  Local Gateway: 10.22.0.2, Remote Gateway: 10.22.0.1
  Traffic Selector Name: __ICL_IPSEC_VPN__multi_node__
  Local Identity: ipv4(180.100.1.2-180.100.1.2)
  Remote Identity: ipv4(180.100.1.1-180.100.1.1)
  TS Type: traffic-selector
  Version: IKEv2
  Quantum Secured: No
  PFS group: N/A
  SRG ID: 0
  DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.16000, Policy-name: ICL_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: 0x0008d9c7, AUX-SPI: 0
                              , VPN Monitoring: -
    Hard lifetime: Expires in 200 seconds
    Lifesize Remaining:  Unlimited
    Soft lifetime: Expires in 115 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: 16776938
  Direction: outbound, SPI: 0x0001a573, AUX-SPI: 0
```

```
                          , VPN Monitoring: -
  Hard lifetime: Expires in 200 seconds
  Lifesize Remaining:  Unlimited
  Soft lifetime: Expires in 115 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: 16776938
```

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

CAUTION: The IP range (180.100.1.x) shown in the command output serves as the ICL IPsec traffic selector. The system dynamically assigns this IP range, and it is essential not to alter or modify it. Additionally, BFD (Bidirectional Forwarding Detection) will be automatically enabled for the broader 180.x.x.x IP range.

Verify Link Encryption Tunnel Statistics

Purpose

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

Action

Run the following command on SRX-01:

```
user@srx-01> show security ipsec statistics ha-link-encryption
ESP Statistics:
  Encrypted bytes:        106294156
  Decrypted bytes:         51961287
  Encrypted packets:         979531
  Decrypted packets:         989651
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

Run the following commands on SRX-01 and SRX-02 devices:

SRX-1

```
user@srx-01> 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
```

SRX-2

```
user@srx-02> 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
```

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.

Confirm VPN Status

Purpose

Confirm VPN status by checking the status of any IKE security associations at SRG level.

Action

Run the following commands on SRX-1, SRX-2, and SRX-3 (VPN peer device):

SRX-01

```
user@srx-01> show security ike security-associations srg-id 1
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
16777319 UP    19e7cd4e503eeb2e  0800a7ceaafda740  IKEv2          10.112.0.1
```

```
user@srx-01> show security ike security-associations srg-id 2
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
33554536 UP    9944aaf1ab914b42  15cef0da496bdd92  IKEv2          10.112.0.5
```

SRX-02

```
user@srx-02> show security ike security-associations srg-id 1
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
16777319 UP    19e7cd4e503eeb2e  0800a7ceaafda740  IKEv2          10.112.0.1
```

```
user@srx-02> show security ike security-associations srg-id 2
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
33554534 UP    366d174d847f8c71  2f654c6f1c463d80  IKEv2          10.112.0.5
```

SRX-3 (VPN Peer Device)

```
user@srx-03> show security ike security-associations
Index   State  Initiator cookie  Responder cookie  Mode           Remote Address
5929032 UP     366d174d847f8c71  2f654c6f1c463d80  IKEv2          10.12.0.1
5929033 UP     19e7cd4e503eeb2e  0800a7ceaafda740  IKEv2          10.11.0.1
```

Meaning

The output indicates that:

- IP addresses of the remote peers.
- The state showing UP for both remote peers indicates the successful association of Phase 1 establishment.
- The remote peer IP address, IKE policy, and external interfaces are all correct.

Display IPsec Security Association Details

Purpose

Display the individual IPsec SA details identified by SRG IDs.

Action

Run the following command on the SRX Series Firewalls:

SRX-1

```
user@srx-01> show security ipsec security-associations srg-id 1
Total active tunnels: 1     Total IPsec sas: 1
  ID        Algorithm              SPI       Life:sec/kb  Mon lsys  Port  Gateway
  <17277223 ESP:aes-cbc-256/sha256 0xc50520d4 1210/ unlim  -   root  500   10.112.0.1
  >17277223 ESP:aes-cbc-256/sha256 0x6d1e9c89 1210/ unlim  -   root  500   10.112.0.1
```

```
user@srx-01> show security ipsec security-associations srg-id 2
Total active tunnels: 1     Total IPsec sas: 1
  ID      Algorithm              SPI       Life:sec/kb  Mon lsys  Port  Gateway
  <34054437 ESP:aes-cbc-256/sha256 0x9feb290c 1382/ unlim  -   root  500   10.112.0.5
  >34054437 ESP:aes-cbc-256/sha256 0xf41d091c 1382/ unlim  -   root  500   10.112.0.5
```

SRX-02

```
user@srx-02> show security ipsec security-associations srg-id 1
Total active tunnels: 1     Total IPsec sas: 1
  ID        Algorithm              SPI       Life:sec/kb  Mon lsys  Port  Gateway
  <17277223 ESP:aes-cbc-256/sha256 0xc50520d4 1286/ unlim  -   root  500   10.112.0.1
  >17277223 ESP:aes-cbc-256/sha256 0x6d1e9c89 1286/ unlim  -   root  500   10.112.0.1
```

```
user@srx-02> show security ipsec security-associations srg-id 2
  Total active tunnels: 1     Total IPsec sas: 1
  ID        Algorithm              SPI       Life:sec/kb   Mon  lsys  Port  Gateway
  <34054437 ESP:aes-cbc-256/sha256 0x9feb290c 1461/ unlim   -    root  500   10.112.0.5
  >34054437 ESP:aes-cbc-256/sha256 0xf41d091c 1461/ unlim   -    root  500   10.112.0.5
```

SRX-03

```
user@srx-03> show security ipsec security-associations

  Total active tunnels: 2     Total Ipsec sas: 2
  ID         Algorithm             SPI      Life:sec/kb  Mon   lsys Port  Gateway
  <67108865 ESP:aes-cbc-256/sha256 6d1e9c89 1392/ unlim  -     root 500 10.11.0.1
  >67108865 ESP:aes-cbc-256/sha256 c50520d4 1392/ unlim  -     root 500 10.11.0.1
```

```
  <67108866 ESP:aes-cbc-256/sha256 f41d091c 1570/ unlim  -     root 500 10.12.0.1
  >67108866 ESP:aes-cbc-256/sha256 9feb290c 1570/ unlim  -     root 500 10.12.0.1
```

Meaning

The output displays the state of the VPN.

Display Active Peers Per SRG

Purpose

Display the list of connected active peers with peer addresses and ports they are using.

Action

Run the following commands on the SRX Series Firewalls:

SRX-01

```
user@srx-01> show security ike active-peer srg-id 1
Remote Address       Port     Peer IKE-ID       AAA username        Assigned IP
10.112.0.1           500      10.112.0.1        not available       0.0.0.0
```

```
user@srx-01> show security ike active-peer srg-id 2
Remote Address      Port     Peer IKE-ID       AAA username      Assigned IP
10.112.0.5          500      10.112.0.5        not available     0.0.0.0
```

SRX-02

```
user@srx-02> show security ike active-peer srg-id 1
Remote Address     Port     Peer IKE-ID       AAA username         Assigned IP
10.112.0.1         500      10.112.0.1        not available        0.0.0.0
```

```
user@srx-02>  show security ike active-peer srg-id 2
Remote Address   Port     Peer IKE-ID        AAA username         Assigned IP
10.112.0.5       500      10.112.0.5         not available        0.0.0.0
```

Meaning

The output displays the list of connected devices with details about the peer addresses and ports used.

Display IP Prefix to SRG Mapping

Purpose

Display IP prefix to SRG mapping information.

Action

Run the following command on SRX-01 device.

```
user@srx-01> show chassis high-availability prefix-srgid-table
        IP SRGID Table:
        SRGID    IP Prefix              Routing Table
        1        10.11.0.0/24           default

        2        10.12.0.0/24           default
```

Meaning

Output shows IP address prefixes mapped to SRGs in the setup.

Display BGP Session Information.

Purpose

Display summary information about BGP and its neighbors to determine if routes are received from peers.

Action

Run the following commands on the SRX Series Firewalls:

SRX-1 Device

```
user@srx-01> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                       0          0          0          0          0          0
Peer                 AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.3.0.1             100      37         40       0       0       15:43 Establ
  inet.0: 0/0/0/0
10.5.0.2             100      37         40       0       0       15:42 Establ
  inet.0: 0/0/0/0
```

SRX-2 Device

```
user@srx-02> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      0          0          0          0          0          0
Peer                    AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
St                                                                    ate|#Active/
Received/Accepted/Damped...
10.2.0.1               100        842        846       0       0     6:18:40
Es                                                                    tabl
```

```
  inet.0: 0/0/0/0
10.4.0.2                100        842        846       0       0     6:18:42
Es                                                                          tabl
  inet.0: 0/0/0/0
```

Meaning

The output shows that the BGP session is established and the peers are exchanging update messages.

SEE ALSO

- Two-Node Multinode High Availability | 573
- IPsec VPN Support in Multinode High Availability | 635
- Prepare Your Environment for Multinode High Availability Deployment | 620

Example: Configure Multinode High Availability with Junos OS Configuration Groups

SUMMARY

Read this topic to understand how to configure Multinode High Availability using Junos OS configuration groups.

IN THIS SECTION

In Multinode High Availability, two Junos OS security devices act as independent devices. These devices have unique hostname and the IP address on fxp0 interface. You can configure Multinode High Availability using Junos groups statements. To ensure identical security configurations and posture between two devices, you can configure groups for Multinode High Availability setup. Multinode High Availability nodes synchronize configurations exclusively based on this group method.

When you need to configure statements that are common on both nodes, you can use one of the following approaches:

- You can configure common configuration (like security) on one device and manually copy and paste on the other device. Or you can use some external tool (example: scripting) to copy the same configuration snippets to both devices as applicable.
- Use common Junos group configuration synchronized between both nodes (but edited on one device). This approach includes:
  - Configure the feature/function as part of groups. These configuration groups enable you to create smaller, more logically constructed configuration files
  - Synchronize the configuration using the `edit system commit peers-synchronize` option.
  - Mention the device name in the group using the `when peers <device-name>` statement.

  When you enable configuration synchronization (by using the peers-synchronize option) on both the devices in a Multinode High Availability, configuration settings you configure on one peer under [groups] will automatically sync to the other peer upon the commit action.

  For more details on configuration groups, see Use Configuration Groups to Quickly Configure Devices .

  Note that on Security Director or Security Director Cloud, the system manages reusable configuration snippets, similar to Junos Groups, through the use of policy templates and shared objects.

In this example, we'll configure Multinode High Availability using Junos groups statements.

TIP:

Table 51: Time Estimates

Reading Time	30 minutes
Configuration Time	60 minutes

Example Prerequisites

Table 52 on page 914 lists the hardware and software components that support the configuration.

Table 52: Requirements

Hardware requirements	Supported firewalls and virtual firewalls.
Software requirements	We've tested this example using Junos OS Release 24.4R1. See Feature Explorer for details about support for Junos OS Groups and Multinode High Availability. Junos IKE package is required on your firewall for Multinode High Availability configuration. This package is available as a default package or as an optional package on the device. See Support for Junos IKE Package for details. If the package is not installed by default on your firewall, use the following command to install it: `user@host> request system software add optional://junos-ike.tgz` You require this step for ICL encryption.
Licensing requirements	No separate license is required to configure Multinode High Availability. Licenses needed for features such as IDP, Application Identification, Juniper ATP Cloud are unique to each firewall and need to be set on each device. Licenses are unique to each device and cannot be shared between the nodes in a Multinode High Availability setup. Therefore, you must use identical licenses on both the nodes.

Before You Begin

Know more	Using groups configuration in Multinode High Availability simplifies the setup by allowing you to create reusable configuration blocks. These groups can be applied across different parts of the configuration, ensuring consistency and reducing the need for repetitive entries. This approach makes the configuration files more concise and logically structured. Group configuration helps in easy maintenance of configuration files on Juniper Networks devices.
Learn more	Multinode High Availability, Use Configuration Groups to Quickly Configure Devices

Functional Overview

Table 53 on page 915 provides a quick summary of the configuration components deployed in this example.

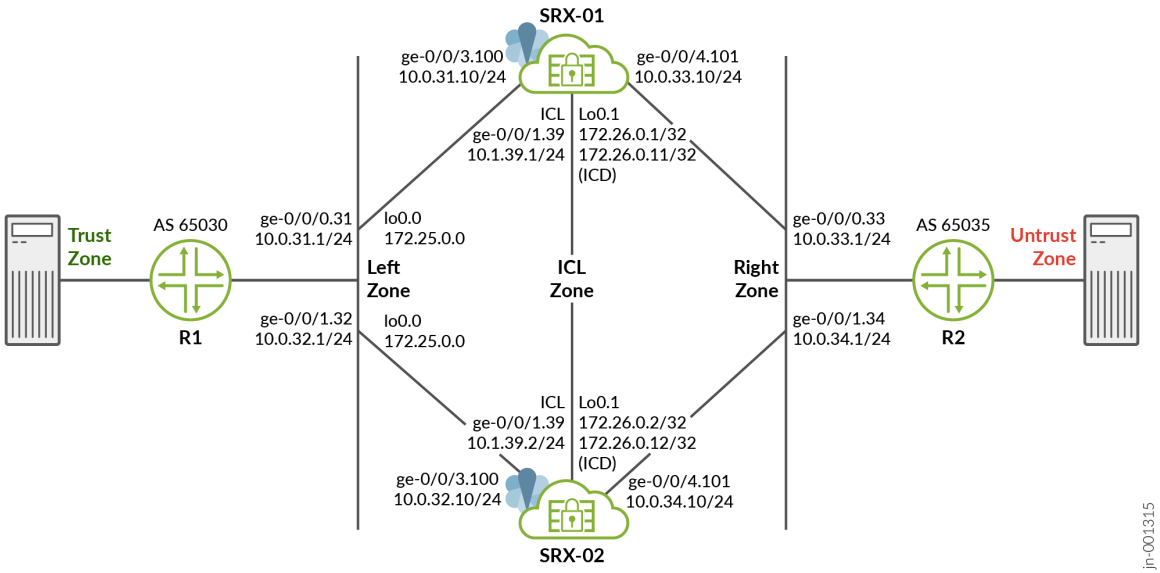
Table 53: Configuration Components

Technologies used	• High availability • Junos OS Configuration Groups • IPsec VPN • Routing policy • Routing options
Primary verification tasks	1. Verify the high availability on both the nodes in the setup. 2. Verify the Multinode High Availability data plane statistics.

Topology Illustration

Figure 69 on page 916 shows the topology used in this configuration example.

Figure 69: Multinode High Availability in Layer 3 Network



As shown in the topology, two SRX devices in MNHA are connected to adjacent routers (vSRX instances acting as routers). An encrypted logical interchassis link (ICL) connects the nodes. The nodes communicate with each other using a routable IP address (floating IP address) over the network. 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.

Loopback interfaces are used to host the IP addresses on firewalls and routers and the IP address on a loopback unit on each respective node is used for communication. In a typical high availability deployment, you have multiple routers and switches on the northbound and southbound sides of the network.

In this example, you'll create multiple configuration groups on devices and synchronize the configuration.

Topology Overview

Table 54 on page 917 shows the details on interfaces configuration used in this example.

Table 54: Interfaces and IP Address Configuration on Security Devices

Device	Interface	IP Address	Zone	Configured For
SRX-01	lo0.1	172.26.0.11/32	ICL Zone	Local forwarding address used to forward data packet over ICD link.
	lo0.1	172.26.0.1/32	ICL Zone	ICL
	lo0.0	172.25.0.0/32	Left Zone	Floating IP address
	ge-0/0/1.39	10.1.39.1/24	ICL Zone	ICL to node 0 connection
	• ge-0/0/3.100 • ge-0/0/4.101	• 10.0.31.10/24 • 10.0.33.10/24	• Left Zone • Right Zone	Connects to upstream and downstream routers.
SRX-02	lo0.1	172.26.0.12/32	ICL Zone	Local forwarding address used to forward data packet over ICD link.
	lo0.1	172.26.0.2/32	ICL Zone	ICL
	lo0.0	172.25.0.0/32	Left Zone	Floating IP address
	ge-0/0/1.39	10.1.39.2/24	ICL Zone	ICL to node 0 connection
	• ge-0/0/3.100 • ge-0/0/4.101	• 10.0.32.10/24 • 10.0.34.10/24	• Left Zone • Right Zone	Connects to upstream and downstream routers.

Table 55: Interfaces and IP Address Configuration on Routing Devices

Device	Interface	IP Address	Configured For
Router 1 (R1)	ge-0/0/0.31	10.0.31.1/24	Connects to SRX-01
	ge-0/0/1.32	10.0.32.1/24	Connects to SRX-02
Router 2 (R2)	ge-0/0/0.33	10.0.33.1/24	Connects to SRX-01

Table 55: Interfaces and IP Address Configuration on Routing Devices *(Continued)*

Device	Interface	IP Address	Configured For
	ge-0/0/1.34	10.0.34.1/24	Connects to SRX-02

Configure Multinode High Availability Using Junos Group Statements

1. Configure common features/functions for Multinode High Availability Using Junos Group statements on active node (SRX-01).

   Note that we have included the term ‘sync’ in the group names as a naming convention to clearly indicate to admins and users that these groups are intended for synchronization.

   a. Configure groups for Multinode High Availability configuration. Within these groups, you can define security zones, security policies, IPsec tunnel definitions, and more.

```
[edit groups mnha-sync]
 user@vsrx-mnha-n0# set when peers vsrx-mnha-n0
 user@vsrx-mnha-n0# set when peers vsrx-mnha-n1
 user@vsrx-mnha-n0# set security ike proposal ike-prop authentication-method pre-shared-
keys
 user@vsrx-mnha-n0# set security ike proposal ike-prop dh-group group20
 user@vsrx-mnha-n0# set security ike proposal ike-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n0# set security ike proposal ike-prop lifetime-seconds 28800
 user@vsrx-mnha-n0# set security ike policy ike-policy proposals ike-prop
 user@vsrx-mnha-n0# set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n0# set security ike policy icl proposals ike-prop
 user@vsrx-mnha-n0# set security ike gateway r1 ike-policy ike-policy
 user@vsrx-mnha-n0# set security ike gateway r1 address 10.0.30.1
 user@vsrx-mnha-n0# set security ike gateway r1 dead-peer-detection probe-idle-tunnel
 user@vsrx-mnha-n0# set security ike gateway r1 dead-peer-detection interval 5
 user@vsrx-mnha-n0# set security ike gateway r1 dead-peer-detection threshold 5
 user@vsrx-mnha-n0# set security ike gateway r1 external-interface lo0.0
 user@vsrx-mnha-n0# set security ike gateway r1 version v2-only
 user@vsrx-mnha-n0# set security ike gateway icl ike-policy icl
 user@vsrx-mnha-n0# set security ike gateway icl version v2-only
 user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop lifetime-seconds 3600
 user@vsrx-mnha-n0# set security ipsec policy ipsec-policy perfect-forward-secrecy keys
```

```
group20
 user@vsrx-mnha-n0# set security ipsec policy ipsec-policy proposals ipsec-prop
 user@vsrx-mnha-n0# set security ipsec vpn r1 bind-interface st0.0
 user@vsrx-mnha-n0# set security ipsec vpn r1 ike gateway r1
 user@vsrx-mnha-n0# set security ipsec vpn r1 ike ipsec-policy ipsec-policy
 user@vsrx-mnha-n0# set security ipsec vpn r1 traffic-selector ts1 local-ip 10.0.35.11/32
 user@vsrx-mnha-n0# set security ipsec vpn r1 traffic-selector ts1 remote-ip 10.0.30.11/32
 user@vsrx-mnha-n0# set security ipsec vpn r1 establish-tunnels immediately
 user@vsrx-mnha-n0# set security ipsec vpn icl ha-link-encryption
 user@vsrx-mnha-n0# set security ipsec vpn icl ike gateway icl
 user@vsrx-mnha-n0# set security ipsec vpn icl ike ipsec-policy ipsec-policy
 user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit match
source-address any
 user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit match
destination-address any
 user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit match
application any
 user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit then
permit
 user@vsrx-mnha-n0# set security policies global policy internal match source-address any
 user@vsrx-mnha-n0# set security policies global policy internal match destination-address
any
 user@vsrx-mnha-n0# set security policies global policy internal match application any
 user@vsrx-mnha-n0# set security policies global policy internal match from-zone right
 user@vsrx-mnha-n0# set security policies global policy internal match from-zone vpn
 user@vsrx-mnha-n0# set security policies global policy internal match from-zone left
 user@vsrx-mnha-n0# set security policies global policy internal match to-zone left
 user@vsrx-mnha-n0# set security policies global policy internal match to-zone right
 user@vsrx-mnha-n0# set security policies global policy internal match to-zone vpn
 user@vsrx-mnha-n0# set security policies global policy internal then permit
 user@vsrx-mnha-n0# set security policies global policy internal then log session-close
 user@vsrx-mnha-n0# set security policies global policy untrust match source-address any
 user@vsrx-mnha-n0# set security policies global policy untrust match destination-address
any
 user@vsrx-mnha-n0# set security policies global policy untrust match application any
 user@vsrx-mnha-n0# set security policies global policy untrust match from-zone left
 user@vsrx-mnha-n0# set security policies global policy untrust match from-zone right
 user@vsrx-mnha-n0# set security policies global policy untrust match to-zone untrust
 user@vsrx-mnha-n0# set security policies global policy untrust then permit
 user@vsrx-mnha-n0# set security zones security-zone vpn interfaces st0.0
 user@vsrx-mnha-n0# set security zones security-zone left interfaces lo0.0 host-inbound-
traffic system-services ike
 user@vsrx-mnha-n0# set security zones security-zone left interfaces lo0.0 host-inbound-
```

```
traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone left interfaces ge-0/0/3.100 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone left interfaces ge-0/0/3.100 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set security zones security-zone left interfaces ge-0/0/3.100 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone right interfaces ge-0/0/4.101 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone right interfaces ge-0/0/4.101 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set security zones security-zone right interfaces ge-0/0/4.101 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone untrust interfaces ge-0/0/0.102 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone untrust interfaces ge-0/0/0.102 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone untrust interfaces ge-0/0/0.102 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set interfaces st0 unit 0 family inet
```

b. Configure groups for Multinode High Availability monitoring options.

```
[edit groups monitor-simple]
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 src-ip 10.0.31.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 interface ge-0/0/0.100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 src-ip 10.0.33.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 interface ge-0/0/0.101
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
```

```
liveliness 10.0.38.1 src-ip 10.0.38.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 interface ge-0/0/0.102
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
interface ge-0/0/0
```

c. Configure groups for Multinode High Availability advance monitoring options.

```
[edit groups monitor-advanced]
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints object-threshold 200
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip threshold 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 weight 50
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 weight 50
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers object-threshold 200
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness threshold 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 src-ip 10.0.31.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 interface ge-0/0/3.100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 weight 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 src-ip 10.0.33.10
```

```
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 interface ge-0/0/4.101
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 weight 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 src-ip 10.0.38.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 interface ge-0/0/0.102
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 weight 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor srg-
threshold 200
```

2. Configure node specific statements on active node.

   a. Configure groups for synchronization.

```
[edit]
 user@vsrx-mnha-n0# set groups mnha-sync-icl system commit peers vsrx-mnha-n1 routing-
instance icl
 user@vsrx-mnha-n0# set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n1 inet
172.26.0.2
 user@vsrx-mnha-n0# set groups icd chassis high-availability local-id local-forwarding-ip
172.26.0.11
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
172.26.0.12
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
interface lo0.1
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
liveness-detection minimum-interval 1000
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
liveness-detection multiplier 5
 user@vsrx-mnha-n0# set groups icd interfaces lo0 unit 1 family inet address 172.26.0.11/32
```

NOTE: You can synchronize the configuration across any interface you choose – normally either through the interface configured as ICL or fxp0, the out-of-band management interface. In this example, we've used the configuration synchronization over ICL.

b. Configure Multinode High Availability related statements.

```
[edit]
 user@vsrx-mnha-n0# set chassis high-availability local-id 1
 user@vsrx-mnha-n0# set chassis high-availability local-id local-ip 172.26.0.1
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 peer-ip 172.26.0.2
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 interface lo0.1
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 routing-instance icl
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 vpn-profile icl
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 liveness-detection multiplier 3
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 0 peer-id 2
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 deployment-
type routing
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 peer-id 2
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip 10.0.30.1
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip src-ip 172.25.0.0
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 active-
signal-route 172.24.0.1
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 backup-
signal-route 172.24.0.0
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 prefix-list
srg1-prefix routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 managed-
services ipsec
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 process-
packet-on-backup
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
priority 100
```

c. Configure IPsec VPN options.

```
[edit]
 user@vsrx-mnha-n0# set security ike proposal ike-prop authentication-method pre-shared-
keys
 user@vsrx-mnha-n0# set security ike proposal ike-prop dh-group group20
 user@vsrx-mnha-n0# set security ike proposal ike-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n0# set security ike proposal ike-prop lifetime-seconds 28800
 user@vsrx-mnha-n0# set security ike policy ike-policy proposals ike-prop
 user@vsrx-mnha-n0# set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n0# set security ike policy icl proposals ike-prop
 user@vsrx-mnha-n0# set security ike policy icl pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n0# set security ike gateway icl ike-policy icl
 user@vsrx-mnha-n0# set security ike gateway icl version v2-only
 user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop lifetime-seconds 3600
 user@vsrx-mnha-n0# set security ipsec policy ipsec-policy perfect-forward-secrecy keys
group20
 user@vsrx-mnha-n0# set security ipsec policy ipsec-policy proposals ipsec-prop
 user@vsrx-mnha-n0# set security ipsec vpn icl ha-link-encryption
 user@vsrx-mnha-n0# set security ipsec vpn icl ike gateway icl
 user@vsrx-mnha-n0# set security ipsec vpn icl ike ipsec-policy ipsec-policy
```

d. Configure security zone.

```
[edit]
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/3.36 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/3.36 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/3.36 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ike
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services high-availability
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ssh
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
```

```
traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bfd
```

e. Configure interfaces.

```
[edit]
user@vsrx-mnha-n0# set interfaces ge-0/0/0 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/0 unit 102 description for-monitoring
user@vsrx-mnha-n0# set interfaces ge-0/0/0 unit 102 vlan-id 38
user@vsrx-mnha-n0# set interfaces ge-0/0/0 unit 102 family inet address 10.0.38.10/24
user@vsrx-mnha-n0# set interfaces ge-0/0/1 description lab-ha-1
user@vsrx-mnha-n0# set interfaces ge-0/0/1 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/1 mtu 9000
user@vsrx-mnha-n0# set interfaces ge-0/0/1 unit 39 description icl-n1
user@vsrx-mnha-n0# set interfaces ge-0/0/1 unit 39 vlan-id 39
user@vsrx-mnha-n0# set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.1/24
user@vsrx-mnha-n0# set interfaces ge-0/0/3 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 36 description icd
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 36 vlan-id 36
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 36 family inet address 10.0.36.10/24
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 100 description vr-left-r1
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 100 vlan-id 31
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 100 family inet address 10.0.31.10/24
user@vsrx-mnha-n0# set interfaces ge-0/0/4 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/4 unit 101 description vr-right-r2
user@vsrx-mnha-n0# set interfaces ge-0/0/4 unit 101 vlan-id 33
user@vsrx-mnha-n0# set interfaces ge-0/0/4 unit 101 family inet address 10.0.33.10/24
user@vsrx-mnha-n0# set interfaces lo0 unit 0 description "Floating IP"
user@vsrx-mnha-n0# set interfaces lo0 unit 0 family inet address 172.25.0.0/32
user@vsrx-mnha-n0# set interfaces lo0 unit 1 description ICL
user@vsrx-mnha-n0# set interfaces lo0 unit 1 family inet address 172.26.0.1/32
```

f. Configure policy options.

```
[edit]
 user@vsrx-mnha-n0# set policy-options prefix-list export-int 0.0.0.0/0
```

```
 user@vsrx-mnha-n0# set policy-options prefix-list export-int 172.25.0.0/32
 user@vsrx-mnha-n0# set policy-options prefix-list export-uplink 10.0.30.0/24
 user@vsrx-mnha-n0# set policy-options prefix-list export-uplink 10.0.35.0/24
 user@vsrx-mnha-n0# set policy-options prefix-list srg1-prefix 172.25.0.0/32
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-r1 term 10 from
interface lo0.1
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-r1 term 10 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-r1 term 100 then reject
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-to-n1 term 10 from
interface lo0.1
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-to-n1 term 10 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-to-n1 term 100 then
reject
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 from prefix-
list export-int
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 from
condition srg1_backup
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 then as-path-
prepend 65031
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 20 from prefix-
list export-int
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 20 from
condition srg1_active
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 20 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 90 from prefix-
list export-int
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 90 then as-path-
prepend "65031 65031"
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 90 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 100 then reject
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 from
prefix-list export-uplink
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 from
condition srg1_backup
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 then as-
path-prepend 65031
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 20 from
prefix-list export-uplink
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 20 from
```

```
condition srg1_active
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 20 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 90 from
prefix-list export-uplink
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 90 then as-
path-prepend "65031 65031"
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 90 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 100 then
reject
 user@vsrx-mnha-n0# set policy-options condition srg1_active if-route-exists 172.24.0.1/32
 user@vsrx-mnha-n0# set policy-options condition srg1_active if-route-exists table inet.0
 user@vsrx-mnha-n0# set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
 user@vsrx-mnha-n0# set policy-options condition srg1_backup if-route-exists table inet.0
```

g. Configure routing instances and routing option.

```
[edit]
 user@vsrx-mnha-n0# set routing-instances icl instance-type virtual-router
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.0.36.1
export export-icl-r1
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.0.36.1
peer-as 65030
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.1.39.2
export export-icl-to-n1
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.1.39.2
peer-as 65032
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp local-as 65031
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp bfd-liveness-detection minimum-
interval 500
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n0# set routing-instances icl interface ge-0/0/1.39
 user@vsrx-mnha-n0# set routing-instances icl interface ge-0/0/3.36
 user@vsrx-mnha-n0# set routing-instances icl interface lo0.1
 user@vsrx-mnha-n0# set routing-instances vr instance-type virtual-router
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1
export export-to-int
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1
peer-as 65030
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1
```

```
export export-to-int
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1
peer-as 65035
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.38.1 export export-to-uplink
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.38.1 peer-as 65039
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp local-as 65031
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp bfd-liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n0# set routing-instances vr interface ge-0/0/0.102
 user@vsrx-mnha-n0# set routing-instances vr interface ge-0/0/3.100
 user@vsrx-mnha-n0# set routing-instances vr interface ge-0/0/4.101
 user@vsrx-mnha-n0# set routing-instances vr interface lo0.0
```

h. Apply configuration groups.

```
[edit]
 user@vsrx-mnha-n0# set apply-groups mnha-sync
 user@vsrx-mnha-n0# set apply-groups mnha-sync-icl
 user@vsrx-mnha-n0# set apply-groups monitor-advanced
 user@vsrx-mnha-n0# set apply-groups icd
```

i. Configure options for the peer node participating in commit synchronization.

```
[edit]
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n1 user user
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n1 authentication "$ABC123"
user@vsrx-mnha-n0# set system services netconf ssh
user@vsrx-mnha-n0# set system static-host-mapping vsrx-mnha-n1 inet 172.26.0.2
```

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

This configuration enables the node to take the configuration commands entered under the sync group and push them to the other node, using the IP address and credentials defined. You must repeat this configuration on the node-01, with changed IP address and hostnames.

3. Configure node-specific statements on the backup node (SRX-02).

   a. Configure groups for enabling peer sync through ICL.

```
[edit]
user@vsrx-mnha-n1# set groups mnha-sync-icl system commit peers vsrx-mnha-n0 routing-
instance icl
user@vsrx-mnha-n1# set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n0 inet
172.26.0.1
user@vsrx-mnha-n1# set groups icd chassis high-availability local-id local-forwarding-ip
172.26.0.12
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
172.26.0.11
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
interface lo0.1
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
liveness-detection minimum-interval 1000
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
liveness-detection multiplier 5
user@vsrx-mnha-n1# set groups icd interfaces lo0 unit 1 family inet address 172.26.0.12/32
```

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

   b. Configure Multinode High Availability options.

```
[edit]
user@vsrx-mnha-n1# set chassis high-availability local-id 2
 user@vsrx-mnha-n1# set chassis high-availability local-id local-ip 172.26.0.2
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 peer-ip 172.26.0.1
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 interface lo0.1
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 routing-instance icl
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 vpn-profile icl
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 liveness-detection multiplier 3
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 0 peer-id 1
```

```
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 deployment-
type routing
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 peer-id 1
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip 10.0.30.1
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip src-ip 172.25.0.0
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip routing-instance vr
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 active-
signal-route 172.24.0.1
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 backup-
signal-route 172.24.0.0
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 prefix-list
srg1-prefix routing-instance vr
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 managed-
services ipsec
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 process-
packet-on-backup
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
priority 200
```

c. Configure IPsec VPN related options.

```
[edit]
user@vsrx-mnha-n1# set security ike proposal ike-prop authentication-method pre-shared-keys
 user@vsrx-mnha-n1# set security ike proposal ike-prop dh-group group20
 user@vsrx-mnha-n1# set security ike proposal ike-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n1# set security ike proposal ike-prop lifetime-seconds 28800
 user@vsrx-mnha-n1# set security ike policy ike-policy proposals ike-prop
 user@vsrx-mnha-n1# set security ike policy ike-policy pre-shared-key ascii-text "$ABC13"
 user@vsrx-mnha-n1# set security ike policy icl proposals ike-prop
 user@vsrx-mnha-n1# set security ike policy icl pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n1# set security ike gateway icl ike-policy icl
 user@vsrx-mnha-n1# set security ike gateway icl version v2-only
 user@vsrx-mnha-n1# set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n1# set security ipsec proposal ipsec-prop lifetime-seconds 3600
 user@vsrx-mnha-n1# set security ipsec policy ipsec-policy perfect-forward-secrecy keys
group20
 user@vsrx-mnha-n1# set security ipsec policy ipsec-policy proposals ipsec-prop
 user@vsrx-mnha-n1# set security ipsec vpn icl ha-link-encryption
```

```
 user@vsrx-mnha-n1# set security ipsec vpn icl ike gateway icl
 user@vsrx-mnha-n1# set security ipsec vpn icl ike ipsec-policy ipsec-policy
```

d. Configure security zones.

```
[edit]
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/3.37 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/3.37 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/3.37 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ping
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ike
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services high-availability
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ssh
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic protocols bfd
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bfd
```

e. Configure interfaces.

```
[edit]
user@vsrx-mnha-n1# set interfaces ge-0/0/0 description for-monitoring
user@vsrx-mnha-n1# set interfaces ge-0/0/0 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/0 unit 102 description vr-uplink-r2
user@vsrx-mnha-n1# set interfaces ge-0/0/0 unit 102 vlan-id 39
user@vsrx-mnha-n1# set interfaces ge-0/0/0 unit 102 family inet address 10.0.39.10/24
user@vsrx-mnha-n1# set interfaces ge-0/0/1 description br-lab-ha-1
user@vsrx-mnha-n1# set interfaces ge-0/0/1 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/1 mtu 9000
user@vsrx-mnha-n1# set interfaces ge-0/0/1 unit 39 description icl-n0
```

```
user@vsrx-mnha-n1# set interfaces ge-0/0/1 unit 39 vlan-id 39
user@vsrx-mnha-n1# set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.2/24
user@vsrx-mnha-n1# set interfaces ge-0/0/3 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 37 description icd
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 37 vlan-id 37
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 37 family inet address 10.0.37.10/24
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 100 description vr-left-r1
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 100 vlan-id 32
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 100 family inet address 10.0.32.10/24
user@vsrx-mnha-n1# set interfaces ge-0/0/4 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/4 unit 101 description vr-right-r2
user@vsrx-mnha-n1# set interfaces ge-0/0/4 unit 101 vlan-id 34
user@vsrx-mnha-n1# set interfaces ge-0/0/4 unit 101 family inet address 10.0.34.10/24
user@vsrx-mnha-n1# set interfaces lo0 unit 0 description "Floating IP"
user@vsrx-mnha-n1# set interfaces lo0 unit 0 family inet address 172.25.0.0/32
user@vsrx-mnha-n1# set interfaces lo0 unit 1 description ICL
user@vsrx-mnha-n1# set interfaces lo0 unit 1 family inet address 172.26.0.2/32
```

f. Configure policy options.

```
[edit]
 user@vsrx-mnha-n1# set policy-options prefix-list export-int 0.0.0.0/0
 user@vsrx-mnha-n1# set policy-options prefix-list export-int 172.25.0.0/32
 user@vsrx-mnha-n1# set policy-options prefix-list export-uplink 10.0.30.0/24
 user@vsrx-mnha-n1# set policy-options prefix-list export-uplink 10.0.35.0/24
 user@vsrx-mnha-n1# set policy-options prefix-list srg1-prefix 172.25.0.0/32
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-r1 term 10 from
interface lo0.1
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-r1 term 10 then accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-r1 term 100 then reject
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-to-n0 term 10 from
interface lo0.1
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-to-n0 term 10 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-to-n0 term 100 then
reject
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 from prefix-
list export-int
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 from
condition srg1_backup
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 then as-path-
prepend 65032
```

```
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 then accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 20 from prefix-
list export-int
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 20 from
condition srg1_active
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 20 then accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 90 from prefix-
list export-int
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 90 then as-path-
prepend "65032 65032 65032"
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 90 then accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 100 then reject
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 from
prefix-list export-uplink
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 from
condition srg1_backup
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 then as-
path-prepend 65032
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 20 from
prefix-list export-uplink
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 20 from
condition srg1_active
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 20 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 90 from
prefix-list export-uplink
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 90 then as-
path-prepend "65032 65032 65032"
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 90 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 100 then
reject
 user@vsrx-mnha-n1# set policy-options condition srg1_active if-route-exists 172.24.0.1/32
 user@vsrx-mnha-n1# set policy-options condition srg1_active if-route-exists table inet.0
 user@vsrx-mnha-n1# set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
 user@vsrx-mnha-n1# set policy-options condition srg1_backup if-route-exists table inet.0
```

g. Configure routing-instances and routing options.

```
[edit]
 user@vsrx-mnha-n1# set routing-instances icl instance-type virtual-router
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.0.37.1
export export-icl-r1
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.0.37.1
peer-as 65030
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.1.39.1
export export-icl-to-n0
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.1.39.1
peer-as 65031
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp local-as 65032
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp bfd-liveness-detection minimum-
interval 500
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n1# set routing-instances icl interface ge-0/0/1.39
 user@vsrx-mnha-n1# set routing-instances icl interface ge-0/0/3.37
 user@vsrx-mnha-n1# set routing-instances icl interface lo0.1
 user@vsrx-mnha-n1# set routing-instances vr instance-type virtual-router
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1
export export-to-int
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1
peer-as 65030
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1
export export-to-int
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1
peer-as 65035
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.39.1 export export-to-uplink
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.39.1 peer-as 65039
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp local-as 65032
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp bfd-liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n1# set routing-instances vr interface ge-0/0/0.102
 user@vsrx-mnha-n1# set routing-instances vr interface ge-0/0/3.100
```

```
user@vsrx-mnha-n1# set routing-instances vr interface ge-0/0/4.101
user@vsrx-mnha-n1# set routing-instances vr interface lo0.0
```

h. Configure options for the peers participating in commit synchronization. Configure options for the peers participating in commit synchronization. This configuration enables the node to take the configuration commands entered under the sync group and push them to the other node, using the IP address and credentials defined.

```
[edit]
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n0 user user
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n1 authentication "$ABC123"
user@vsrx-mnha-n0# set system services netconf ssh
user@vsrx-mnha-n0# set system static-host-mapping vsrx-mnha-n0 inet 172.26.0.1
```

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

4. Use the following command to configure the `commit` command to automatically perform a peers-synchronize action between peers:

```
[edit]
user@host# set system commit peers-synchronize
```

The local peer (or requesting peer) on which you enable the `peers-synchronize` statement copies and loads its configuration to the remote (or responding) peer.

NOTE: Use the `set security ssh-known-hosts fetch-from-server` and `set security ssh-known hosts host`statements to include the other node as known host. When you commit the configuration, the system displays following message:

```
user@03-vsrx-mnha-n0# set security ssh-known-hosts fetch-from-server
03-vsrx-mnha-n1 The authenticity of host '03-vsrx-mnha-n1 (172.26.0.2)' can't be
established.
ECDSA key fingerprint is SHA256:abc123.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vsrx-mnha-2, 172.26.0.2' (ECDSA) to the list of known
hosts.
```

You must add the SSH key fingerprint for the Multinode High Availability peer. This step is needed for the configuration synchronization to work.

Verification

IN THIS SECTION

Use the following show commands to verify the feature in this example.

Command	Verification Task
show chassis high availability information	Displays Multinode High Availability details including status.
show chassis high-availability peer-info	Displays details such as peer node, connection details, and packet statistics of the peer node in a Multinode High Availability setup.
show chassis high-availability services-redundancy-group	Display the service redundancy group information in a Multinode High Availability setup.

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 commands on both nodes:

```
use@vsrx-mnha-n0> 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: 172.26.0.1
Local Forwarding IP: 172.26.0.11
HA Peer Information:

    Peer Id: 2        IP address: 172.26.0.2    Interface: lo0.1
    Routing Instance: icl
    Encrypted: YES

                  Conn State: UP
    Configured BFD Detection Time: 3 * 1000ms
    Cold Sync Status: COMPLETE
    Peer Forwarding IP: 172.26.0.12             Interface: lo0.1
    Peer ICD Conn State: UP

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: 100
        Preemption: DISABLED
```

```
        Process Packet In Backup State: YES
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
          Peer Id: 2
          Status : ACTIVE
          Health Status: HEALTHY
          Failover Readiness: N/A
```

```
user@vsrx-mnha-n1# 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: 172.26.0.2
Local Forwarding IP: 172.26.0.12
HA Peer Information:

    Peer Id: 1        IP address: 172.26.0.1    Interface: lo0.1
    Routing Instance: icl
    Encrypted: YES

                  Conn State: UP
    Configured BFD Detection Time: 3 * 1000ms
    Cold Sync Status: COMPLETE
    Peer Forwarding IP: 172.26.0.11             Interface: lo0.1
    Peer ICD Conn State: UP

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: 200
        Preemption: DISABLED
        Process Packet In Backup State: YES
        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
```

Meaning

Verify these details from the command output:

- Local node and peer node details such as IP address and ID.
- `Node Status: ONLINE` indicates that the node is up.
- `Conn State: UP` indicates that the ICL link is established and operational.
- `Peer ICD Conn State: UP` indicates that the ICD link is established and operational.
- `Encrypted: YES` indicates that ICL connection is encrypted.
- `Peer Information` Services Redundancy Group indicates peer node is healthy and ready for failover.

Check Multinode High Availability Peer Node Details

Purpose

View details of the peer node in the Multinode High Availability setup.

Action

From operational mode, run the following command:

```
user@vsrx-mnha-n0> show chassis high-availability peer-info

HA Peer Information:

   Peer-ID: 2        IP address: 172.26.0.2    Interface: lo0.1
   Routing Instance: icl
   Encrypted: YES    Conn State: UP
   Cold Sync Status: COMPLETE
   Peer Forwarding IP: 172.26.0.12             Interface: lo0.1
   Peer ICD Conn State: UP
   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           12              9
        SRG Status Ack            9              9
        Attribute Msg             7              4
        Attribute Ack             4              4
```

Meaning

You can get the following details from the command output:

- `Peer ID: 2` shows the ID of the other node.
- `Conn State: UP` and `Peer ICD Conn State: UP` indicate that the both ICL and ICD link are established.
- `Packet Statistics` shows packets transferred between the nodes.

Check Multinode High Availability Service Redundancy Group Details

Purpose

View and verify the details of the Multinode High Availability SRG details.

Action

From operational mode, run the following command:

SRX-01 Device

```
                user@vsrx-mnha-n0> 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: BACKUP
        Activeness Priority: 100
        Preemption: DISABLED
        Process Packet In Backup State: YES
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
        Peer Id: 2


                Status : ACTIVE
          Health Status: HEALTHY
          Failover Readiness: N/A

        Signal Route Info:
          Active Signal Route:
          IP: 172.24.0.1
          Routing Instance: default
          Status: NOT INSTALLED

          Backup Signal Route:
          IP: 172.24.0.0
```

```
          Routing Instance: default
          Status: INSTALLED

        Split-brain Prevention Probe Info:
          DST-IP: 10.0.30.1
          SRC-IP: 172.25.0.0
          Routing Instance: vr
          Type: ICMP Probe
          Status: NOT RUNNING
          Result: N/A           Reason: N/A

        SRG Path Monitor Info:
          SRG Monitor Status: UP
          SRG Monitor Threshold: 200
          SRG Monitor Weight: 0
          SRG Monitor Failed Objects: NONE

          Object Name: routers
          Object Status: UP
          Object Monitored Entries: [ BFD ]
          Object Failures: [ BFD ]
          Object Threshold: 200
          Object Current Weight: 100

          Object Name: endpoints
          Object Status: UP
          Object Monitored Entries: [ IP ]
          Object Failures: [ IP ]
          Object Threshold: 200
          Object Current Weight: 100

IP SRGID Table:
        SRGID    IP Prefix                                Routing Table
        1        172.25.0.0/32                            vr
```

Now run the same command on SRX-02 device and notice the command output differences such as Status, Peer Information and so on.

```
user@vsrx-mnha-n1> 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: DISABLED
        Process Packet In Backup State: YES
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
          Peer Id: 1


                  Status : BACKUP
          Health Status: HEALTHY
          Failover Readiness: READY

        Signal Route Info:
          Active Signal Route:
          IP: 172.24.0.1
          Routing Instance: default
          Status: INSTALLED

          Backup Signal Route:
          IP: 172.24.0.0
          Routing Instance: default
          Status: NOT INSTALLED

        Split-brain Prevention Probe Info:
          DST-IP: 10.0.30.1
          SRC-IP: 172.25.0.0
          Routing Instance: vr
          Type: ICMP Probe
          Status: NOT RUNNING
          Result: N/A           Reason: N/A


        SRG Path Monitor Info:
```

```
          SRG Monitor Status: UP
          SRG Monitor Threshold: 200
          SRG Monitor Weight: 0
          SRG Monitor Failed Objects: NONE

          Object Name: routers
          Object Status: UP
          Object Monitored Entries: [ BFD ]
          Object Failures: [ BFD ]
          Object Threshold: 200
          Object Current Weight: 100

          Object Name: endpoints
          Object Status: UP
          Object Monitored Entries: [ IP ]
          Object Failures: [ IP ]
          Object Threshold: 200
          Object Current Weight: 100

IP SRGID Table:
        SRGID    IP Prefix                                  Routing Table
        1        172.25.0.0/32                              vr
```

Meaning

Verify these details from the command output:

- `Deployment Type: ROUTING` indicates the Multinode High Availability is setup for Layer 3 (Routing) mode.
- `Status: BACKUP` indicates currently the node is operating as Backup node.
- `Peer Information` provides peer node details such as deployment type, status, and active and back up signal routes.
- The output also indicates configured monitoring options and failure events (if any).

Set Commands on All Devices

IN THIS SECTION

Device Configured as Active Node (vsrx-mnha-n0)

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

```
set groups mnha-sync when peers vsrx-mnha-n0
set groups mnha-sync when peers vsrx-mnha-n1
set groups mnha-sync security ike proposal ike-prop authentication-method pre-shared-keys
set groups mnha-sync security ike proposal ike-prop dh-group group20
set groups mnha-sync security ike proposal ike-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ike proposal ike-prop lifetime-seconds 28800
set groups mnha-sync security ike policy ike-policy proposals ike-prop
set groups mnha-sync security ike policy ike-policy pre-shared-key ascii-text "$ABc123"
set groups mnha-sync security ike policy icl proposals ike-prop
set groups mnha-sync security ike gateway r1 ike-policy ike-policy
set groups mnha-sync security ike gateway r1 address 10.0.30.1
set groups mnha-sync security ike gateway r1 dead-peer-detection probe-idle-tunnel
set groups mnha-sync security ike gateway r1 dead-peer-detection interval 5
set groups mnha-sync security ike gateway r1 dead-peer-detection threshold 5
set groups mnha-sync security ike gateway r1 external-interface lo0.0
set groups mnha-sync security ike gateway r1 version v2-only
set groups mnha-sync security ike gateway icl ike-policy icl
set groups mnha-sync security ike gateway icl version v2-only
set groups mnha-sync security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ipsec proposal ipsec-prop lifetime-seconds 3600
set groups mnha-sync security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
```

```
set groups mnha-sync security ipsec policy ipsec-policy proposals ipsec-prop
set groups mnha-sync security ipsec vpn r1 bind-interface st0.0
set groups mnha-sync security ipsec vpn r1 ike gateway r1
set groups mnha-sync security ipsec vpn r1 ike ipsec-policy ipsec-policy
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 local-ip 10.0.35.11/32
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 remote-ip 10.0.30.11/32
set groups mnha-sync security ipsec vpn r1 establish-tunnels immediately
set groups mnha-sync security ipsec vpn icl ha-link-encryption
set groups mnha-sync security ipsec vpn icl ike gateway icl
set groups mnha-sync security ipsec vpn icl ike ipsec-policy ipsec-policy
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match source-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match destination-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match application
any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit then permit
set groups mnha-sync security policies global policy internal match source-address any
set groups mnha-sync security policies global policy internal match destination-address any
set groups mnha-sync security policies global policy internal match application any
set groups mnha-sync security policies global policy internal match from-zone right
set groups mnha-sync security policies global policy internal match from-zone vpn
set groups mnha-sync security policies global policy internal match from-zone left
set groups mnha-sync security policies global policy internal match to-zone left
set groups mnha-sync security policies global policy internal match to-zone right
set groups mnha-sync security policies global policy internal match to-zone vpn
set groups mnha-sync security policies global policy internal then permit
set groups mnha-sync security policies global policy internal then log session-close
set groups mnha-sync security policies global policy untrust match source-address any
set groups mnha-sync security policies global policy untrust match destination-address any
set groups mnha-sync security policies global policy untrust match application any
set groups mnha-sync security policies global policy untrust match from-zone left
set groups mnha-sync security policies global policy untrust match from-zone right
set groups mnha-sync security policies global policy untrust match to-zone untrust
set groups mnha-sync security policies global policy untrust then permit
set groups mnha-sync security zones security-zone vpn interfaces st0.0
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ike
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
```

```
traffic protocols bgp
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic protocols bgp
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bgp
set groups mnha-sync interfaces st0 unit 0 family inet
set groups mnha-sync-icl system commit peers vsrx-mnha-n1 routing-instance icl
set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n1 inet 172.26.0.2
set groups icd chassis high-availability local-id local-forwarding-ip 172.26.0.11
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip 172.26.0.12
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip interface lo0.1
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection minimum-
interval 1000
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection
multiplier 5
set groups icd interfaces lo0 unit 1 family inet address 172.26.0.11/32
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 src-ip 10.0.31.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 interface ge-0/0/0.100
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 src-ip 10.0.33.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 interface ge-0/0/0.101
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
```

```
liveliness 10.0.38.1 src-ip 10.0.38.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 interface ge-0/0/0.102
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor
interface ge-0/0/0
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 src-ip 10.0.31.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 interface ge-0/0/3.100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 src-ip 10.0.33.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 interface ge-0/0/4.101
```

```
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 src-ip 10.0.38.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 interface ge-0/0/0.102
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor srg-
threshold 200
set apply-groups mnha-sync
set apply-groups mnha-sync-icl
set apply-groups monitor-advanced
set apply-groups icd
set system commit peers vsrx-mnha-n1 user user
set system commit peers vsrx-mnha-n1 authentication "$ABC123"
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 172.26.0.1
set chassis high-availability peer-id 2 peer-ip 172.26.0.2
set chassis high-availability peer-id 2 interface lo0.1
set chassis high-availability peer-id 2 routing-instance icl
set chassis high-availability peer-id 2 vpn-profile icl
set chassis high-availability peer-id 2 liveness-detection minimum-interval 1000
set chassis high-availability peer-id 2 liveness-detection multiplier 3
set chassis high-availability services-redundancy-group 0 peer-id 2
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 10.0.30.1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip src-ip
172.25.0.0
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip routing-
instance vr
set chassis high-availability services-redundancy-group 1 active-signal-route 172.24.0.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 172.24.0.0
set chassis high-availability services-redundancy-group 1 prefix-list srg1-prefix routing-
instance vr
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 process-packet-on-backup
set chassis high-availability services-redundancy-group 1 activeness-priority 100
```

```
set security ike proposal ike-prop authentication-method pre-shared-keys
set security ike proposal ike-prop dh-group group20
set security ike proposal ike-prop encryption-algorithm aes-256-gcm
set security ike proposal ike-prop lifetime-seconds 28800
set security ike policy ike-policy proposals ike-prop
set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
set security ike policy icl proposals ike-prop
set security ike policy icl pre-shared-key ascii-text "$ABC123."
set security ike gateway icl ike-policy icl
set security ike gateway icl version v2-only
set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set security ipsec proposal ipsec-prop lifetime-seconds 3600
set security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
set security ipsec policy ipsec-policy proposals ipsec-prop
set security ipsec vpn icl ha-link-encryption
set security ipsec vpn icl ike gateway icl
set security ipsec vpn icl ike ipsec-policy ipsec-policy
set security zones security-zone icl interfaces ge-0/0/3.36 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/3.36 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/3.36 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ping
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ike
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services high-
availability
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ssh
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bfd
set interfaces ge-0/0/0 description for-monitoring
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 102 description vr-uplink-r2
set interfaces ge-0/0/0 unit 102 vlan-id 38
set interfaces ge-0/0/0 unit 102 family inet address 10.0.38.10/24
set interfaces ge-0/0/1 description br-lab-ha-1
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 mtu 9000
set interfaces ge-0/0/1 unit 39 description icl-n1
set interfaces ge-0/0/1 unit 39 vlan-id 39
set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.1/24
set interfaces ge-0/0/3 vlan-tagging
```

```
set interfaces ge-0/0/3 unit 36 description icl-r1
set interfaces ge-0/0/3 unit 36 vlan-id 36
set interfaces ge-0/0/3 unit 36 family inet address 10.0.36.10/24
set interfaces ge-0/0/3 unit 100 description vr-left-r1
set interfaces ge-0/0/3 unit 100 vlan-id 31
set interfaces ge-0/0/3 unit 100 family inet address 10.0.31.10/24
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 101 description vr-right-r2
set interfaces ge-0/0/4 unit 101 vlan-id 33
set interfaces ge-0/0/4 unit 101 family inet address 10.0.33.10/24
set interfaces lo0 unit 0 description "Floating IP"
set interfaces lo0 unit 0 family inet address 172.25.0.0/32
set interfaces lo0 unit 1 description ICL
set interfaces lo0 unit 1 family inet address 172.26.0.1/32
set policy-options prefix-list export-int 0.0.0.0/0
set policy-options prefix-list export-int 172.25.0.0/32
set policy-options prefix-list export-uplink 10.0.30.0/24
set policy-options prefix-list export-uplink 10.0.35.0/24
set policy-options prefix-list srg1-prefix 172.25.0.0/32
set policy-options policy-statement export-icl-r1 term 10 from interface lo0.1
set policy-options policy-statement export-icl-r1 term 10 then accept
set policy-options policy-statement export-icl-r1 term 100 then reject
set policy-options policy-statement export-icl-to-n1 term 10 from interface lo0.1
set policy-options policy-statement export-icl-to-n1 term 10 then accept
set policy-options policy-statement export-icl-to-n1 term 100 then reject
set policy-options policy-statement export-to-int term 10 from prefix-list export-int
set policy-options policy-statement export-to-int term 10 from condition srg1_backup
set policy-options policy-statement export-to-int term 10 then as-path-prepend 65031
set policy-options policy-statement export-to-int term 10 then accept
set policy-options policy-statement export-to-int term 20 from prefix-list export-int
set policy-options policy-statement export-to-int term 20 from condition srg1_active
set policy-options policy-statement export-to-int term 20 then accept
set policy-options policy-statement export-to-int term 90 from prefix-list export-int
set policy-options policy-statement export-to-int term 90 then as-path-prepend "65031 65031"
set policy-options policy-statement export-to-int term 90 then accept
set policy-options policy-statement export-to-int term 100 then reject
set policy-options policy-statement export-to-uplink term 10 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 10 from condition srg1_backup
set policy-options policy-statement export-to-uplink term 10 then as-path-prepend 65031
set policy-options policy-statement export-to-uplink term 10 then accept
set policy-options policy-statement export-to-uplink term 20 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 20 from condition srg1_active
set policy-options policy-statement export-to-uplink term 20 then accept
```

```
set policy-options policy-statement export-to-uplink term 90 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 90 then as-path-prepend "65031 65031"
set policy-options policy-statement export-to-uplink term 90 then accept
set policy-options policy-statement export-to-uplink term 100 then reject
set policy-options condition srg1_active if-route-exists 172.24.0.1/32
set policy-options condition srg1_active if-route-exists table inet.0
set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
set policy-options condition srg1_backup if-route-exists table inet.0
set routing-instances icl instance-type virtual-router
set routing-instances icl protocols bgp group icl neighbor 10.0.36.1 export export-icl-r1
set routing-instances icl protocols bgp group icl neighbor 10.0.36.1 peer-as 65030
set routing-instances icl protocols bgp group icl neighbor 10.1.39.2 export export-icl-to-n1
set routing-instances icl protocols bgp group icl neighbor 10.1.39.2 peer-as 65032
set routing-instances icl protocols bgp local-as 65031
set routing-instances icl protocols bgp bfd-liveness-detection minimum-interval 500
set routing-instances icl protocols bgp bfd-liveness-detection multiplier 3
set routing-instances icl interface ge-0/0/1.39
set routing-instances icl interface ge-0/0/3.36
set routing-instances icl interface lo0.1
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1 export export-to-int
set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1 peer-as 65030
set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1 export export-to-int
set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1 peer-as 65035
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.38.1 export export-to-uplink
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.38.1 peer-as 65039
set routing-instances vr protocols bgp local-as 65031
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.102
set routing-instances vr interface ge-0/0/3.100
set routing-instances vr interface ge-0/0/4.101
set routing-instances vr interface lo0.0
```

Device Configured as Backup Node (SRX-02)

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

```
set groups mnha-sync-icl system commit peers vsrx-mnha-n0 routing-instance icl
set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n0 inet 172.26.0.1
set groups mnha-sync when peers vsrx-mnha-n0
set groups mnha-sync when peers vsrx-mnha-n1
set groups mnha-sync security ike proposal ike-prop authentication-method pre-shared-keys
set groups mnha-sync security ike proposal ike-prop dh-group group20
set groups mnha-sync security ike proposal ike-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ike proposal ike-prop lifetime-seconds 28800
set groups mnha-sync security ike policy ike-policy proposals ike-prop
set groups mnha-sync security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
set groups mnha-sync security ike policy icl proposals ike-prop
set groups mnha-sync security ike gateway r1 ike-policy ike-policy
set groups mnha-sync security ike gateway r1 address 10.0.30.1
set groups mnha-sync security ike gateway r1 dead-peer-detection probe-idle-tunnel
set groups mnha-sync security ike gateway r1 dead-peer-detection interval 5
set groups mnha-sync security ike gateway r1 dead-peer-detection threshold 5
set groups mnha-sync security ike gateway r1 external-interface lo0.0
set groups mnha-sync security ike gateway r1 version v2-only
set groups mnha-sync security ike gateway icl ike-policy icl
set groups mnha-sync security ike gateway icl version v2-only
set groups mnha-sync security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ipsec proposal ipsec-prop lifetime-seconds 3600
set groups mnha-sync security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
set groups mnha-sync security ipsec policy ipsec-policy proposals ipsec-prop
set groups mnha-sync security ipsec vpn r1 bind-interface st0.0
set groups mnha-sync security ipsec vpn r1 ike gateway r1
set groups mnha-sync security ipsec vpn r1 ike ipsec-policy ipsec-policy
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 local-ip 10.0.35.11/32
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 remote-ip 10.0.30.11/32
set groups mnha-sync security ipsec vpn r1 establish-tunnels immediately
set groups mnha-sync security ipsec vpn icl ha-link-encryption
set groups mnha-sync security ipsec vpn icl ike gateway icl
set groups mnha-sync security ipsec vpn icl ike ipsec-policy ipsec-policy
set groups mnha-sync security flow tcp-mss ipsec-vpn mss 1400
set groups mnha-sync security flow tcp-session strict-syn-check
```

```
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match source-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match destination-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match application
any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit then permit
set groups mnha-sync security policies global policy internal match source-address any
set groups mnha-sync security policies global policy internal match destination-address any
set groups mnha-sync security policies global policy internal match application any
set groups mnha-sync security policies global policy internal match from-zone right
set groups mnha-sync security policies global policy internal match from-zone vpn
set groups mnha-sync security policies global policy internal match from-zone left
set groups mnha-sync security policies global policy internal match to-zone left
set groups mnha-sync security policies global policy internal match to-zone right
set groups mnha-sync security policies global policy internal match to-zone vpn
set groups mnha-sync security policies global policy internal then permit
set groups mnha-sync security policies global policy internal then log session-close
set groups mnha-sync security policies global policy untrust match source-address any
set groups mnha-sync security policies global policy untrust match destination-address any
set groups mnha-sync security policies global policy untrust match application any
set groups mnha-sync security policies global policy untrust match from-zone left
set groups mnha-sync security policies global policy untrust match from-zone right
set groups mnha-sync security policies global policy untrust match to-zone untrust
set groups mnha-sync security policies global policy untrust then permit
set groups mnha-sync security zones security-zone vpn interfaces st0.0
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ike
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic protocols bgp
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic protocols bgp
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
```

```
traffic system-services ping
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bgp
set groups mnha-sync interfaces st0 unit 0 family inet
set groups icd chassis high-availability local-id local-forwarding-ip 172.26.0.12
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip 172.26.0.11
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip interface lo0.1
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection minimum-
interval 1000
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection
multiplier 5
set groups icd interfaces lo0 unit 1 family inet address 172.26.0.12/32
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 src-ip 10.0.32.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 interface ge-0/0/3.100
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 src-ip 10.0.34.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 interface ge-0/0/4.101
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 src-ip 10.0.39.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 interface ge-0/0/0.102
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor
interface ge-0/0/0
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
```

```
monitor-object endpoints ip threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 src-ip 10.0.32.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 interface ge-0/0/3.100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 src-ip 10.0.34.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 interface ge-0/0/4.101
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 src-ip 10.0.39.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 interface ge-0/0/0.102
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 weight 100
```

```
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor srg-
threshold 200
set apply-groups mnha-sync
set apply-groups mnha-sync-icl
set apply-groups monitor-advanced
set apply-groups icd
set system commit peers vsrx-mnha-n0 user user
set system commit peers vsrx-mnha-n0 authentication "$ABC123"
set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 172.26.0.2
set chassis high-availability peer-id 1 peer-ip 172.26.0.1
set chassis high-availability peer-id 1 interface lo0.1
set chassis high-availability peer-id 1 routing-instance icl
set chassis high-availability peer-id 1 vpn-profile icl
set chassis high-availability peer-id 1 liveness-detection minimum-interval 1000
set chassis high-availability peer-id 1 liveness-detection multiplier 3
set chassis high-availability services-redundancy-group 0 peer-id 1
set chassis high-availability services-redundancy-group 1 deployment-type routing
set chassis high-availability services-redundancy-group 1 peer-id 1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip 10.0.30.1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip src-ip
172.25.0.0
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip routing-
instance vr
set chassis high-availability services-redundancy-group 1 active-signal-route 172.24.0.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 172.24.0.0
set chassis high-availability services-redundancy-group 1 prefix-list srg1-prefix routing-
instance vr
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 process-packet-on-backup
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set security ike proposal ike-prop authentication-method pre-shared-keys
set security ike proposal ike-prop dh-group group20
set security ike proposal ike-prop encryption-algorithm aes-256-gcm
set security ike proposal ike-prop lifetime-seconds 28800
set security ike policy ike-policy proposals ike-prop
set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
set security ike policy icl proposals ike-prop
set security ike policy icl pre-shared-key ascii-text "$ABC123"
set security ike gateway icl ike-policy icl
set security ike gateway icl version v2-only
set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set security ipsec proposal ipsec-prop lifetime-seconds 3600
```

```
set security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
set security ipsec policy ipsec-policy proposals ipsec-prop
set security ipsec vpn icl ha-link-encryption
set security ipsec vpn icl ike gateway icl
set security ipsec vpn icl ike ipsec-policy ipsec-policy
set security zones security-zone icl interfaces ge-0/0/3.37 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/3.37 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/3.37 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ping
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ike
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services high-
availability
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ssh
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bfd
set interfaces ge-0/0/0 description for-monitoring
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 102 description vr-uplink-r2
set interfaces ge-0/0/0 unit 102 vlan-id 39
set interfaces ge-0/0/0 unit 102 family inet address 10.0.39.10/24
set interfaces ge-0/0/1 description br-lab-ha-1
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 mtu 9000
set interfaces ge-0/0/1 unit 39 description icl-n0
set interfaces ge-0/0/1 unit 39 vlan-id 39
set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.2/24
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 37 description icl-r1
set interfaces ge-0/0/3 unit 37 vlan-id 37
set interfaces ge-0/0/3 unit 37 family inet address 10.0.37.10/24
set interfaces ge-0/0/3 unit 100 description vr-left-r1
set interfaces ge-0/0/3 unit 100 vlan-id 32
set interfaces ge-0/0/3 unit 100 family inet address 10.0.32.10/24
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 101 description vr-right-r2
set interfaces ge-0/0/4 unit 101 vlan-id 34
set interfaces ge-0/0/4 unit 101 family inet address 10.0.34.10/24
set interfaces lo0 unit 0 description "Floating IP"
set interfaces lo0 unit 0 family inet address 172.25.0.0/32
```

```
set interfaces lo0 unit 1 description ICL
set interfaces lo0 unit 1 family inet address 172.26.0.2/32
set policy-options prefix-list export-int 0.0.0.0/0
set policy-options prefix-list export-int 172.25.0.0/32
set policy-options prefix-list export-uplink 10.0.30.0/24
set policy-options prefix-list export-uplink 10.0.35.0/24
set policy-options prefix-list srg1-prefix 172.25.0.0/32
set policy-options policy-statement export-icl-r1 term 10 from interface lo0.1
set policy-options policy-statement export-icl-r1 term 10 then accept
set policy-options policy-statement export-icl-r1 term 100 then reject
set policy-options policy-statement export-icl-to-n0 term 10 from interface lo0.1
set policy-options policy-statement export-icl-to-n0 term 10 then accept
set policy-options policy-statement export-icl-to-n0 term 100 then reject
set policy-options policy-statement export-to-int term 10 from prefix-list export-int
set policy-options policy-statement export-to-int term 10 from condition srg1_backup
set policy-options policy-statement export-to-int term 10 then as-path-prepend 65032
set policy-options policy-statement export-to-int term 10 then accept
set policy-options policy-statement export-to-int term 20 from prefix-list export-int
set policy-options policy-statement export-to-int term 20 from condition srg1_active
set policy-options policy-statement export-to-int term 20 then accept
set policy-options policy-statement export-to-int term 90 from prefix-list export-int
set policy-options policy-statement export-to-int term 90 then as-path-prepend "65032 65032
65032"
set policy-options policy-statement export-to-int term 90 then accept
set policy-options policy-statement export-to-int term 100 then reject
set policy-options policy-statement export-to-uplink term 10 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 10 from condition srg1_backup
set policy-options policy-statement export-to-uplink term 10 then as-path-prepend 65032
set policy-options policy-statement export-to-uplink term 10 then accept
set policy-options policy-statement export-to-uplink term 20 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 20 from condition srg1_active
set policy-options policy-statement export-to-uplink term 20 then accept
set policy-options policy-statement export-to-uplink term 90 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 90 then as-path-prepend "65032 65032
65032"
set policy-options policy-statement export-to-uplink term 90 then accept
set policy-options policy-statement export-to-uplink term 100 then reject
set policy-options condition srg1_active if-route-exists 172.24.0.1/32
set policy-options condition srg1_active if-route-exists table inet.0
set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
set policy-options condition srg1_backup if-route-exists table inet.0
set routing-instances icl instance-type virtual-router
set routing-instances icl protocols bgp group icl neighbor 10.0.37.1 export export-icl-r1
```

```
set routing-instances icl protocols bgp group icl neighbor 10.0.37.1 peer-as 65030
set routing-instances icl protocols bgp group icl neighbor 10.1.39.1 export export-icl-to-n0
set routing-instances icl protocols bgp group icl neighbor 10.1.39.1 peer-as 65031
set routing-instances icl protocols bgp local-as 65032
set routing-instances icl protocols bgp bfd-liveness-detection minimum-interval 500
set routing-instances icl protocols bgp bfd-liveness-detection multiplier 3
set routing-instances icl interface ge-0/0/1.39
set routing-instances icl interface ge-0/0/3.37
set routing-instances icl interface lo0.1
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1 export export-to-int
set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1 peer-as 65030
set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1 export export-to-int
set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1 peer-as 65035
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.39.1 export export-to-uplink
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.39.1 peer-as 65039
set routing-instances vr protocols bgp local-as 65032
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.102
set routing-instances vr interface ge-0/0/3.100
set routing-instances vr interface ge-0/0/4.101
set routing-instances vr interface lo0.0
```

Router 1 (Device Configured as Router)

```
set security policies default-policy permit-all
set security zones security-zone left host-inbound-traffic system-services ping
set security zones security-zone left host-inbound-traffic system-services ike
set security zones security-zone left host-inbound-traffic protocols bgp
set security zones security-zone left host-inbound-traffic protocols bfd
set security zones security-zone left interfaces ge-0/0/2.30
set security zones security-zone left interfaces ge-0/0/0.31
set security zones security-zone left interfaces ge-0/0/1.32
set security zones security-zone left interfaces st0.0
set security zones security-zone left enable-reverse-reroute
set security zones security-zone icl host-inbound-traffic system-services ping
set security zones security-zone icl host-inbound-traffic protocols bgp
set security zones security-zone icl host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces ge-0/0/0.36
set security zones security-zone icl interfaces ge-0/0/1.37
```

```
set interfaces ge-0/0/0 description br-lab-1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 31 description vr-mnha-n0
set interfaces ge-0/0/0 unit 31 vlan-id 31
set interfaces ge-0/0/0 unit 31 family inet address 10.0.31.1/24
set interfaces ge-0/0/0 unit 36 description icl-n0
set interfaces ge-0/0/0 unit 36 vlan-id 36
set interfaces ge-0/0/0 unit 36 family inet address 10.0.36.1/24
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 32 description vr-mnha-n1
set interfaces ge-0/0/1 unit 32 vlan-id 32
set interfaces ge-0/0/1 unit 32 family inet address 10.0.32.1/24
set interfaces ge-0/0/1 unit 37 description icl-n1
set interfaces ge-0/0/1 unit 37 vlan-id 37
set interfaces ge-0/0/1 unit 37 family inet address 10.0.37.1/24
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 30 description vr-linux-1
set interfaces ge-0/0/2 unit 30 vlan-id 30
set interfaces ge-0/0/2 unit 30 family inet address 10.0.30.1/24
set interfaces st0 unit 0 family inet
set policy-options policy-statement export-icl-n0 term 10 from interface ge-0/0/1.37
set policy-options policy-statement export-icl-n0 term 10 then accept
set policy-options policy-statement export-icl-n0 term 100 then reject
set policy-options policy-statement export-icl-n1 term 10 from interface ge-0/0/0.36
set policy-options policy-statement export-icl-n1 term 10 then accept
set policy-options policy-statement export-icl-n1 term 100 then reject
set policy-options policy-statement export-to-mnha-fws term 10 from interface ge-0/0/2.30
set policy-options policy-statement export-to-mnha-fws term 10 then accept
set policy-options policy-statement export-to-mnha-fws term 100 then reject
set routing-instances icl instance-type virtual-router
set routing-instances icl protocols bgp group icl local-as 65030
set routing-instances icl protocols bgp group icl bfd-liveness-detection minimum-interval 500
set routing-instances icl protocols bgp group icl bfd-liveness-detection multiplier 3
set routing-instances icl protocols bgp group icl neighbor 10.0.36.10 export export-icl-n0
set routing-instances icl protocols bgp group icl neighbor 10.0.36.10 peer-as 65031
set routing-instances icl protocols bgp group icl neighbor 10.0.37.10 export export-icl-n1
set routing-instances icl protocols bgp group icl neighbor 10.0.37.10 peer-as 65032
set routing-instances icl interface ge-0/0/0.36
set routing-instances icl interface ge-0/0/1.37
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp group mnha-n0 neighbor 10.0.31.10 peer-as 65031
set routing-instances vr protocols bgp group mnha-n1 neighbor 10.0.32.10 peer-as 65032
set routing-instances vr protocols bgp export export-to-mnha-fws
```

```
set routing-instances vr protocols bgp local-as 65030
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.31
set routing-instances vr interface ge-0/0/1.32
set routing-instances vr interface ge-0/0/2.30
set routing-instances vr interface st0.0
```

Router 2 (Device Configured as Router)

```
set security policies default-policy permit-all
set security zones security-zone right host-inbound-traffic system-services ping
set security zones security-zone right host-inbound-traffic protocols bgp
set security zones security-zone right host-inbound-traffic protocols bfd
set security zones security-zone right interfaces ge-0/0/0.33
set security zones security-zone right interfaces ge-0/0/1.34
set security zones security-zone right interfaces ge-0/0/2.35
set security zones security-zone right enable-reverse-reroute
set security zones security-zone trust tcp-rst
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust host-inbound-traffic protocols bgp
set security zones security-zone trust host-inbound-traffic protocols bfd
set security zones security-zone trust interfaces ge-0/0/0.39
set security zones security-zone trust interfaces ge-0/0/0.38
set interfaces ge-0/0/0 description br-lab-1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 33 description vr-mnha-n0
set interfaces ge-0/0/0 unit 33 vlan-id 33
set interfaces ge-0/0/0 unit 33 family inet address 10.0.33.1/24
set interfaces ge-0/0/0 unit 38 description uplink-mnha-n0
set interfaces ge-0/0/0 unit 38 vlan-id 38
set interfaces ge-0/0/0 unit 38 family inet address 10.0.38.1/24
set interfaces ge-0/0/0 unit 39 description uplink-mnha-n1
set interfaces ge-0/0/0 unit 39 vlan-id 39
set interfaces ge-0/0/0 unit 39 family inet address 10.0.39.1/24
set interfaces ge-0/0/1 description br-poc-mgmt
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 34 description vr-mnha-n1
set interfaces ge-0/0/1 unit 34 vlan-id 34
set interfaces ge-0/0/1 unit 34 family inet address 10.0.34.1/24
set interfaces ge-0/0/2 vlan-tagging
```

```
set interfaces ge-0/0/2 unit 35 description vr-linux-2
set interfaces ge-0/0/2 unit 35 vlan-id 35
set interfaces ge-0/0/2 unit 35 family inet address 10.0.35.1/24
set policy-options policy-statement export-default term 10 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement export-default term 10 then accept
set policy-options policy-statement export-default term 100 then reject
set policy-options policy-statement export-to-mnha-fws term 10 from interface ge-0/0/0.35
set policy-options policy-statement export-to-mnha-fws term 10 then accept
set policy-options policy-statement export-to-mnha-fws term 100 then reject
set policy-options policy-statement import-from-n1 from neighbor 10.0.34.10
set policy-options policy-statement import-from-n1 then local-preference 1000
set routing-instances uplink instance-type virtual-router
set routing-instances uplink routing-options static route 0.0.0.0/0 next-hop 172.30.192.1
set routing-instances uplink protocols bgp family inet unicast loops 1
set routing-instances uplink protocols bgp group trust export export-default
set routing-instances uplink protocols bgp group trust local-as 65039
set routing-instances uplink protocols bgp group trust bfd-liveness-detection minimum-interval
1000
set routing-instances uplink protocols bgp group trust bfd-liveness-detection multiplier 3
set routing-instances uplink protocols bgp group trust neighbor 10.0.38.10 peer-as 65031
set routing-instances uplink protocols bgp group trust neighbor 10.0.39.10 peer-as 65032
set routing-instances uplink interface ge-0/0/0.38
set routing-instances uplink interface ge-0/0/0.39
set routing-instances uplink interface ge-0/0/1.0
deactivate routing-instances uplink interface ge-0/0/1.0
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp family inet unicast loops 1
set routing-instances vr protocols bgp group mnha-n0 neighbor 10.0.33.10 peer-as 65031
set routing-instances vr protocols bgp group mnha-n1 neighbor 10.0.34.10 import import-from-n1
set routing-instances vr protocols bgp group mnha-n1 neighbor 10.0.34.10 peer-as 65032
set routing-instances vr protocols bgp export export-to-mnha-fws
set routing-instances vr protocols bgp local-as 65035
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.33
set routing-instances vr interface ge-0/0/1.34
set routing-instances vr interface ge-0/0/2.35
```

Show Configuration Output

IN THIS SECTION

From configuration mode, confirm your configuration by entering the `show high availability`, `show groups`, and other details. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

SRX-01 (Active Node)

```
[edit]
user@vsrx-mnha-n0# show chassis high-availability
local-id {
    1;
    local-ip 172.26.0.1;
}
peer-id 2 {
    peer-ip 172.26.0.2;
    interface lo0.1;
    routing-instance icl;
    vpn-profile icl;
    liveness-detection {
        minimum-interval 1000;
        multiplier 3;
    }
}
services-redundancy-group 0 {
    peer-id {
        2;
    }
}
services-redundancy-group 1 {
    deployment-type routing;
    peer-id {
```

```
        2;
    }
    activeness-probe {
        dest-ip {
            10.0.30.1;
            src-ip 172.25.0.0;
            routing-instance vr;
        }
    }
    active-signal-route {
        172.24.0.1;
    }
    backup-signal-route {
        172.24.0.0;
    }
    prefix-list srg1-prefix {
        routing-instance vr;
    }
    managed-services ipsec;
    process-packet-on-backup;
    activeness-priority 100;
}
```

```
[edit]
user@vsrx-mnha-n0# show groups mnha-sync
when {
    peers [ vsrx-mnha-n0 vsrx-mnha-n1 ];
}
security {
    ike {
        proposal ike-prop {
            authentication-method pre-shared-keys;
            dh-group group20;
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 28800;
        }
        policy ike-policy {
            proposals ike-prop;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        policy icl {
```

```
            proposals ike-prop;
        }
        gateway r1 {
            ike-policy ike-policy;
            address 10.0.30.1;
            dead-peer-detection {
                probe-idle-tunnel;
                interval 5;
                threshold 5;
            }
            external-interface lo0.0;
            version v2-only;
        }
        gateway icl {
            ike-policy icl;
            version v2-only;
        }
    }
    ipsec {
        proposal ipsec-prop {
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 3600;
        }
        policy ipsec-policy {
            perfect-forward-secrecy {
                keys group20;
            }
            proposals ipsec-prop;
        }
        vpn r1 {
            bind-interface st0.0;
            ike {
                gateway r1;
                ipsec-policy ipsec-policy;
            }
            traffic-selector ts1 {
                local-ip 10.0.35.11/32;
                remote-ip 10.0.30.11/32;
            }
            establish-tunnels immediately;
        }
        vpn icl {
            ha-link-encryption;
```

```
            ike {
                gateway icl;
                ipsec-policy ipsec-policy;
            }
        }
    }
    policies {
        from-zone icl to-zone icl {
            policy permit {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        global {
            policy internal {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                    from-zone [ right vpn left ];
                    to-zone [ left right vpn ];
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
            }
            policy untrust {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                    from-zone [ left right ];
                    to-zone untrust;
                }
```

```
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone vpn {
            interfaces {
                st0.0;
            }
        }
        security-zone left {
            interfaces {
                lo0.0 {
                    host-inbound-traffic {
                        system-services {
                            ike;
                            ping;
                        }
                    }
                }
                ge-0/0/3.100 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
                            bgp;
                            bfd;
                        }
                    }
                }
            }
        }
        security-zone right {
            interfaces {
                ge-0/0/4.101 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
```

```
                            bgp;
                            bfd;
                        }
                    }
                }
            }
        }
        security-zone untrust {
            interfaces {
                ge-0/0/0.102 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
                            bfd;
                            bgp;
                        }
                    }
                }
            }
        }
    }
}
interfaces {
    st0 {
        unit 0 {
            family inet;
        }
    }
}
routing-instances {
    vr {
        interface st0.0;
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups monitor-simple
chassis {
    high-availability {
```

```
        services-redundancy-group 1 {
            monitor {
                bfd-liveliness 10.0.31.1 {
                    src-ip 10.0.31.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.100;
                }
                bfd-liveliness 10.0.33.1 {
                    src-ip 10.0.33.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.101;
                }
                bfd-liveliness 10.0.38.1 {
                    src-ip 10.0.38.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.102;
                }
                interface {
                    ge-0/0/0;
                }
            }
        }
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups monitor-advanced
chassis {
    high-availability {
        services-redundancy-group 1 {
            monitor {
                monitor-object endpoints {
                    object-threshold 200;
                    ip {
                        threshold 100;
                        destination-ip 10.0.30.10 {
                            routing-instance vr;
                            weight 50;
```

```
                    }
                    destination-ip 10.0.35.10 {
                        routing-instance vr;
                        weight 50;
                    }
                }
            }
            monitor-object routers {
                object-threshold 200;
                bfd-liveliness {
                    threshold 100;
                    destination-ip 10.0.31.1 {
                        src-ip 10.0.31.10;
                        routing-instance vr;
                        session-type singlehop;
                        interface ge-0/0/3.100;
                        weight 100;
                    }
                    destination-ip 10.0.33.1 {
                        src-ip 10.0.33.10;
                        routing-instance vr;
                        session-type singlehop;
                        interface ge-0/0/4.101;
                        weight 100;
                    }
                    destination-ip 10.0.38.1 {
                        src-ip 10.0.38.10;
                        routing-instance vr;
                        session-type singlehop;
                        interface ge-0/0/0.102;
                        weight 100;
                    }
                }
            }
            srg-threshold 200;
        }
    }
```

```
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups mnha-sync-icl
system {
    commit {
        peers {
            vsrx-mnha-n1 {
                routing-instance icl;
            }
        }
    }
    static-host-mapping {
        vsrx-mnha-n1 inet 172.26.0.2;
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups icd
chassis {
    high-availability {
        local-id {
            local-forwarding-ip 172.26.0.11;
        }
        peer-id 2 {
            peer-forwarding-ip {
                172.26.0.12;
                interface lo0.1;
                liveness-detection {
                    minimum-interval 1000;
                    multiplier 5;
                }
            }
        }
    }
}
interfaces {
```

```
    lo0 {
        unit 1 {
            family inet {
                address 172.26.0.11/32;
            }
        }
    }
}
```

SRX-02

```
[edit]
user@vsrx-mnha-n1# show chassis high-availability
local-id {
    2;
    local-ip 172.26.0.2;
}
peer-id 1 {
    peer-ip 172.26.0.1;
    interface lo0.1;
    routing-instance icl;
    vpn-profile icl;
    liveness-detection {
        minimum-interval 1000;
        multiplier 3;
    }
}
services-redundancy-group 0 {
    peer-id {
        1;
    }
}
services-redundancy-group 1 {
    deployment-type routing;
    peer-id {
        1;
    }
    activeness-probe {
        dest-ip {
            10.0.30.1;
            src-ip 172.25.0.0;
```

```
            routing-instance vr;
        }
    }
    active-signal-route {
        172.24.0.1;
    }
    backup-signal-route {
        172.24.0.0;
    }
    prefix-list srg1-prefix {
        routing-instance vr;
    }
    managed-services ipsec;
    process-packet-on-backup;
    activeness-priority 200;
}
```

```
[edit]
user@vsrx-mnha-n1# show groups mnha-sync

when {
    peers [ vsrx-mnha-n0 vsrx-mnha-n1 ];
}
security {
    ike {
        proposal ike-prop {
            authentication-method pre-shared-keys;
            dh-group group20;
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 28800;
        }
        policy ike-policy {
            proposals ike-prop;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        policy icl {
            proposals ike-prop;
        }
        gateway r1 {
            ike-policy ike-policy;
            address 10.0.30.1;
```

```
            dead-peer-detection {
                probe-idle-tunnel;
                interval 5;
                threshold 5;
            }
            external-interface lo0.0;
            version v2-only;
        }
        gateway icl {
            ike-policy icl;
            version v2-only;
        }
    }
    ipsec {
        proposal ipsec-prop {
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 3600;
        }
        policy ipsec-policy {
            perfect-forward-secrecy {
                keys group20;
            }
            proposals ipsec-prop;
        }
        vpn r1 {
            bind-interface st0.0;
            ike {
                gateway r1;
                ipsec-policy ipsec-policy;
            }
            traffic-selector ts1 {
                local-ip 10.0.35.11/32;
                remote-ip 10.0.30.11/32;
            }
            establish-tunnels immediately;
        }
        vpn icl {
            ha-link-encryption;
            ike {
                gateway icl;
                ipsec-policy ipsec-policy;
            }
        }
```

```
    }
    flow {
        tcp-mss {
            ipsec-vpn {
                mss 1400;
            }
        }
        tcp-session {
            strict-syn-check;
        }
    }
    policies {
        from-zone icl to-zone icl {
            policy permit {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        global {
            policy internal {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                    from-zone [ right vpn left ];
                    to-zone [ left right vpn ];
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
            }
            policy untrust {
                match {
                    source-address any;
```

```
                destination-address any;
                application any;
                from-zone [ left right ];
                to-zone untrust;
            }
            then {
                permit;
            }
        }
    }
}
zones {
    security-zone vpn {
        interfaces {
            st0.0;
        }
    }
    security-zone left {
        interfaces {
            lo0.0 {
                host-inbound-traffic {
                    system-services {
                        ike;
                        ping;
                    }
                }
            }
            ge-0/0/3.100 {
                host-inbound-traffic {
                    system-services {
                        ping;
                    }
                    protocols {
                        bgp;
                        bfd;
                    }
                }
            }
        }
    }
    security-zone right {
        interfaces {
            ge-0/0/4.101 {
```

```
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
                            bgp;
                            bfd;
                        }
                    }
                }
            }
        }
        security-zone untrust {
            interfaces {
                ge-0/0/0.102 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
                            bfd;
                            bgp;
                        }
                    }
                }
            }
        }
    }
}
interfaces {
    st0 {
        unit 0 {
            family inet;
        }
    }
}
routing-instances {
    vr {
        interface st0.0;
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups monitor-simple
chassis {
    high-availability {
        services-redundancy-group 1 {
            monitor {
                bfd-liveliness 10.0.32.1 {
                    src-ip 10.0.32.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/3.100;
                }
                bfd-liveliness 10.0.34.1 {
                    src-ip 10.0.34.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/4.101;
                }
                bfd-liveliness 10.0.39.1 {
                    src-ip 10.0.39.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.102;
                }
                interface {
                    ge-0/0/0;
                }
            }
        }
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups monitor-advanced
chassis {
    high-availability {
        services-redundancy-group 1 {
            monitor {
                monitor-object endpoints {
                    object-threshold 200;
```

```
                    ip {
                        threshold 100;
                        destination-ip 10.0.30.10 {
                            routing-instance vr;
                            weight 50;
                        }
                        destination-ip 10.0.35.10 {
                            routing-instance vr;
                            weight 50;
                        }
                    }
                }
                monitor-object routers {
                    object-threshold 200;
                    bfd-liveliness {
                        threshold 100;
                        destination-ip 10.0.32.1 {
                            src-ip 10.0.32.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/3.100;
                            weight 100;
                        }
                        destination-ip 10.0.34.1 {
                            src-ip 10.0.34.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/4.101;
                            weight 100;
                        }
                        destination-ip 10.0.39.1 {
                            src-ip 10.0.39.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/0.102;
                            weight 100;
                        }
                    }
                }
                srg-threshold 200;
            }
        }
```

```
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups mnha-sync-icl

system {
    commit {
        peers {
            vsrx-mnha-n0 {
                routing-instance icl;
            }
        }
    }
    static-host-mapping {
        vsrx-mnha-n0 inet 172.26.0.1;
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups icd
chassis {
    high-availability {
        local-id {
            local-forwarding-ip 172.26.0.12;
        }
        peer-id 1 {
            peer-forwarding-ip {
                172.26.0.11;
                interface lo0.1;
                liveness-detection {
                    minimum-interval 1000;
                    multiplier 5;
                }
            }
        }
    }
}
interfaces {
    lo0 {
```

```
        unit 1 {
            family inet {
                address 172.26.0.12/32;
            }
        }
    }
}
```

Example: Configure Multinode High Availability with Junos OS Configuration Groups

SUMMARY

Read this topic to understand how to configure Multinode High Availability using Junos OS configuration groups.

IN THIS SECTION

In Multinode High Availability, two Junos OS security devices act as independent devices. These devices have unique hostname and the IP address on fxp0 interface. You can configure Multinode High Availability using Junos groups statements. To ensure identical security configurations and posture between two devices, you can configure groups for Multinode High Availability setup. Multinode High Availability nodes synchronize configurations exclusively based on this group method.

When you need to configure statements that are common on both nodes, you can use one of the following approaches:

- You can configure common configuration (like security) on one device and manually copy and paste on the other device. Or you can use some external tool (example: scripting) to copy the same configuration snippets to both devices as applicable.
- Use common Junos group configuration synchronized between both nodes (but edited on one device). This approach includes:
  - Configure the feature/function as part of groups. These configuration groups enable you to create smaller, more logically constructed configuration files
  - Synchronize the configuration using the `edit system commit peers-synchronize` option.
  - Mention the device name in the group using the `when peers <device-name>` statement.

  When you enable configuration synchronization (by using the peers-synchronize option) on both the devices in a Multinode High Availability, configuration settings you configure on one peer under [groups] will automatically sync to the other peer upon the commit action.

  For more details on configuration groups, see Use Configuration Groups to Quickly Configure Devices .

  Note that on Security Director or Security Director Cloud, the system manages reusable configuration snippets, similar to Junos Groups, through the use of policy templates and shared objects.

In this example, we'll configure Multinode High Availability using Junos groups statements.

TIP:

Table 56: Time Estimates

Reading Time	30 minutes
Configuration Time	60 minutes

Example Prerequisites

Table 52 on page 914 lists the hardware and software components that support the configuration.

Table 57: Requirements

Hardware requirements	Supported firewalls and virtual firewalls.

Software requirements	We've tested this example using Junos OS Release 24.4R1. See Feature Explorer for details about support for Junos OS Groups and Multinode High Availability. Junos IKE package is required on your firewall for Multinode High Availability configuration. This package is available as a default package or as an optional package on the device. See Support for Junos IKE Package for details. If the package is not installed by default on your firewall, use the following command to install it: `user@host> request system software add optional://junos-ike.tgz` You require this step for ICL encryption.
Licensing requirements	No separate license is required to configure Multinode High Availability. Licenses needed for features such as IDP, Application Identification, Juniper ATP Cloud are unique to each firewall and need to be set on each device. Licenses are unique to each device and cannot be shared between the nodes in a Multinode High Availability setup. Therefore, you must use identical licenses on both the nodes.

Before You Begin

Know more	Using groups configuration in Multinode High Availability simplifies the setup by allowing you to create reusable configuration blocks. These groups can be applied across different parts of the configuration, ensuring consistency and reducing the need for repetitive entries. This approach makes the configuration files more concise and logically structured. Group configuration helps in easy maintenance of configuration files on Juniper Networks devices.
Learn more	Multinode High Availability, Use Configuration Groups to Quickly Configure Devices

Functional Overview

Table 53 on page 915 provides a quick summary of the configuration components deployed in this example.

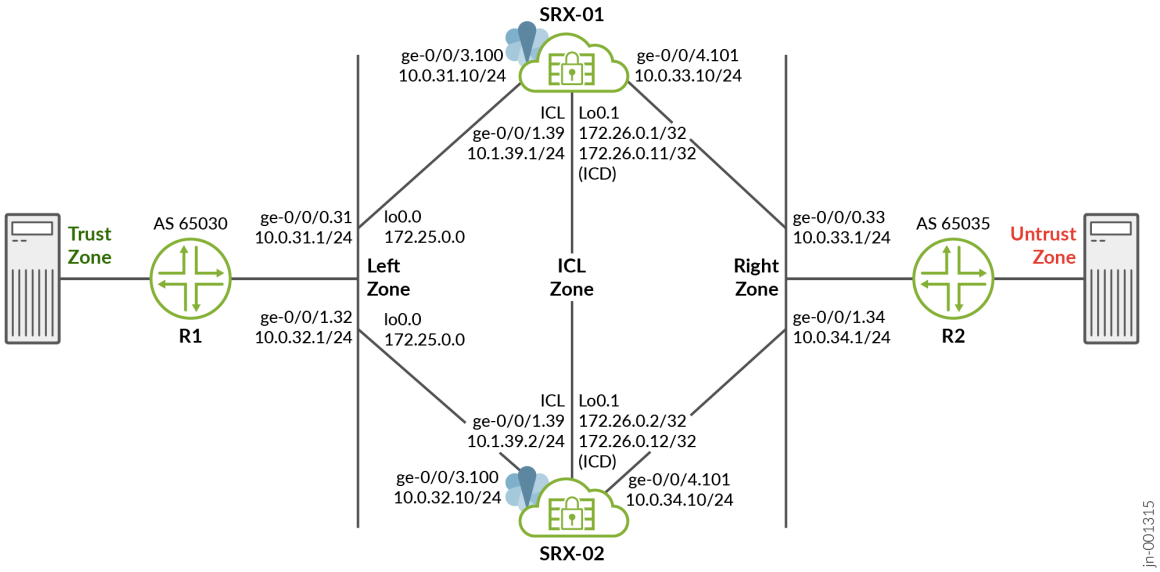
Table 58: Configuration Components

Technologies used	• High availability • Junos OS Configuration Groups • IPsec VPN • Routing policy • Routing options
Primary verification tasks	1. Verify the high availability on both the nodes in the setup. 2. Verify the Multinode High Availability data plane statistics.

Topology Illustration

Figure 69 on page 916 shows the topology used in this configuration example.

Figure 70: Multinode High Availability in Layer 3 Network



As shown in the topology, two SRX devices in MNHA are connected to adjacent routers (vSRX instances acting as routers). An encrypted logical interchassis link (ICL) connects the nodes. The nodes communicate with each other using a routable IP address (floating IP address) over the network. 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.

Loopback interfaces are used to host the IP addresses on firewalls and routers and the IP address on a loopback unit on each respective node is used for communication. In a typical high availability deployment, you have multiple routers and switches on the northbound and southbound sides of the network.

In this example, you'll create multiple configuration groups on devices and synchronize the configuration.

Topology Overview

Table 54 on page 917 shows the details on interfaces configuration used in this example.

Table 59: Interfaces and IP Address Configuration on Security Devices

Device	Interface	IP Address	Zone	Configured For
SRX-01	lo0.1	172.26.0.11/32	ICL Zone	Local forwarding address used to forward data packet over ICD link.
	lo0.1	172.26.0.1/32	ICL Zone	ICL
	lo0.0	172.25.0.0/32	Left Zone	Floating IP address
	ge-0/0/1.39	10.1.39.1/24	ICL Zone	ICL to node 0 connection
	• ge-0/0/3.100 • ge-0/0/4.101	• 10.0.31.10/24 • 10.0.33.10/24	• Left Zone • Right Zone	Connects to upstream and downstream routers.
SRX-02	lo0.1	172.26.0.12/32	ICL Zone	Local forwarding address used to forward data packet over ICD link.
	lo0.1	172.26.0.2/32	ICL Zone	ICL
	lo0.0	172.25.0.0/32	Left Zone	Floating IP address
	ge-0/0/1.39	10.1.39.2/24	ICL Zone	ICL to node 0 connection

Table 59: Interfaces and IP Address Configuration on Security Devices *(Continued)*

Device	Interface	IP Address	Zone	Configured For
	• ge-0/0/3.100 • ge-0/0/4.101	• 10.0.32.10/24 • 10.0.34.10/24	• Left Zone • Right Zone	Connects to upstream and downstream routers.

Table 60: Interfaces and IP Address Configuration on Routing Devices

Device	Interface	IP Address	Configured For
Router 1 (R1)	ge-0/0/0.31	10.0.31.1/24	Connects to SRX-01
	ge-0/0/1.32	10.0.32.1/24	Connects to SRX-02
Router 2 (R2)	ge-0/0/0.33	10.0.33.1/24	Connects to SRX-01
	ge-0/0/1.34	10.0.34.1/24	Connects to SRX-02

Configure Multinode High Availability Using Junos Group Statements

1. Configure common features/functions for Multinode High Availability Using Junos Group statements on active node (SRX-01).

   Note that we have included the term ‘sync’ in the group names as a naming convention to clearly indicate to admins and users that these groups are intended for synchronization.

   a. Configure groups for Multinode High Availability configuration. Within these groups, you can define security zones, security policies, IPsec tunnel definitions, and more.

```
[edit groups mnha-sync]
 user@vsrx-mnha-n0# set when peers vsrx-mnha-n0
 user@vsrx-mnha-n0# set when peers vsrx-mnha-n1
 user@vsrx-mnha-n0# set security ike proposal ike-prop authentication-method pre-shared-
keys
 user@vsrx-mnha-n0# set security ike proposal ike-prop dh-group group20
 user@vsrx-mnha-n0# set security ike proposal ike-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n0# set security ike proposal ike-prop lifetime-seconds 28800
 user@vsrx-mnha-n0# set security ike policy ike-policy proposals ike-prop
 user@vsrx-mnha-n0# set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n0# set security ike policy icl proposals ike-prop
```

```
user@vsrx-mnha-n0# set security ike gateway r1 ike-policy ike-policy
user@vsrx-mnha-n0# set security ike gateway r1 address 10.0.30.1
user@vsrx-mnha-n0# set security ike gateway r1 dead-peer-detection probe-idle-tunnel
user@vsrx-mnha-n0# set security ike gateway r1 dead-peer-detection interval 5
user@vsrx-mnha-n0# set security ike gateway r1 dead-peer-detection threshold 5
user@vsrx-mnha-n0# set security ike gateway r1 external-interface lo0.0
user@vsrx-mnha-n0# set security ike gateway r1 version v2-only
user@vsrx-mnha-n0# set security ike gateway icl ike-policy icl
user@vsrx-mnha-n0# set security ike gateway icl version v2-only
user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop lifetime-seconds 3600
user@vsrx-mnha-n0# set security ipsec policy ipsec-policy perfect-forward-secrecy keys
group20
user@vsrx-mnha-n0# set security ipsec policy ipsec-policy proposals ipsec-prop
user@vsrx-mnha-n0# set security ipsec vpn r1 bind-interface st0.0
user@vsrx-mnha-n0# set security ipsec vpn r1 ike gateway r1
user@vsrx-mnha-n0# set security ipsec vpn r1 ike ipsec-policy ipsec-policy
user@vsrx-mnha-n0# set security ipsec vpn r1 traffic-selector ts1 local-ip 10.0.35.11/32
user@vsrx-mnha-n0# set security ipsec vpn r1 traffic-selector ts1 remote-ip 10.0.30.11/32
user@vsrx-mnha-n0# set security ipsec vpn r1 establish-tunnels immediately
user@vsrx-mnha-n0# set security ipsec vpn icl ha-link-encryption
user@vsrx-mnha-n0# set security ipsec vpn icl ike gateway icl
user@vsrx-mnha-n0# set security ipsec vpn icl ike ipsec-policy ipsec-policy
user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit match
source-address any
user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit match
destination-address any
user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit match
application any
user@vsrx-mnha-n0# set security policies from-zone icl to-zone icl policy permit then
permit
user@vsrx-mnha-n0# set security policies global policy internal match source-address any
user@vsrx-mnha-n0# set security policies global policy internal match destination-address
any
user@vsrx-mnha-n0# set security policies global policy internal match application any
user@vsrx-mnha-n0# set security policies global policy internal match from-zone right
user@vsrx-mnha-n0# set security policies global policy internal match from-zone vpn
user@vsrx-mnha-n0# set security policies global policy internal match from-zone left
user@vsrx-mnha-n0# set security policies global policy internal match to-zone left
user@vsrx-mnha-n0# set security policies global policy internal match to-zone right
user@vsrx-mnha-n0# set security policies global policy internal match to-zone vpn
user@vsrx-mnha-n0# set security policies global policy internal then permit
user@vsrx-mnha-n0# set security policies global policy internal then log session-close
```

```
 user@vsrx-mnha-n0# set security policies global policy untrust match source-address any
 user@vsrx-mnha-n0# set security policies global policy untrust match destination-address
any
 user@vsrx-mnha-n0# set security policies global policy untrust match application any
 user@vsrx-mnha-n0# set security policies global policy untrust match from-zone left
 user@vsrx-mnha-n0# set security policies global policy untrust match from-zone right
 user@vsrx-mnha-n0# set security policies global policy untrust match to-zone untrust
 user@vsrx-mnha-n0# set security policies global policy untrust then permit
 user@vsrx-mnha-n0# set security zones security-zone vpn interfaces st0.0
 user@vsrx-mnha-n0# set security zones security-zone left interfaces lo0.0 host-inbound-
traffic system-services ike
 user@vsrx-mnha-n0# set security zones security-zone left interfaces lo0.0 host-inbound-
traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone left interfaces ge-0/0/3.100 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone left interfaces ge-0/0/3.100 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set security zones security-zone left interfaces ge-0/0/3.100 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone right interfaces ge-0/0/4.101 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone right interfaces ge-0/0/4.101 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set security zones security-zone right interfaces ge-0/0/4.101 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone untrust interfaces ge-0/0/0.102 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone untrust interfaces ge-0/0/0.102 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone untrust interfaces ge-0/0/0.102 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set interfaces st0 unit 0 family inet
```

b. Configure groups for Multinode High Availability monitoring options.

```
[edit groups monitor-simple]
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 src-ip 10.0.31.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
```

```
liveliness 10.0.31.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 interface ge-0/0/0.100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 src-ip 10.0.33.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 interface ge-0/0/0.101
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 src-ip 10.0.38.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 interface ge-0/0/0.102
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
interface ge-0/0/0
```

c. Configure groups for Multinode High Availability advance monitoring options.

```
[edit groups monitor-advanced]
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints object-threshold 200
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip threshold 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 weight 50
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 weight 50
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers object-threshold 200
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness threshold 100
```

```
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 src-ip 10.0.31.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 interface ge-0/0/3.100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 weight 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 src-ip 10.0.33.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 interface ge-0/0/4.101
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 weight 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 src-ip 10.0.38.10
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 session-type singlehop
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 interface ge-0/0/0.102
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 weight 100
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 monitor srg-
threshold 200
```

2. Configure node specific statements on active node.

   a. Configure groups for synchronization.

```
[edit]
 user@vsrx-mnha-n0# set groups mnha-sync-icl system commit peers vsrx-mnha-n1 routing-
instance icl
 user@vsrx-mnha-n0# set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n1 inet
172.26.0.2
 user@vsrx-mnha-n0# set groups icd chassis high-availability local-id local-forwarding-ip
```

```
172.26.0.11
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
172.26.0.12
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
interface lo0.1
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
liveness-detection minimum-interval 1000
 user@vsrx-mnha-n0# set groups icd chassis high-availability peer-id 2 peer-forwarding-ip
liveness-detection multiplier 5
 user@vsrx-mnha-n0# set groups icd interfaces lo0 unit 1 family inet address 172.26.0.11/32
```

NOTE: You can synchronize the configuration across any interface you choose – normally either through the interface configured as ICL or fxp0, the out-of-band management interface. In this example, we've used the configuration synchronization over ICL.

b. Configure Multinode High Availability related statements.

```
[edit]
 user@vsrx-mnha-n0# set chassis high-availability local-id 1
 user@vsrx-mnha-n0# set chassis high-availability local-id local-ip 172.26.0.1
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 peer-ip 172.26.0.2
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 interface lo0.1
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 routing-instance icl
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 vpn-profile icl
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n0# set chassis high-availability peer-id 2 liveness-detection multiplier 3
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 0 peer-id 2
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 deployment-
type routing
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 peer-id 2
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip 10.0.30.1
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip src-ip 172.25.0.0
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 active-
signal-route 172.24.0.1
```

```
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 backup-
signal-route 172.24.0.0
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 prefix-list
srg1-prefix routing-instance vr
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 managed-
services ipsec
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 process-
packet-on-backup
 user@vsrx-mnha-n0# set chassis high-availability services-redundancy-group 1 activeness-
priority 100
```

c. Configure IPsec VPN options.

```
[edit]
 user@vsrx-mnha-n0# set security ike proposal ike-prop authentication-method pre-shared-
keys
 user@vsrx-mnha-n0# set security ike proposal ike-prop dh-group group20
 user@vsrx-mnha-n0# set security ike proposal ike-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n0# set security ike proposal ike-prop lifetime-seconds 28800
 user@vsrx-mnha-n0# set security ike policy ike-policy proposals ike-prop
 user@vsrx-mnha-n0# set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n0# set security ike policy icl proposals ike-prop
 user@vsrx-mnha-n0# set security ike policy icl pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n0# set security ike gateway icl ike-policy icl
 user@vsrx-mnha-n0# set security ike gateway icl version v2-only
 user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n0# set security ipsec proposal ipsec-prop lifetime-seconds 3600
 user@vsrx-mnha-n0# set security ipsec policy ipsec-policy perfect-forward-secrecy keys
group20
 user@vsrx-mnha-n0# set security ipsec policy ipsec-policy proposals ipsec-prop
 user@vsrx-mnha-n0# set security ipsec vpn icl ha-link-encryption
 user@vsrx-mnha-n0# set security ipsec vpn icl ike gateway icl
 user@vsrx-mnha-n0# set security ipsec vpn icl ike ipsec-policy ipsec-policy
```

d. Configure security zone.

```
[edit]
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/3.36 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/3.36 host-
inbound-traffic protocols bgp
```

```
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/3.36 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ike
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services high-availability
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ssh
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic protocols bfd
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n0# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bfd
```

e. Configure interfaces.

```
[edit]
user@vsrx-mnha-n0# set interfaces ge-0/0/0 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/0 unit 102 description for-monitoring
user@vsrx-mnha-n0# set interfaces ge-0/0/0 unit 102 vlan-id 38
user@vsrx-mnha-n0# set interfaces ge-0/0/0 unit 102 family inet address 10.0.38.10/24
user@vsrx-mnha-n0# set interfaces ge-0/0/1 description lab-ha-1
user@vsrx-mnha-n0# set interfaces ge-0/0/1 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/1 mtu 9000
user@vsrx-mnha-n0# set interfaces ge-0/0/1 unit 39 description icl-n1
user@vsrx-mnha-n0# set interfaces ge-0/0/1 unit 39 vlan-id 39
user@vsrx-mnha-n0# set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.1/24
user@vsrx-mnha-n0# set interfaces ge-0/0/3 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 36 description icd
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 36 vlan-id 36
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 36 family inet address 10.0.36.10/24
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 100 description vr-left-r1
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 100 vlan-id 31
user@vsrx-mnha-n0# set interfaces ge-0/0/3 unit 100 family inet address 10.0.31.10/24
user@vsrx-mnha-n0# set interfaces ge-0/0/4 vlan-tagging
user@vsrx-mnha-n0# set interfaces ge-0/0/4 unit 101 description vr-right-r2
user@vsrx-mnha-n0# set interfaces ge-0/0/4 unit 101 vlan-id 33
```

```
user@vsrx-mnha-n0# set interfaces ge-0/0/4 unit 101 family inet address 10.0.33.10/24
user@vsrx-mnha-n0# set interfaces lo0 unit 0 description "Floating IP"
user@vsrx-mnha-n0# set interfaces lo0 unit 0 family inet address 172.25.0.0/32
user@vsrx-mnha-n0# set interfaces lo0 unit 1 description ICL
user@vsrx-mnha-n0# set interfaces lo0 unit 1 family inet address 172.26.0.1/32
```

f. Configure policy options.

```
[edit]
 user@vsrx-mnha-n0# set policy-options prefix-list export-int 0.0.0.0/0
 user@vsrx-mnha-n0# set policy-options prefix-list export-int 172.25.0.0/32
 user@vsrx-mnha-n0# set policy-options prefix-list export-uplink 10.0.30.0/24
 user@vsrx-mnha-n0# set policy-options prefix-list export-uplink 10.0.35.0/24
 user@vsrx-mnha-n0# set policy-options prefix-list srg1-prefix 172.25.0.0/32
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-r1 term 10 from
interface lo0.1
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-r1 term 10 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-r1 term 100 then reject
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-to-n1 term 10 from
interface lo0.1
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-to-n1 term 10 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-icl-to-n1 term 100 then
reject
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 from prefix-
list export-int
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 from
condition srg1_backup
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 then as-path-
prepend 65031
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 10 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 20 from prefix-
list export-int
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 20 from
condition srg1_active
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 20 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 90 from prefix-
list export-int
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 90 then as-path-
prepend "65031 65031"
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 90 then accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-int term 100 then reject
```

```
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 from
prefix-list export-uplink
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 from
condition srg1_backup
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 then as-
path-prepend 65031
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 10 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 20 from
prefix-list export-uplink
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 20 from
condition srg1_active
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 20 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 90 from
prefix-list export-uplink
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 90 then as-
path-prepend "65031 65031"
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 90 then
accept
 user@vsrx-mnha-n0# set policy-options policy-statement export-to-uplink term 100 then
reject
 user@vsrx-mnha-n0# set policy-options condition srg1_active if-route-exists 172.24.0.1/32
 user@vsrx-mnha-n0# set policy-options condition srg1_active if-route-exists table inet.0
 user@vsrx-mnha-n0# set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
 user@vsrx-mnha-n0# set policy-options condition srg1_backup if-route-exists table inet.0
```

g. Configure routing instances and routing option.

```
[edit]
 user@vsrx-mnha-n0# set routing-instances icl instance-type virtual-router
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.0.36.1
export export-icl-r1
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.0.36.1
peer-as 65030
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.1.39.2
export export-icl-to-n1
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp group icl neighbor 10.1.39.2
peer-as 65032
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp local-as 65031
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp bfd-liveness-detection minimum-
interval 500
```

```
 user@vsrx-mnha-n0# set routing-instances icl protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n0# set routing-instances icl interface ge-0/0/1.39
 user@vsrx-mnha-n0# set routing-instances icl interface ge-0/0/3.36
 user@vsrx-mnha-n0# set routing-instances icl interface lo0.1
 user@vsrx-mnha-n0# set routing-instances vr instance-type virtual-router
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1
export export-to-int
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1
peer-as 65030
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1
export export-to-int
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1
peer-as 65035
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.38.1 export export-to-uplink
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.38.1 peer-as 65039
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp local-as 65031
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp bfd-liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n0# set routing-instances vr protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n0# set routing-instances vr interface ge-0/0/0.102
 user@vsrx-mnha-n0# set routing-instances vr interface ge-0/0/3.100
 user@vsrx-mnha-n0# set routing-instances vr interface ge-0/0/4.101
 user@vsrx-mnha-n0# set routing-instances vr interface lo0.0
```

h. Apply configuration groups.

```
[edit]
 user@vsrx-mnha-n0# set apply-groups mnha-sync
 user@vsrx-mnha-n0# set apply-groups mnha-sync-icl
 user@vsrx-mnha-n0# set apply-groups monitor-advanced
 user@vsrx-mnha-n0# set apply-groups icd
```

i. Configure options for the peer node participating in commit synchronization.

```
[edit]
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n1 user user
```

```
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n1 authentication "$ABC123"
user@vsrx-mnha-n0# set system services netconf ssh
user@vsrx-mnha-n0# set system static-host-mapping vsrx-mnha-n1 inet 172.26.0.2
```

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

This configuration enables the node to take the configuration commands entered under the sync group and push them to the other node, using the IP address and credentials defined. You must repeat this configuration on the node-01, with changed IP address and hostnames.

3. Configure node-specific statements on the backup node (SRX-02).

   a. Configure groups for enabling peer sync through ICL.

```
[edit]
user@vsrx-mnha-n1# set groups mnha-sync-icl system commit peers vsrx-mnha-n0 routing-
instance icl
user@vsrx-mnha-n1# set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n0 inet
172.26.0.1
user@vsrx-mnha-n1# set groups icd chassis high-availability local-id local-forwarding-ip
172.26.0.12
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
172.26.0.11
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
interface lo0.1
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
liveness-detection minimum-interval 1000
user@vsrx-mnha-n1# set groups icd chassis high-availability peer-id 1 peer-forwarding-ip
liveness-detection multiplier 5
user@vsrx-mnha-n1# set groups icd interfaces lo0 unit 1 family inet address 172.26.0.12/32
```

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

b. Configure Multinode High Availability options.

```
[edit]
user@vsrx-mnha-n1# set chassis high-availability local-id 2
 user@vsrx-mnha-n1# set chassis high-availability local-id local-ip 172.26.0.2
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 peer-ip 172.26.0.1
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 interface lo0.1
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 routing-instance icl
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 vpn-profile icl
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n1# set chassis high-availability peer-id 1 liveness-detection multiplier 3
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 0 peer-id 1
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 deployment-
type routing
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 peer-id 1
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip 10.0.30.1
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip src-ip 172.25.0.0
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
probe dest-ip routing-instance vr
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 active-
signal-route 172.24.0.1
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 backup-
signal-route 172.24.0.0
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 prefix-list
srg1-prefix routing-instance vr
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 managed-
services ipsec
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 process-
packet-on-backup
 user@vsrx-mnha-n1# set chassis high-availability services-redundancy-group 1 activeness-
priority 200
```

c. Configure IPsec VPN related options.

```
[edit]
user@vsrx-mnha-n1# set security ike proposal ike-prop authentication-method pre-shared-keys
 user@vsrx-mnha-n1# set security ike proposal ike-prop dh-group group20
 user@vsrx-mnha-n1# set security ike proposal ike-prop encryption-algorithm aes-256-gcm
```

```
 user@vsrx-mnha-n1# set security ike proposal ike-prop lifetime-seconds 28800
 user@vsrx-mnha-n1# set security ike policy ike-policy proposals ike-prop
 user@vsrx-mnha-n1# set security ike policy ike-policy pre-shared-key ascii-text "$ABC13"
 user@vsrx-mnha-n1# set security ike policy icl proposals ike-prop
 user@vsrx-mnha-n1# set security ike policy icl pre-shared-key ascii-text "$ABC123"
 user@vsrx-mnha-n1# set security ike gateway icl ike-policy icl
 user@vsrx-mnha-n1# set security ike gateway icl version v2-only
 user@vsrx-mnha-n1# set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
 user@vsrx-mnha-n1# set security ipsec proposal ipsec-prop lifetime-seconds 3600
 user@vsrx-mnha-n1# set security ipsec policy ipsec-policy perfect-forward-secrecy keys
group20
 user@vsrx-mnha-n1# set security ipsec policy ipsec-policy proposals ipsec-prop
 user@vsrx-mnha-n1# set security ipsec vpn icl ha-link-encryption
 user@vsrx-mnha-n1# set security ipsec vpn icl ike gateway icl
 user@vsrx-mnha-n1# set security ipsec vpn icl ike ipsec-policy ipsec-policy
```

d. Configure security zones.

```
[edit]
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/3.37 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/3.37 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/3.37 host-
inbound-traffic protocols bfd
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ping
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ike
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services high-availability
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic system-services ssh
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces lo0.1 host-inbound-
traffic protocols bfd
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic system-services ping
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bgp
 user@vsrx-mnha-n1# set security zones security-zone icl interfaces ge-0/0/1.39 host-
inbound-traffic protocols bfd
```

e. Configure interfaces.

```
[edit]
user@vsrx-mnha-n1# set interfaces ge-0/0/0 description for-monitoring
user@vsrx-mnha-n1# set interfaces ge-0/0/0 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/0 unit 102 description vr-uplink-r2
user@vsrx-mnha-n1# set interfaces ge-0/0/0 unit 102 vlan-id 39
user@vsrx-mnha-n1# set interfaces ge-0/0/0 unit 102 family inet address 10.0.39.10/24
user@vsrx-mnha-n1# set interfaces ge-0/0/1 description br-lab-ha-1
user@vsrx-mnha-n1# set interfaces ge-0/0/1 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/1 mtu 9000
user@vsrx-mnha-n1# set interfaces ge-0/0/1 unit 39 description icl-n0
user@vsrx-mnha-n1# set interfaces ge-0/0/1 unit 39 vlan-id 39
user@vsrx-mnha-n1# set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.2/24
user@vsrx-mnha-n1# set interfaces ge-0/0/3 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 37 description icd
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 37 vlan-id 37
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 37 family inet address 10.0.37.10/24
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 100 description vr-left-r1
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 100 vlan-id 32
user@vsrx-mnha-n1# set interfaces ge-0/0/3 unit 100 family inet address 10.0.32.10/24
user@vsrx-mnha-n1# set interfaces ge-0/0/4 vlan-tagging
user@vsrx-mnha-n1# set interfaces ge-0/0/4 unit 101 description vr-right-r2
user@vsrx-mnha-n1# set interfaces ge-0/0/4 unit 101 vlan-id 34
user@vsrx-mnha-n1# set interfaces ge-0/0/4 unit 101 family inet address 10.0.34.10/24
user@vsrx-mnha-n1# set interfaces lo0 unit 0 description "Floating IP"
user@vsrx-mnha-n1# set interfaces lo0 unit 0 family inet address 172.25.0.0/32
user@vsrx-mnha-n1# set interfaces lo0 unit 1 description ICL
user@vsrx-mnha-n1# set interfaces lo0 unit 1 family inet address 172.26.0.2/32
```

f. Configure policy options.

```
[edit]
 user@vsrx-mnha-n1# set policy-options prefix-list export-int 0.0.0.0/0
 user@vsrx-mnha-n1# set policy-options prefix-list export-int 172.25.0.0/32
 user@vsrx-mnha-n1# set policy-options prefix-list export-uplink 10.0.30.0/24
 user@vsrx-mnha-n1# set policy-options prefix-list export-uplink 10.0.35.0/24
 user@vsrx-mnha-n1# set policy-options prefix-list srg1-prefix 172.25.0.0/32
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-r1 term 10 from
interface lo0.1
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-r1 term 10 then accept
```

```
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-r1 term 100 then reject
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-to-n0 term 10 from
interface lo0.1
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-to-n0 term 10 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-icl-to-n0 term 100 then
reject
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 from prefix-
list export-int
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 from
condition srg1_backup
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 then as-path-
prepend 65032
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 10 then accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 20 from prefix-
list export-int
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 20 from
condition srg1_active
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 20 then accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 90 from prefix-
list export-int
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 90 then as-path-
prepend "65032 65032 65032"
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 90 then accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-int term 100 then reject
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 from
prefix-list export-uplink
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 from
condition srg1_backup
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 then as-
path-prepend 65032
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 10 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 20 from
prefix-list export-uplink
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 20 from
condition srg1_active
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 20 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 90 from
prefix-list export-uplink
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 90 then as-
path-prepend "65032 65032 65032"
```

```
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 90 then
accept
 user@vsrx-mnha-n1# set policy-options policy-statement export-to-uplink term 100 then
reject
 user@vsrx-mnha-n1# set policy-options condition srg1_active if-route-exists 172.24.0.1/32
 user@vsrx-mnha-n1# set policy-options condition srg1_active if-route-exists table inet.0
 user@vsrx-mnha-n1# set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
 user@vsrx-mnha-n1# set policy-options condition srg1_backup if-route-exists table inet.0
```

g. Configure routing-instances and routing options.

```
[edit]
 user@vsrx-mnha-n1# set routing-instances icl instance-type virtual-router
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.0.37.1
export export-icl-r1
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.0.37.1
peer-as 65030
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.1.39.1
export export-icl-to-n0
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp group icl neighbor 10.1.39.1
peer-as 65031
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp local-as 65032
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp bfd-liveness-detection minimum-
interval 500
 user@vsrx-mnha-n1# set routing-instances icl protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n1# set routing-instances icl interface ge-0/0/1.39
 user@vsrx-mnha-n1# set routing-instances icl interface ge-0/0/3.37
 user@vsrx-mnha-n1# set routing-instances icl interface lo0.1
 user@vsrx-mnha-n1# set routing-instances vr instance-type virtual-router
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1
export export-to-int
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1
peer-as 65030
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1
export export-to-int
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1
peer-as 65035
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.39.1 export export-to-uplink
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp group uplink-r2 neighbor
10.0.39.1 peer-as 65039
```

```
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp local-as 65032
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp bfd-liveness-detection minimum-
interval 1000
 user@vsrx-mnha-n1# set routing-instances vr protocols bgp bfd-liveness-detection
multiplier 3
 user@vsrx-mnha-n1# set routing-instances vr interface ge-0/0/0.102
 user@vsrx-mnha-n1# set routing-instances vr interface ge-0/0/3.100
 user@vsrx-mnha-n1# set routing-instances vr interface ge-0/0/4.101
 user@vsrx-mnha-n1# set routing-instances vr interface lo0.0
```

h. Configure options for the peers participating in commit synchronization. Configure options for the peers participating in commit synchronization. This configuration enables the node to take the configuration commands entered under the sync group and push them to the other node, using the IP address and credentials defined.

```
[edit]
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n0 user user
user@vsrx-mnha-n0# set system commit peers vsrx-mnha-n1 authentication "$ABC123"
user@vsrx-mnha-n0# set system services netconf ssh
user@vsrx-mnha-n0# set system static-host-mapping vsrx-mnha-n0 inet 172.26.0.1
```

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

4. Use the following command to configure the `commit` command to automatically perform a peers-synchronize action between peers:

```
[edit]
user@host# set system commit peers-synchronize
```

The local peer (or requesting peer) on which you enable the `peers-synchronize` statement copies and loads its configuration to the remote (or responding) peer.

NOTE: Use the `set security ssh-known-hosts fetch-from-server` and `set security ssh-known hosts host`statements to include the other node as known host. When you commit the configuration, the system displays following message:

```
user@03-vsrx-mnha-n0# set security ssh-known-hosts fetch-from-server
03-vsrx-mnha-n1 The authenticity of host '03-vsrx-mnha-n1 (172.26.0.2)' can't be
established.
ECDSA key fingerprint is SHA256:abc123.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vsrx-mnha-2, 172.26.0.2' (ECDSA) to the list of known
hosts.
```

You must add the SSH key fingerprint for the Multinode High Availability peer. This step is needed for the configuration synchronization to work.

Verification

IN THIS SECTION

Use the following show commands to verify the feature in this example.

Command	Verification Task
show chassis high availability information	Displays Multinode High Availability details including status.
show chassis high-availability peer-info	Displays details such as peer node, connection details, and packet statistics of the peer node in a Multinode High Availability setup.
show chassis high-availability services-redundancy-group	Display the service redundancy group information in a Multinode High Availability setup.

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 commands on both nodes:

```
use@vsrx-mnha-n0> 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: 172.26.0.1
Local Forwarding IP: 172.26.0.11
HA Peer Information:

    Peer Id: 2        IP address: 172.26.0.2    Interface: lo0.1
    Routing Instance: icl
    Encrypted: YES

                  Conn State: UP
    Configured BFD Detection Time: 3 * 1000ms
    Cold Sync Status: COMPLETE
    Peer Forwarding IP: 172.26.0.12             Interface: lo0.1
    Peer ICD Conn State: UP

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: 100
        Preemption: DISABLED
```

```
        Process Packet In Backup State: YES
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
          Peer Id: 2
          Status : ACTIVE
          Health Status: HEALTHY
          Failover Readiness: N/A
```

```
user@vsrx-mnha-n1# 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: 172.26.0.2
Local Forwarding IP: 172.26.0.12
HA Peer Information:

    Peer Id: 1        IP address: 172.26.0.1    Interface: lo0.1
    Routing Instance: icl
    Encrypted: YES

                  Conn State: UP
    Configured BFD Detection Time: 3 * 1000ms
    Cold Sync Status: COMPLETE
    Peer Forwarding IP: 172.26.0.11             Interface: lo0.1
    Peer ICD Conn State: UP

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: 200
        Preemption: DISABLED
        Process Packet In Backup State: YES
        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
```

Meaning

Verify these details from the command output:

- Local node and peer node details such as IP address and ID.
- `Node Status: ONLINE` indicates that the node is up.
- `Conn State: UP` indicates that the ICL link is established and operational.
- `Peer ICD Conn State: UP` indicates that the ICD link is established and operational.
- `Encrypted: YES` indicates that ICL connection is encrypted.
- `Peer Information` Services Redundancy Group indicates peer node is healthy and ready for failover.

Check Multinode High Availability Peer Node Details

Purpose

View details of the peer node in the Multinode High Availability setup.

Action

From operational mode, run the following command:

```
user@vsrx-mnha-n0> show chassis high-availability peer-info

HA Peer Information:

   Peer-ID: 2        IP address: 172.26.0.2    Interface: lo0.1
   Routing Instance: icl
   Encrypted: YES    Conn State: UP
   Cold Sync Status: COMPLETE
   Peer Forwarding IP: 172.26.0.12             Interface: lo0.1
   Peer ICD Conn State: UP
   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           12              9
        SRG Status Ack            9              9
        Attribute Msg             7              4
        Attribute Ack             4              4
```

Meaning

You can get the following details from the command output:

- `Peer ID: 2` shows the ID of the other node.
- `Conn State: UP` and `Peer ICD Conn State: UP` indicate that the both ICL and ICD link are established.
- `Packet Statistics` shows packets transferred between the nodes.

Check Multinode High Availability Service Redundancy Group Details

Purpose

View and verify the details of the Multinode High Availability SRG details.

Action

From operational mode, run the following command:

SRX-01 Device

```
                user@vsrx-mnha-n0> 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: BACKUP
        Activeness Priority: 100
        Preemption: DISABLED
        Process Packet In Backup State: YES
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
        Peer Id: 2


                Status : ACTIVE
          Health Status: HEALTHY
          Failover Readiness: N/A

        Signal Route Info:
          Active Signal Route:
          IP: 172.24.0.1
          Routing Instance: default
          Status: NOT INSTALLED

          Backup Signal Route:
          IP: 172.24.0.0
```

```
          Routing Instance: default
          Status: INSTALLED

        Split-brain Prevention Probe Info:
          DST-IP: 10.0.30.1
          SRC-IP: 172.25.0.0
          Routing Instance: vr
          Type: ICMP Probe
          Status: NOT RUNNING
          Result: N/A           Reason: N/A

        SRG Path Monitor Info:
          SRG Monitor Status: UP
          SRG Monitor Threshold: 200
          SRG Monitor Weight: 0
          SRG Monitor Failed Objects: NONE

          Object Name: routers
          Object Status: UP
          Object Monitored Entries: [ BFD ]
          Object Failures: [ BFD ]
          Object Threshold: 200
          Object Current Weight: 100

          Object Name: endpoints
          Object Status: UP
          Object Monitored Entries: [ IP ]
          Object Failures: [ IP ]
          Object Threshold: 200
          Object Current Weight: 100

IP SRGID Table:
        SRGID    IP Prefix                                  Routing Table
        1        172.25.0.0/32                              vr
```

Now run the same command on SRX-02 device and notice the command output differences such as Status, Peer Information and so on.

```
user@vsrx-mnha-n1> 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: DISABLED
        Process Packet In Backup State: YES
        Control Plane State: READY
        System Integrity Check: COMPLETE
        Failure Events: NONE
        Peer Information:
          Peer Id: 1


                  Status : BACKUP
          Health Status: HEALTHY
          Failover Readiness: READY

        Signal Route Info:
          Active Signal Route:
          IP: 172.24.0.1
          Routing Instance: default
          Status: INSTALLED

          Backup Signal Route:
          IP: 172.24.0.0
          Routing Instance: default
          Status: NOT INSTALLED

        Split-brain Prevention Probe Info:
          DST-IP: 10.0.30.1
          SRC-IP: 172.25.0.0
          Routing Instance: vr
          Type: ICMP Probe
          Status: NOT RUNNING
          Result: N/A           Reason: N/A


        SRG Path Monitor Info:
```

```
            SRG Monitor Status: UP
            SRG Monitor Threshold: 200
            SRG Monitor Weight: 0
            SRG Monitor Failed Objects: NONE

            Object Name: routers
            Object Status: UP
            Object Monitored Entries: [ BFD ]
            Object Failures: [ BFD ]
            Object Threshold: 200
            Object Current Weight: 100

            Object Name: endpoints
            Object Status: UP
            Object Monitored Entries: [ IP ]
            Object Failures: [ IP ]
            Object Threshold: 200
            Object Current Weight: 100

IP SRGID Table:
        SRGID    IP Prefix                                Routing Table
        1        172.25.0.0/32                            vr
```

Meaning

Verify these details from the command output:

- `Deployment Type: ROUTING` indicates the Multinode High Availability is setup for Layer 3 (Routing) mode.
- `Status: BACKUP` indicates currently the node is operating as Backup node.
- `Peer Information` provides peer node details such as deployment type, status, and active and back up signal routes.
- The output also indicates configured monitoring options and failure events (if any).

Set Commands on All Devices

IN THIS SECTION

Device Configured as Active Node (vsrx-mnha-n0)

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

```
set groups mnha-sync when peers vsrx-mnha-n0
set groups mnha-sync when peers vsrx-mnha-n1
set groups mnha-sync security ike proposal ike-prop authentication-method pre-shared-keys
set groups mnha-sync security ike proposal ike-prop dh-group group20
set groups mnha-sync security ike proposal ike-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ike proposal ike-prop lifetime-seconds 28800
set groups mnha-sync security ike policy ike-policy proposals ike-prop
set groups mnha-sync security ike policy ike-policy pre-shared-key ascii-text "$ABc123"
set groups mnha-sync security ike policy icl proposals ike-prop
set groups mnha-sync security ike gateway r1 ike-policy ike-policy
set groups mnha-sync security ike gateway r1 address 10.0.30.1
set groups mnha-sync security ike gateway r1 dead-peer-detection probe-idle-tunnel
set groups mnha-sync security ike gateway r1 dead-peer-detection interval 5
set groups mnha-sync security ike gateway r1 dead-peer-detection threshold 5
set groups mnha-sync security ike gateway r1 external-interface lo0.0
set groups mnha-sync security ike gateway r1 version v2-only
set groups mnha-sync security ike gateway icl ike-policy icl
set groups mnha-sync security ike gateway icl version v2-only
set groups mnha-sync security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ipsec proposal ipsec-prop lifetime-seconds 3600
set groups mnha-sync security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
set groups mnha-sync security ipsec policy ipsec-policy proposals ipsec-prop
set groups mnha-sync security ipsec vpn r1 bind-interface st0.0
set groups mnha-sync security ipsec vpn r1 ike gateway r1
set groups mnha-sync security ipsec vpn r1 ike ipsec-policy ipsec-policy
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 local-ip 10.0.35.11/32
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 remote-ip 10.0.30.11/32
```

```
set groups mnha-sync security ipsec vpn r1 establish-tunnels immediately
set groups mnha-sync security ipsec vpn icl ha-link-encryption
set groups mnha-sync security ipsec vpn icl ike gateway icl
set groups mnha-sync security ipsec vpn icl ike ipsec-policy ipsec-policy
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match source-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match destination-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match application
any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit then permit
set groups mnha-sync security policies global policy internal match source-address any
set groups mnha-sync security policies global policy internal match destination-address any
set groups mnha-sync security policies global policy internal match application any
set groups mnha-sync security policies global policy internal match from-zone right
set groups mnha-sync security policies global policy internal match from-zone vpn
set groups mnha-sync security policies global policy internal match from-zone left
set groups mnha-sync security policies global policy internal match to-zone left
set groups mnha-sync security policies global policy internal match to-zone right
set groups mnha-sync security policies global policy internal match to-zone vpn
set groups mnha-sync security policies global policy internal then permit
set groups mnha-sync security policies global policy internal then log session-close
set groups mnha-sync security policies global policy untrust match source-address any
set groups mnha-sync security policies global policy untrust match destination-address any
set groups mnha-sync security policies global policy untrust match application any
set groups mnha-sync security policies global policy untrust match from-zone left
set groups mnha-sync security policies global policy untrust match from-zone right
set groups mnha-sync security policies global policy untrust match to-zone untrust
set groups mnha-sync security policies global policy untrust then permit
set groups mnha-sync security zones security-zone vpn interfaces st0.0
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ike
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic protocols bgp
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
```

```
traffic protocols bgp
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bgp
set groups mnha-sync interfaces st0 unit 0 family inet
set groups mnha-sync-icl system commit peers vsrx-mnha-n1 routing-instance icl
set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n1 inet 172.26.0.2
set groups icd chassis high-availability local-id local-forwarding-ip 172.26.0.11
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip 172.26.0.12
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip interface lo0.1
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection minimum-
interval 1000
set groups icd chassis high-availability peer-id 2 peer-forwarding-ip liveness-detection
multiplier 5
set groups icd interfaces lo0 unit 1 family inet address 172.26.0.11/32
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 src-ip 10.0.31.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.31.1 interface ge-0/0/0.100
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 src-ip 10.0.33.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.33.1 interface ge-0/0/0.101
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 src-ip 10.0.38.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.38.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
```

```
liveliness 10.0.38.1 interface ge-0/0/0.102
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor
interface ge-0/0/0
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 src-ip 10.0.31.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 interface ge-0/0/3.100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.31.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 src-ip 10.0.33.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 interface ge-0/0/4.101
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.33.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 src-ip 10.0.38.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 routing-instance vr
```

```
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 interface ge-0/0/0.102
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.38.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor srg-
threshold 200
set apply-groups mnha-sync
set apply-groups mnha-sync-icl
set apply-groups monitor-advanced
set apply-groups icd
set system commit peers vsrx-mnha-n1 user user
set system commit peers vsrx-mnha-n1 authentication "$ABC123"
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 172.26.0.1
set chassis high-availability peer-id 2 peer-ip 172.26.0.2
set chassis high-availability peer-id 2 interface lo0.1
set chassis high-availability peer-id 2 routing-instance icl
set chassis high-availability peer-id 2 vpn-profile icl
set chassis high-availability peer-id 2 liveness-detection minimum-interval 1000
set chassis high-availability peer-id 2 liveness-detection multiplier 3
set chassis high-availability services-redundancy-group 0 peer-id 2
set chassis high-availability services-redundancy-group 1 deployment-type routing
set chassis high-availability services-redundancy-group 1 peer-id 2
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip 10.0.30.1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip src-ip
172.25.0.0
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip routing-
instance vr
set chassis high-availability services-redundancy-group 1 active-signal-route 172.24.0.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 172.24.0.0
set chassis high-availability services-redundancy-group 1 prefix-list srg1-prefix routing-
instance vr
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 process-packet-on-backup
set chassis high-availability services-redundancy-group 1 activeness-priority 100
set security ike proposal ike-prop authentication-method pre-shared-keys
set security ike proposal ike-prop dh-group group20
set security ike proposal ike-prop encryption-algorithm aes-256-gcm
set security ike proposal ike-prop lifetime-seconds 28800
set security ike policy ike-policy proposals ike-prop
set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
```

```
set security ike policy icl proposals ike-prop
set security ike policy icl pre-shared-key ascii-text "$ABC123."
set security ike gateway icl ike-policy icl
set security ike gateway icl version v2-only
set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set security ipsec proposal ipsec-prop lifetime-seconds 3600
set security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
set security ipsec policy ipsec-policy proposals ipsec-prop
set security ipsec vpn icl ha-link-encryption
set security ipsec vpn icl ike gateway icl
set security ipsec vpn icl ike ipsec-policy ipsec-policy
set security zones security-zone icl interfaces ge-0/0/3.36 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/3.36 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/3.36 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ping
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ike
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services high-
availability
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ssh
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bfd
set interfaces ge-0/0/0 description for-monitoring
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 102 description vr-uplink-r2
set interfaces ge-0/0/0 unit 102 vlan-id 38
set interfaces ge-0/0/0 unit 102 family inet address 10.0.38.10/24
set interfaces ge-0/0/1 description br-lab-ha-1
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 mtu 9000
set interfaces ge-0/0/1 unit 39 description icl-n1
set interfaces ge-0/0/1 unit 39 vlan-id 39
set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.1/24
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 36 description icl-r1
set interfaces ge-0/0/3 unit 36 vlan-id 36
set interfaces ge-0/0/3 unit 36 family inet address 10.0.36.10/24
set interfaces ge-0/0/3 unit 100 description vr-left-r1
set interfaces ge-0/0/3 unit 100 vlan-id 31
set interfaces ge-0/0/3 unit 100 family inet address 10.0.31.10/24
```

```
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 101 description vr-right-r2
set interfaces ge-0/0/4 unit 101 vlan-id 33
set interfaces ge-0/0/4 unit 101 family inet address 10.0.33.10/24
set interfaces lo0 unit 0 description "Floating IP"
set interfaces lo0 unit 0 family inet address 172.25.0.0/32
set interfaces lo0 unit 1 description ICL
set interfaces lo0 unit 1 family inet address 172.26.0.1/32
set policy-options prefix-list export-int 0.0.0.0/0
set policy-options prefix-list export-int 172.25.0.0/32
set policy-options prefix-list export-uplink 10.0.30.0/24
set policy-options prefix-list export-uplink 10.0.35.0/24
set policy-options prefix-list srg1-prefix 172.25.0.0/32
set policy-options policy-statement export-icl-r1 term 10 from interface lo0.1
set policy-options policy-statement export-icl-r1 term 10 then accept
set policy-options policy-statement export-icl-r1 term 100 then reject
set policy-options policy-statement export-icl-to-n1 term 10 from interface lo0.1
set policy-options policy-statement export-icl-to-n1 term 10 then accept
set policy-options policy-statement export-icl-to-n1 term 100 then reject
set policy-options policy-statement export-to-int term 10 from prefix-list export-int
set policy-options policy-statement export-to-int term 10 from condition srg1_backup
set policy-options policy-statement export-to-int term 10 then as-path-prepend 65031
set policy-options policy-statement export-to-int term 10 then accept
set policy-options policy-statement export-to-int term 20 from prefix-list export-int
set policy-options policy-statement export-to-int term 20 from condition srg1_active
set policy-options policy-statement export-to-int term 20 then accept
set policy-options policy-statement export-to-int term 90 from prefix-list export-int
set policy-options policy-statement export-to-int term 90 then as-path-prepend "65031 65031"
set policy-options policy-statement export-to-int term 90 then accept
set policy-options policy-statement export-to-int term 100 then reject
set policy-options policy-statement export-to-uplink term 10 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 10 from condition srg1_backup
set policy-options policy-statement export-to-uplink term 10 then as-path-prepend 65031
set policy-options policy-statement export-to-uplink term 10 then accept
set policy-options policy-statement export-to-uplink term 20 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 20 from condition srg1_active
set policy-options policy-statement export-to-uplink term 20 then accept
set policy-options policy-statement export-to-uplink term 90 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 90 then as-path-prepend "65031 65031"
set policy-options policy-statement export-to-uplink term 90 then accept
set policy-options policy-statement export-to-uplink term 100 then reject
set policy-options condition srg1_active if-route-exists 172.24.0.1/32
set policy-options condition srg1_active if-route-exists table inet.0
```

```
set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
set policy-options condition srg1_backup if-route-exists table inet.0
set routing-instances icl instance-type virtual-router
set routing-instances icl protocols bgp group icl neighbor 10.0.36.1 export export-icl-r1
set routing-instances icl protocols bgp group icl neighbor 10.0.36.1 peer-as 65030
set routing-instances icl protocols bgp group icl neighbor 10.1.39.2 export export-icl-to-n1
set routing-instances icl protocols bgp group icl neighbor 10.1.39.2 peer-as 65032
set routing-instances icl protocols bgp local-as 65031
set routing-instances icl protocols bgp bfd-liveness-detection minimum-interval 500
set routing-instances icl protocols bgp bfd-liveness-detection multiplier 3
set routing-instances icl interface ge-0/0/1.39
set routing-instances icl interface ge-0/0/3.36
set routing-instances icl interface lo0.1
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1 export export-to-int
set routing-instances vr protocols bgp group r1 neighbor 10.0.31.1 peer-as 65030
set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1 export export-to-int
set routing-instances vr protocols bgp group r2 neighbor 10.0.33.1 peer-as 65035
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.38.1 export export-to-uplink
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.38.1 peer-as 65039
set routing-instances vr protocols bgp local-as 65031
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.102
set routing-instances vr interface ge-0/0/3.100
set routing-instances vr interface ge-0/0/4.101
set routing-instances vr interface lo0.0
```

Device Configured as Backup Node (SRX-02)

NOTE: The device names used in this example are vsrx-mnha-n0 and vsrx-mnha-n1. Ensure that you use the host name of your device for this configuration.

```
set groups mnha-sync-icl system commit peers vsrx-mnha-n0 routing-instance icl
set groups mnha-sync-icl system static-host-mapping vsrx-mnha-n0 inet 172.26.0.1
set groups mnha-sync when peers vsrx-mnha-n0
set groups mnha-sync when peers vsrx-mnha-n1
set groups mnha-sync security ike proposal ike-prop authentication-method pre-shared-keys
set groups mnha-sync security ike proposal ike-prop dh-group group20
```

```
set groups mnha-sync security ike proposal ike-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ike proposal ike-prop lifetime-seconds 28800
set groups mnha-sync security ike policy ike-policy proposals ike-prop
set groups mnha-sync security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
set groups mnha-sync security ike policy icl proposals ike-prop
set groups mnha-sync security ike gateway r1 ike-policy ike-policy
set groups mnha-sync security ike gateway r1 address 10.0.30.1
set groups mnha-sync security ike gateway r1 dead-peer-detection probe-idle-tunnel
set groups mnha-sync security ike gateway r1 dead-peer-detection interval 5
set groups mnha-sync security ike gateway r1 dead-peer-detection threshold 5
set groups mnha-sync security ike gateway r1 external-interface lo0.0
set groups mnha-sync security ike gateway r1 version v2-only
set groups mnha-sync security ike gateway icl ike-policy icl
set groups mnha-sync security ike gateway icl version v2-only
set groups mnha-sync security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set groups mnha-sync security ipsec proposal ipsec-prop lifetime-seconds 3600
set groups mnha-sync security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
set groups mnha-sync security ipsec policy ipsec-policy proposals ipsec-prop
set groups mnha-sync security ipsec vpn r1 bind-interface st0.0
set groups mnha-sync security ipsec vpn r1 ike gateway r1
set groups mnha-sync security ipsec vpn r1 ike ipsec-policy ipsec-policy
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 local-ip 10.0.35.11/32
set groups mnha-sync security ipsec vpn r1 traffic-selector ts1 remote-ip 10.0.30.11/32
set groups mnha-sync security ipsec vpn r1 establish-tunnels immediately
set groups mnha-sync security ipsec vpn icl ha-link-encryption
set groups mnha-sync security ipsec vpn icl ike gateway icl
set groups mnha-sync security ipsec vpn icl ike ipsec-policy ipsec-policy
set groups mnha-sync security flow tcp-mss ipsec-vpn mss 1400
set groups mnha-sync security flow tcp-session strict-syn-check
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match source-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match destination-
address any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit match application
any
set groups mnha-sync security policies from-zone icl to-zone icl policy permit then permit
set groups mnha-sync security policies global policy internal match source-address any
set groups mnha-sync security policies global policy internal match destination-address any
set groups mnha-sync security policies global policy internal match application any
set groups mnha-sync security policies global policy internal match from-zone right
set groups mnha-sync security policies global policy internal match from-zone vpn
set groups mnha-sync security policies global policy internal match from-zone left
set groups mnha-sync security policies global policy internal match to-zone left
```

```
set groups mnha-sync security policies global policy internal match to-zone right
set groups mnha-sync security policies global policy internal match to-zone vpn
set groups mnha-sync security policies global policy internal then permit
set groups mnha-sync security policies global policy internal then log session-close
set groups mnha-sync security policies global policy untrust match source-address any
set groups mnha-sync security policies global policy untrust match destination-address any
set groups mnha-sync security policies global policy untrust match application any
set groups mnha-sync security policies global policy untrust match from-zone left
set groups mnha-sync security policies global policy untrust match from-zone right
set groups mnha-sync security policies global policy untrust match to-zone untrust
set groups mnha-sync security policies global policy untrust then permit
set groups mnha-sync security zones security-zone vpn interfaces st0.0
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ike
set groups mnha-sync security zones security-zone left interfaces lo0.0 host-inbound-traffic
system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic protocols bgp
set groups mnha-sync security zones security-zone left interfaces ge-0/0/3.100 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic protocols bgp
set groups mnha-sync security zones security-zone right interfaces ge-0/0/4.101 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic system-services ping
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bfd
set groups mnha-sync security zones security-zone untrust interfaces ge-0/0/0.102 host-inbound-
traffic protocols bgp
set groups mnha-sync interfaces st0 unit 0 family inet
set groups icd chassis high-availability local-id local-forwarding-ip 172.26.0.12
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip 172.26.0.11
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip interface lo0.1
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection minimum-
interval 1000
set groups icd chassis high-availability peer-id 1 peer-forwarding-ip liveness-detection
multiplier 5
set groups icd interfaces lo0 unit 1 family inet address 172.26.0.12/32
```

```
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 src-ip 10.0.32.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.32.1 interface ge-0/0/3.100
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 src-ip 10.0.34.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.34.1 interface ge-0/0/4.101
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 src-ip 10.0.39.10
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 routing-instance vr
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 session-type singlehop
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor bfd-
liveliness 10.0.39.1 interface ge-0/0/0.102
set groups monitor-simple chassis high-availability services-redundancy-group 1 monitor
interface ge-0/0/0
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.30.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object endpoints ip destination-ip 10.0.35.10 weight 50
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers object-threshold 200
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness threshold 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
```

```
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 src-ip 10.0.32.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 interface ge-0/0/3.100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.32.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 src-ip 10.0.34.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 interface ge-0/0/4.101
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.34.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 src-ip 10.0.39.10
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 routing-instance vr
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 session-type singlehop
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 interface ge-0/0/0.102
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor
monitor-object routers bfd-liveliness destination-ip 10.0.39.1 weight 100
set groups monitor-advanced chassis high-availability services-redundancy-group 1 monitor srg-
threshold 200
set apply-groups mnha-sync
set apply-groups mnha-sync-icl
set apply-groups monitor-advanced
set apply-groups icd
set system commit peers vsrx-mnha-n0 user user
set system commit peers vsrx-mnha-n0 authentication "$ABC123"
set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 172.26.0.2
set chassis high-availability peer-id 1 peer-ip 172.26.0.1
set chassis high-availability peer-id 1 interface lo0.1
set chassis high-availability peer-id 1 routing-instance icl
set chassis high-availability peer-id 1 vpn-profile icl
```

```
set chassis high-availability peer-id 1 liveness-detection minimum-interval 1000
set chassis high-availability peer-id 1 liveness-detection multiplier 3
set chassis high-availability services-redundancy-group 0 peer-id 1
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 10.0.30.1
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip src-ip
172.25.0.0
set chassis high-availability services-redundancy-group 1 activeness-probe dest-ip routing-
instance vr
set chassis high-availability services-redundancy-group 1 active-signal-route 172.24.0.1
set chassis high-availability services-redundancy-group 1 backup-signal-route 172.24.0.0
set chassis high-availability services-redundancy-group 1 prefix-list srg1-prefix routing-
instance vr
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 process-packet-on-backup
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set security ike proposal ike-prop authentication-method pre-shared-keys
set security ike proposal ike-prop dh-group group20
set security ike proposal ike-prop encryption-algorithm aes-256-gcm
set security ike proposal ike-prop lifetime-seconds 28800
set security ike policy ike-policy proposals ike-prop
set security ike policy ike-policy pre-shared-key ascii-text "$ABC123"
set security ike policy icl proposals ike-prop
set security ike policy icl pre-shared-key ascii-text "$ABC123"
set security ike gateway icl ike-policy icl
set security ike gateway icl version v2-only
set security ipsec proposal ipsec-prop encryption-algorithm aes-256-gcm
set security ipsec proposal ipsec-prop lifetime-seconds 3600
set security ipsec policy ipsec-policy perfect-forward-secrecy keys group20
set security ipsec policy ipsec-policy proposals ipsec-prop
set security ipsec vpn icl ha-link-encryption
set security ipsec vpn icl ike gateway icl
set security ipsec vpn icl ike ipsec-policy ipsec-policy
set security zones security-zone icl interfaces ge-0/0/3.37 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/3.37 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/3.37 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ping
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ike
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services high-
availability
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic system-services ssh
```

```
set security zones security-zone icl interfaces lo0.1 host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic system-services
ping
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bgp
set security zones security-zone icl interfaces ge-0/0/1.39 host-inbound-traffic protocols bfd
set interfaces ge-0/0/0 description for-monitoring
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 102 description vr-uplink-r2
set interfaces ge-0/0/0 unit 102 vlan-id 39
set interfaces ge-0/0/0 unit 102 family inet address 10.0.39.10/24
set interfaces ge-0/0/1 description br-lab-ha-1
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 mtu 9000
set interfaces ge-0/0/1 unit 39 description icl-n0
set interfaces ge-0/0/1 unit 39 vlan-id 39
set interfaces ge-0/0/1 unit 39 family inet address 10.1.39.2/24
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 37 description icl-r1
set interfaces ge-0/0/3 unit 37 vlan-id 37
set interfaces ge-0/0/3 unit 37 family inet address 10.0.37.10/24
set interfaces ge-0/0/3 unit 100 description vr-left-r1
set interfaces ge-0/0/3 unit 100 vlan-id 32
set interfaces ge-0/0/3 unit 100 family inet address 10.0.32.10/24
set interfaces ge-0/0/4 vlan-tagging
set interfaces ge-0/0/4 unit 101 description vr-right-r2
set interfaces ge-0/0/4 unit 101 vlan-id 34
set interfaces ge-0/0/4 unit 101 family inet address 10.0.34.10/24
set interfaces lo0 unit 0 description "Floating IP"
set interfaces lo0 unit 0 family inet address 172.25.0.0/32
set interfaces lo0 unit 1 description ICL
set interfaces lo0 unit 1 family inet address 172.26.0.2/32
set policy-options prefix-list export-int 0.0.0.0/0
set policy-options prefix-list export-int 172.25.0.0/32
set policy-options prefix-list export-uplink 10.0.30.0/24
set policy-options prefix-list export-uplink 10.0.35.0/24
set policy-options prefix-list srg1-prefix 172.25.0.0/32
set policy-options policy-statement export-icl-r1 term 10 from interface lo0.1
set policy-options policy-statement export-icl-r1 term 10 then accept
set policy-options policy-statement export-icl-r1 term 100 then reject
set policy-options policy-statement export-icl-to-n0 term 10 from interface lo0.1
set policy-options policy-statement export-icl-to-n0 term 10 then accept
set policy-options policy-statement export-icl-to-n0 term 100 then reject
set policy-options policy-statement export-to-int term 10 from prefix-list export-int
```

```
set policy-options policy-statement export-to-int term 10 from condition srg1_backup
set policy-options policy-statement export-to-int term 10 then as-path-prepend 65032
set policy-options policy-statement export-to-int term 10 then accept
set policy-options policy-statement export-to-int term 20 from prefix-list export-int
set policy-options policy-statement export-to-int term 20 from condition srg1_active
set policy-options policy-statement export-to-int term 20 then accept
set policy-options policy-statement export-to-int term 90 from prefix-list export-int
set policy-options policy-statement export-to-int term 90 then as-path-prepend "65032 65032
65032"
set policy-options policy-statement export-to-int term 90 then accept
set policy-options policy-statement export-to-int term 100 then reject
set policy-options policy-statement export-to-uplink term 10 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 10 from condition srg1_backup
set policy-options policy-statement export-to-uplink term 10 then as-path-prepend 65032
set policy-options policy-statement export-to-uplink term 10 then accept
set policy-options policy-statement export-to-uplink term 20 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 20 from condition srg1_active
set policy-options policy-statement export-to-uplink term 20 then accept
set policy-options policy-statement export-to-uplink term 90 from prefix-list export-uplink
set policy-options policy-statement export-to-uplink term 90 then as-path-prepend "65032 65032
65032"
set policy-options policy-statement export-to-uplink term 90 then accept
set policy-options policy-statement export-to-uplink term 100 then reject
set policy-options condition srg1_active if-route-exists 172.24.0.1/32
set policy-options condition srg1_active if-route-exists table inet.0
set policy-options condition srg1_backup if-route-exists 172.24.0.0/32
set policy-options condition srg1_backup if-route-exists table inet.0
set routing-instances icl instance-type virtual-router
set routing-instances icl protocols bgp group icl neighbor 10.0.37.1 export export-icl-r1
set routing-instances icl protocols bgp group icl neighbor 10.0.37.1 peer-as 65030
set routing-instances icl protocols bgp group icl neighbor 10.1.39.1 export export-icl-to-n0
set routing-instances icl protocols bgp group icl neighbor 10.1.39.1 peer-as 65031
set routing-instances icl protocols bgp local-as 65032
set routing-instances icl protocols bgp bfd-liveness-detection minimum-interval 500
set routing-instances icl protocols bgp bfd-liveness-detection multiplier 3
set routing-instances icl interface ge-0/0/1.39
set routing-instances icl interface ge-0/0/3.37
set routing-instances icl interface lo0.1
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1 export export-to-int
set routing-instances vr protocols bgp group r1 neighbor 10.0.32.1 peer-as 65030
set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1 export export-to-int
set routing-instances vr protocols bgp group r2 neighbor 10.0.34.1 peer-as 65035
```

```
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.39.1 export export-to-uplink
set routing-instances vr protocols bgp group uplink-r2 neighbor 10.0.39.1 peer-as 65039
set routing-instances vr protocols bgp local-as 65032
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.102
set routing-instances vr interface ge-0/0/3.100
set routing-instances vr interface ge-0/0/4.101
set routing-instances vr interface lo0.0
```

Router 1 (Device Configured as Router)

```
set security policies default-policy permit-all
set security zones security-zone left host-inbound-traffic system-services ping
set security zones security-zone left host-inbound-traffic system-services ike
set security zones security-zone left host-inbound-traffic protocols bgp
set security zones security-zone left host-inbound-traffic protocols bfd
set security zones security-zone left interfaces ge-0/0/2.30
set security zones security-zone left interfaces ge-0/0/0.31
set security zones security-zone left interfaces ge-0/0/1.32
set security zones security-zone left interfaces st0.0
set security zones security-zone left enable-reverse-reroute
set security zones security-zone icl host-inbound-traffic system-services ping
set security zones security-zone icl host-inbound-traffic protocols bgp
set security zones security-zone icl host-inbound-traffic protocols bfd
set security zones security-zone icl interfaces ge-0/0/0.36
set security zones security-zone icl interfaces ge-0/0/1.37
set interfaces ge-0/0/0 description br-lab-1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 31 description vr-mnha-n0
set interfaces ge-0/0/0 unit 31 vlan-id 31
set interfaces ge-0/0/0 unit 31 family inet address 10.0.31.1/24
set interfaces ge-0/0/0 unit 36 description icl-n0
set interfaces ge-0/0/0 unit 36 vlan-id 36
set interfaces ge-0/0/0 unit 36 family inet address 10.0.36.1/24
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 32 description vr-mnha-n1
set interfaces ge-0/0/1 unit 32 vlan-id 32
set interfaces ge-0/0/1 unit 32 family inet address 10.0.32.1/24
set interfaces ge-0/0/1 unit 37 description icl-n1
set interfaces ge-0/0/1 unit 37 vlan-id 37
```

```
set interfaces ge-0/0/1 unit 37 family inet address 10.0.37.1/24
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 30 description vr-linux-1
set interfaces ge-0/0/2 unit 30 vlan-id 30
set interfaces ge-0/0/2 unit 30 family inet address 10.0.30.1/24
set interfaces st0 unit 0 family inet
set policy-options policy-statement export-icl-n0 term 10 from interface ge-0/0/1.37
set policy-options policy-statement export-icl-n0 term 10 then accept
set policy-options policy-statement export-icl-n0 term 100 then reject
set policy-options policy-statement export-icl-n1 term 10 from interface ge-0/0/0.36
set policy-options policy-statement export-icl-n1 term 10 then accept
set policy-options policy-statement export-icl-n1 term 100 then reject
set policy-options policy-statement export-to-mnha-fws term 10 from interface ge-0/0/2.30
set policy-options policy-statement export-to-mnha-fws term 10 then accept
set policy-options policy-statement export-to-mnha-fws term 100 then reject
set routing-instances icl instance-type virtual-router
set routing-instances icl protocols bgp group icl local-as 65030
set routing-instances icl protocols bgp group icl bfd-liveness-detection minimum-interval 500
set routing-instances icl protocols bgp group icl bfd-liveness-detection multiplier 3
set routing-instances icl protocols bgp group icl neighbor 10.0.36.10 export export-icl-n0
set routing-instances icl protocols bgp group icl neighbor 10.0.36.10 peer-as 65031
set routing-instances icl protocols bgp group icl neighbor 10.0.37.10 export export-icl-n1
set routing-instances icl protocols bgp group icl neighbor 10.0.37.10 peer-as 65032
set routing-instances icl interface ge-0/0/0.36
set routing-instances icl interface ge-0/0/1.37
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp group mnha-n0 neighbor 10.0.31.10 peer-as 65031
set routing-instances vr protocols bgp group mnha-n1 neighbor 10.0.32.10 peer-as 65032
set routing-instances vr protocols bgp export export-to-mnha-fws
set routing-instances vr protocols bgp local-as 65030
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.31
set routing-instances vr interface ge-0/0/1.32
set routing-instances vr interface ge-0/0/2.30
set routing-instances vr interface st0.0
```

Router 2 (Device Configured as Router)

```
set security policies default-policy permit-all
set security zones security-zone right host-inbound-traffic system-services ping
```

```
set security zones security-zone right host-inbound-traffic protocols bgp
set security zones security-zone right host-inbound-traffic protocols bfd
set security zones security-zone right interfaces ge-0/0/0.33
set security zones security-zone right interfaces ge-0/0/1.34
set security zones security-zone right interfaces ge-0/0/2.35
set security zones security-zone right enable-reverse-reroute
set security zones security-zone trust tcp-rst
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust host-inbound-traffic protocols bgp
set security zones security-zone trust host-inbound-traffic protocols bfd
set security zones security-zone trust interfaces ge-0/0/0.39
set security zones security-zone trust interfaces ge-0/0/0.38
set interfaces ge-0/0/0 description br-lab-1
set interfaces ge-0/0/0 vlan-tagging
set interfaces ge-0/0/0 unit 33 description vr-mnha-n0
set interfaces ge-0/0/0 unit 33 vlan-id 33
set interfaces ge-0/0/0 unit 33 family inet address 10.0.33.1/24
set interfaces ge-0/0/0 unit 38 description uplink-mnha-n0
set interfaces ge-0/0/0 unit 38 vlan-id 38
set interfaces ge-0/0/0 unit 38 family inet address 10.0.38.1/24
set interfaces ge-0/0/0 unit 39 description uplink-mnha-n1
set interfaces ge-0/0/0 unit 39 vlan-id 39
set interfaces ge-0/0/0 unit 39 family inet address 10.0.39.1/24
set interfaces ge-0/0/1 description br-poc-mgmt
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 unit 34 description vr-mnha-n1
set interfaces ge-0/0/1 unit 34 vlan-id 34
set interfaces ge-0/0/1 unit 34 family inet address 10.0.34.1/24
set interfaces ge-0/0/2 vlan-tagging
set interfaces ge-0/0/2 unit 35 description vr-linux-2
set interfaces ge-0/0/2 unit 35 vlan-id 35
set interfaces ge-0/0/2 unit 35 family inet address 10.0.35.1/24
set policy-options policy-statement export-default term 10 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement export-default term 10 then accept
set policy-options policy-statement export-default term 100 then reject
set policy-options policy-statement export-to-mnha-fws term 10 from interface ge-0/0/0.35
set policy-options policy-statement export-to-mnha-fws term 10 then accept
set policy-options policy-statement export-to-mnha-fws term 100 then reject
set policy-options policy-statement import-from-n1 from neighbor 10.0.34.10
set policy-options policy-statement import-from-n1 then local-preference 1000
set routing-instances uplink instance-type virtual-router
set routing-instances uplink routing-options static route 0.0.0.0/0 next-hop 172.30.192.1
set routing-instances uplink protocols bgp family inet unicast loops 1
```

```
set routing-instances uplink protocols bgp group trust export export-default
set routing-instances uplink protocols bgp group trust local-as 65039
set routing-instances uplink protocols bgp group trust bfd-liveness-detection minimum-interval
1000
set routing-instances uplink protocols bgp group trust bfd-liveness-detection multiplier 3
set routing-instances uplink protocols bgp group trust neighbor 10.0.38.10 peer-as 65031
set routing-instances uplink protocols bgp group trust neighbor 10.0.39.10 peer-as 65032
set routing-instances uplink interface ge-0/0/0.38
set routing-instances uplink interface ge-0/0/0.39
set routing-instances uplink interface ge-0/0/1.0
deactivate routing-instances uplink interface ge-0/0/1.0
set routing-instances vr instance-type virtual-router
set routing-instances vr protocols bgp family inet unicast loops 1
set routing-instances vr protocols bgp group mnha-n0 neighbor 10.0.33.10 peer-as 65031
set routing-instances vr protocols bgp group mnha-n1 neighbor 10.0.34.10 import import-from-n1
set routing-instances vr protocols bgp group mnha-n1 neighbor 10.0.34.10 peer-as 65032
set routing-instances vr protocols bgp export export-to-mnha-fws
set routing-instances vr protocols bgp local-as 65035
set routing-instances vr protocols bgp bfd-liveness-detection minimum-interval 1000
set routing-instances vr protocols bgp bfd-liveness-detection multiplier 3
set routing-instances vr interface ge-0/0/0.33
set routing-instances vr interface ge-0/0/1.34
set routing-instances vr interface ge-0/0/2.35
```

Show Configuration Output

IN THIS SECTION

From configuration mode, confirm your configuration by entering the `show high availability`, `show groups`, and other details. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

SRX-01 (Active Node)

```
[edit]
user@vsrx-mnha-n0# show chassis high-availability
local-id {
    1;
    local-ip 172.26.0.1;
}
peer-id 2 {
    peer-ip 172.26.0.2;
    interface lo0.1;
    routing-instance icl;
    vpn-profile icl;
    liveness-detection {
        minimum-interval 1000;
        multiplier 3;
    }
}
services-redundancy-group 0 {
    peer-id {
        2;
    }
}
services-redundancy-group 1 {
    deployment-type routing;
    peer-id {
        2;
    }
    activeness-probe {
        dest-ip {
            10.0.30.1;
            src-ip 172.25.0.0;
            routing-instance vr;
        }
    }
    active-signal-route {
        172.24.0.1;
    }
    backup-signal-route {
        172.24.0.0;
    }
    prefix-list srg1-prefix {
```

```
        routing-instance vr;
    }
    managed-services ipsec;
    process-packet-on-backup;
    activeness-priority 100;
}
```

```
[edit]
user@vsrx-mnha-n0# show groups mnha-sync
when {
    peers [ vsrx-mnha-n0 vsrx-mnha-n1 ];
}
security {
    ike {
        proposal ike-prop {
            authentication-method pre-shared-keys;
            dh-group group20;
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 28800;
        }
        policy ike-policy {
            proposals ike-prop;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        policy icl {
            proposals ike-prop;
        }
        gateway r1 {
            ike-policy ike-policy;
            address 10.0.30.1;
            dead-peer-detection {
                probe-idle-tunnel;
                interval 5;
                threshold 5;
            }
            external-interface lo0.0;
            version v2-only;
        }
        gateway icl {
            ike-policy icl;
            version v2-only;
```

```
        }
    }
    ipsec {
        proposal ipsec-prop {
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 3600;
        }
        policy ipsec-policy {
            perfect-forward-secrecy {
                keys group20;
            }
            proposals ipsec-prop;
        }
        vpn r1 {
            bind-interface st0.0;
            ike {
                gateway r1;
                ipsec-policy ipsec-policy;
            }
            traffic-selector ts1 {
                local-ip 10.0.35.11/32;
                remote-ip 10.0.30.11/32;
            }
            establish-tunnels immediately;
        }
        vpn icl {
            ha-link-encryption;
            ike {
                gateway icl;
                ipsec-policy ipsec-policy;
            }
        }
    }
    policies {
        from-zone icl to-zone icl {
            policy permit {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
```

```
                }
            }
        }
        global {
            policy internal {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                    from-zone [ right vpn left ];
                    to-zone [ left right vpn ];
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
            }
            policy untrust {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                    from-zone [ left right ];
                    to-zone untrust;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone vpn {
            interfaces {
                st0.0;
            }
        }
        security-zone left {
            interfaces {
                lo0.0 {
                    host-inbound-traffic {
```

```
                    system-services {
                        ike;
                        ping;
                    }
                }
            }
            ge-0/0/3.100 {
                host-inbound-traffic {
                    system-services {
                        ping;
                    }
                    protocols {
                        bgp;
                        bfd;
                    }
                }
            }
        }
    }
    security-zone right {
        interfaces {
            ge-0/0/4.101 {
                host-inbound-traffic {
                    system-services {
                        ping;
                    }
                    protocols {
                        bgp;
                        bfd;
                    }
                }
            }
        }
    }
    security-zone untrust {
        interfaces {
            ge-0/0/0.102 {
                host-inbound-traffic {
                    system-services {
                        ping;
                    }
                    protocols {
                        bfd;
```

```
                            bgp;
                        }
                    }
                }
            }
        }
    }
}
interfaces {
    st0 {
        unit 0 {
            family inet;
        }
    }
}
routing-instances {
    vr {
        interface st0.0;
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups monitor-simple
chassis {
    high-availability {
        services-redundancy-group 1 {
            monitor {
                bfd-liveliness 10.0.31.1 {
                    src-ip 10.0.31.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.100;
                }
                bfd-liveliness 10.0.33.1 {
                    src-ip 10.0.33.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.101;
                }
                bfd-liveliness 10.0.38.1 {
                    src-ip 10.0.38.10;
```

```
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.102;
                }
                interface {
                    ge-0/0/0;
                }
            }
        }
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups monitor-advanced
chassis {
    high-availability {
        services-redundancy-group 1 {
            monitor {
                monitor-object endpoints {
                    object-threshold 200;
                    ip {
                        threshold 100;
                        destination-ip 10.0.30.10 {
                            routing-instance vr;
                            weight 50;
                        }
                        destination-ip 10.0.35.10 {
                            routing-instance vr;
                            weight 50;
                        }
                    }
                }
                monitor-object routers {
                    object-threshold 200;
                    bfd-liveliness {
                        threshold 100;
                        destination-ip 10.0.31.1 {
                            src-ip 10.0.31.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/3.100;
```

```
                            weight 100;
                        }
                        destination-ip 10.0.33.1 {
                            src-ip 10.0.33.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/4.101;
                            weight 100;
                        }
                        destination-ip 10.0.38.1 {
                            src-ip 10.0.38.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/0.102;
                            weight 100;
                        }
                    }
                }
                srg-threshold 200;
            }
        }
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups mnha-sync-icl
system {
    commit {
        peers {
            vsrx-mnha-n1 {
                routing-instance icl;
            }
        }
    }
    static-host-mapping {
        vsrx-mnha-n1 inet 172.26.0.2;
```

```
    }
}
```

```
[edit]
user@vsrx-mnha-n0# show groups icd
chassis {
    high-availability {
        local-id {
            local-forwarding-ip 172.26.0.11;
        }
        peer-id 2 {
            peer-forwarding-ip {
                172.26.0.12;
                interface lo0.1;
                liveness-detection {
                    minimum-interval 1000;
                    multiplier 5;
                }
            }
        }
    }
}
interfaces {
    lo0 {
        unit 1 {
            family inet {
                address 172.26.0.11/32;
            }
        }
    }
}
```

SRX-02

```
[edit]
user@vsrx-mnha-n1# show chassis high-availability
local-id {
    2;
    local-ip 172.26.0.2;
```

```
}
peer-id 1 {
    peer-ip 172.26.0.1;
    interface lo0.1;
    routing-instance icl;
    vpn-profile icl;
    liveness-detection {
        minimum-interval 1000;
        multiplier 3;
    }
}
services-redundancy-group 0 {
    peer-id {
        1;
    }
}
services-redundancy-group 1 {
    deployment-type routing;
    peer-id {
        1;
    }
    activeness-probe {
        dest-ip {
            10.0.30.1;
            src-ip 172.25.0.0;
            routing-instance vr;
        }
    }
    active-signal-route {
        172.24.0.1;
    }
    backup-signal-route {
        172.24.0.0;
    }
    prefix-list srg1-prefix {
        routing-instance vr;
    }
    managed-services ipsec;
    process-packet-on-backup;
```

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

```
[edit]
user@vsrx-mnha-n1# show groups mnha-sync

when {
    peers [ vsrx-mnha-n0 vsrx-mnha-n1 ];
}
security {
    ike {
        proposal ike-prop {
            authentication-method pre-shared-keys;
            dh-group group20;
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 28800;
        }
        policy ike-policy {
            proposals ike-prop;
            pre-shared-key ascii-text "$ABC123"; ## SECRET-DATA
        }
        policy icl {
            proposals ike-prop;
        }
        gateway r1 {
            ike-policy ike-policy;
            address 10.0.30.1;
            dead-peer-detection {
                probe-idle-tunnel;
                interval 5;
                threshold 5;
            }
            external-interface lo0.0;
            version v2-only;
        }
        gateway icl {
            ike-policy icl;
            version v2-only;
        }
    }
    ipsec {
```

```
        proposal ipsec-prop {
            encryption-algorithm aes-256-gcm;
            lifetime-seconds 3600;
        }
        policy ipsec-policy {
            perfect-forward-secrecy {
                keys group20;
            }
            proposals ipsec-prop;
        }
        vpn r1 {
            bind-interface st0.0;
            ike {
                gateway r1;
                ipsec-policy ipsec-policy;
            }
            traffic-selector ts1 {
                local-ip 10.0.35.11/32;
                remote-ip 10.0.30.11/32;
            }
            establish-tunnels immediately;
        }
        vpn icl {
            ha-link-encryption;
            ike {
                gateway icl;
                ipsec-policy ipsec-policy;
            }
        }
    }
    flow {
        tcp-mss {
            ipsec-vpn {
                mss 1400;
            }
        }
        tcp-session {
            strict-syn-check;
        }
    }
    policies {
        from-zone icl to-zone icl {
            policy permit {
```

```
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        global {
            policy internal {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                    from-zone [ right vpn left ];
                    to-zone [ left right vpn ];
                }
                then {
                    permit;
                    log {
                        session-close;
                    }
                }
            }
            policy untrust {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                    from-zone [ left right ];
                    to-zone untrust;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone vpn {
            interfaces {
```

```
            st0.0;
        }
    }
    security-zone left {
        interfaces {
            lo0.0 {
                host-inbound-traffic {
                    system-services {
                        ike;
                        ping;
                    }
                }
            }
            ge-0/0/3.100 {
                host-inbound-traffic {
                    system-services {
                        ping;
                    }
                    protocols {
                        bgp;
                        bfd;
                    }
                }
            }
        }
    }
    security-zone right {
        interfaces {
            ge-0/0/4.101 {
                host-inbound-traffic {
                    system-services {
                        ping;
                    }
                    protocols {
                        bgp;
                        bfd;
                    }
                }
            }
        }
    }
    security-zone untrust {
        interfaces {
```

```
                ge-0/0/0.102 {
                    host-inbound-traffic {
                        system-services {
                            ping;
                        }
                        protocols {
                            bfd;
                            bgp;
                        }
                    }
                }
            }
        }
    }
}
interfaces {
    st0 {
        unit 0 {
            family inet;
        }
    }
}
routing-instances {
    vr {
        interface st0.0;
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups monitor-simple
chassis {
    high-availability {
        services-redundancy-group 1 {
            monitor {
                bfd-liveliness 10.0.32.1 {
                    src-ip 10.0.32.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/3.100;
                }
```

```
                bfd-liveliness 10.0.34.1 {
                    src-ip 10.0.34.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/4.101;
                }
                bfd-liveliness 10.0.39.1 {
                    src-ip 10.0.39.10;
                    routing-instance vr;
                    session-type singlehop;
                    interface ge-0/0/0.102;
                }
                interface {
                    ge-0/0/0;
                }
            }
        }
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups monitor-advanced
chassis {
    high-availability {
        services-redundancy-group 1 {
            monitor {
                monitor-object endpoints {
                    object-threshold 200;
                    ip {
                        threshold 100;
                        destination-ip 10.0.30.10 {
                            routing-instance vr;
                            weight 50;
                        }
                        destination-ip 10.0.35.10 {
                            routing-instance vr;
                            weight 50;
                        }
                    }
                }
                monitor-object routers {
```

```
                    object-threshold 200;
                    bfd-liveliness {
                        threshold 100;
                        destination-ip 10.0.32.1 {
                            src-ip 10.0.32.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/3.100;
                            weight 100;
                        }
                        destination-ip 10.0.34.1 {
                            src-ip 10.0.34.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/4.101;
                            weight 100;
                        }
                        destination-ip 10.0.39.1 {
                            src-ip 10.0.39.10;
                            routing-instance vr;
                            session-type singlehop;
                            interface ge-0/0/0.102;
                            weight 100;
                        }
                    }
                }
                srg-threshold 200;
            }
        }
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups mnha-sync-icl

system {
    commit {
        peers {
            vsrx-mnha-n0 {
                routing-instance icl;
            }
```

```
        }
    }
    static-host-mapping {
        vsrx-mnha-n0 inet 172.26.0.1;
    }
}
```

```
[edit]
user@vsrx-mnha-n1# show groups icd
chassis {
    high-availability {
        local-id {
            local-forwarding-ip 172.26.0.12;
        }
        peer-id 1 {
            peer-forwarding-ip {
                172.26.0.11;
                interface lo0.1;
                liveness-detection {
                    minimum-interval 1000;
                    multiplier 5;
                }
            }
        }
    }
}
interfaces {
    lo0 {
        unit 1 {
            family inet {
                address 172.26.0.12/32;
            }
        }
    }
}
```

Software Upgrade in Multinode High Availability

IN THIS SECTION

Overview

SRX Series Firewalls deployed in an MNHA configuration can be upgraded with minimal disruption by sequentially upgrading each device. Depending on your device architecture, use one of the following CLI commands to initiate the Junos upgrade- `request system software add` or `request vmhost software add` .

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

- Releases 22.4R1 and later are not compatible with earlier Junos OS releases for synchronizing sessions during a regular upgrade. Use the Isolated Nodes Upgrade Procedurein such cases.
- Upgrading from 22.3 to the next release may cause brief traffic disruption.
- You may see `Peer Hardware Incompatible: SPU SLOT MISMATCH` during upgrades from 21.4R1 onwards.
- NAT sessions are not synced during interim upgrade stages in releases prior to 23.4R2.
- Always upgrade both nodes to the same Junos OS version.

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.

When you are upgrading SRX Series Firewalls in Multinode High Availability to Junos OS Release 22.4R1 or to a higher release, from an earlier Junos OS release, you can use the Isolated Nodes Upgrade Procedure. Junos OS Release 22.4R1 and higher releases are not compatible with earlier Junos OS releases for synchronizing sessions during a regular upgrade.

Before You Begin

Before performing an upgrade on an SRX Series device in MNHA) configuration, it is recommended to redirect the traffic away from the device in a controlled way. This can be done using one of the following methods:

- Manual failover —Trigger a manual failover to shift traffic to the peer device.
- Software upgrade mode —Temporarily configure the device with the following command:

```
user@host# set chassis high-availability software-upgrade
```

This command introduces a device failure with failure code SU (Software Upgrade). As a result, Services Redundancy Groups (SRG) 1 and above will transition to an Ineligible state (instead of Active or Backup) on the device being upgraded. This causes the associated traffic to automatically fail over to the other MNHA cluster member.

NOTE: If your MNHA cluster is configured with only SRG0 and includes the `install-on-failure-route` option, you can still redirect traffic by using the `set chassis high-availability software-upgrade` configuration to move traffic off the device gracefully.

Software Upgrade

Preparation Checklist

Consider the following best practices when you plan your software upgrade:

- Ensure both nodes are online and running the same Junos OS version. Check the current Junos OS software version on your device using the show version command.
- Verify storage availability: `show system storage`
- Check hardware status:
  - `show chassis fpc pic-status`
  - `show chassis alarms`
- Ensure that there are no uncommitted changes.
- Backup configuration and license keys.

- Download the Junos OS image to /var/tmp on both devices.
- Ensure your high availability setup is healthy, functional, and that the interchassis link (ICL) is up.

  `show chassis high-availability information`
- Prepare your SRX Series Firewalls for an upgrade using the checklist available in .

TIP: We recommend that you perform software upgrades during a maintenance window. For details on preparing your device for an upgrade, see Preparing for Software Installation and Upgrade (Junos OS).

Download Software

Download the Junos OS image from the Juniper Networks Support page on both SRX Series Firewalls and save it in the **/var/tmp** location. Example:

```
user@host> request system software add /var/tmp/junos-install-vsrx3-x86-64-22.3R1.3.tgz no-copy
```

Upgrade Procedure

Follow the steps in this procedure to upgrade SRX Series devices configured in a Multinode High Availability (MNHA) setup. In this example, the cluster consists of two devices: srx-01 (currently active) and srx-02 (currently backup). The upgrade process begins with the backup node (srx-02), followed by the active node (srx-01), ensuring minimal service disruption.

1. Ensure 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
```

2. Initiate the software upgrade process on the backup node (srx-02) and commit the configuration

```
user@srx-02# set chassis high-availability software-upgrade
```

This command triggers a local failover for SRG0 and marks SRG1 (if present) as INELIGIBLE, allowing the peer node to take or retain the active role

3. Verify the status of Multinode High Availability. 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.

```
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
```

4. Confirm that the other device (srx-01) is in an 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**.

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 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
```

6. Reboot the device using the `request system reboot` command after successful installation.

7. Check the Junos OS version after reboot.

```
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 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`.

9. Remove the `software-upgrade` statement and commit the configuration.

```
user@srx-02# delete chassis high-availability software-upgrade
```

When you remove the `software-upgrade` statement, the node failover state and any installed routes are cleared. Until this statement is removed, the node remains offline and all SRGs stay in the INELIGIBLE state. This effectively isolates the node from handling traffic during the upgrade, as long as the peer remains healthy.

10. 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.

11. Check interfaces, routing protocols, routes advertised and so on to confirm that your setup is operating normally.

12. Now you can proceed to upgrade the other device (SRX-01) using the same procedure.

NOTE: (Optional) In case if you face any issues and cannot 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 install-on-failure-route

For setups using only SRG0 (without A/B state support), we recommend configuring the install-on-failure-route. This route can be referenced in route policies to advertise less preferred paths during software upgrade scenarios or node failures. In this method, you can divert the traffic by changing the route. Here, traffic can still go through the node and interface remains up.

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

```
set routing-instances MNHA-signal-routes instance-type virtual-router
```

2. Configure the `install-on-failure-route` 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.

```
set routing-instances MNHA-signal-routes instance-type virtual-router
set chassis high-availability services-redundancy-group 0 install-on-failure-route 10.39.1.3
routing-instance MNHA-signal-routes
set chassis high-availability services-redundancy-group 1 active-signal-route 10.39.1.1
routing-instance MNHA-signal-routes
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`.

```
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
MNHA-signal-routes.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
MNHA-signal-routes.inet.0
set policy-options condition failure_route_exists if-route-exists address-family inet
10.39.1.3/32
set policy-options condition failure_route_exists if-route-exists address-family inet table
MNHA-signal-routes.inet.0
```

4. Create the policy statement to refer the condition as one of the matching term.

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

5. Initiate software upgrade as mentioned in previous steps ("Software Upgrade" on page 1052).

Deprecated Method (shutdown-on-failure interface)

Starting in Junos OS Release 24.3R1 onwards, the `shutdown-on-failure` functionality is deprecated— rather than immediately removed—to provide backward compatibility and an opportunity to bring your configuration into compliance with the new configuration. As a part of this change, the [**set chassis high-availability services-redundancy-group 0 shutdown-on-failure interface-name**] configuration statement deprecated.

Previously, traffic had to be diverted manually by shutting down interfaces. You can now use the software-upgrade command to keep the node offline and all SRGs in the INELIGIBLE state for the duration of the upgrade. This effectively isolates the node from handling traffic.

If you're using Junos OS 22.4 or earlier, we recommend using the legacy methods to divert traffic during the upgrade.

RELATED DOCUMENTATION

- Two-Node Multinode High Availability | 573
- Prepare Your Environment for Multinode High Availability Deployment | 620
- Insert Additional SRX5K-SPC3 in a Multinode High Availability Setup | 1063
- Example: Configure Multinode High Availability in a Default Gateway Deployment | 743
- Example: Configure Multinode High Availability in a Layer 3 Network | 698
- Example: Configure Multinode High Availability in a Hybrid Deployment | 778

Insert Additional SRX5K-SPC3 in a Multinode High Availability Setup

IN THIS SECTION

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

Two-Node Multinode High Availability | **573**

Prepare Your Environment for Multinode High Availability Deployment | **620**

Software Upgrade in Multinode High Availability | **1051**

Example: Configure Multinode High Availability in a Default Gateway Deployment | **743**

Example: Configure Multinode High Availability in a Layer 3 Network | **698**

Example: Configure Multinode High Availability in a Hybrid Deployment | **778**

Multinode High Availability Support for vSRX Virtual Firewall Instances

IN THIS SECTION

Multinode High Availability addresses high availability requirements for private and public cloud deployments by offering interchassis resiliency.

We support Multinode High Availability on Juniper Networks vSRX Virtual Firewall Virtual Firewalls for the private (Kernel-based virtual machine [KVM] and VMware ESXi) and public cloud (AWS) deployments.

You can configure Multinode High Availability on vSRX instances by using the same method as for physical SRX Series firewalls for private cloud deployments.

To configure Multinode High Availability in VMware ESXi, and KVM:

- Deploy two vSRX Virtual Firewalls instances in private clouds. Refer Install vSRX Virtual Firewall with KVM or Install vSRX Virtual Firewall with VMware vSphere Web Client.
- Setup Multinode High Availability using the instructions available in the following topics:
  - Example: Configure Multinode High Availability in a Default Gateway Deployment
  - Example: Configure Multinode High Availability in a Layer 3 Network
  - Example: Configure Multinode High Availability in a Hybrid Deployment

To configure Multinode High Availability in public cloud deployments:

- See "Multinode High Availability in AWS Deployments" on page 1071.

ICL Encryption and Flexible Datapath Failure Detection Support

The vSRX Virtual Firewall in Multinode High Availability deployed in private clouds (KVM and VMware ESXi) supports ICL Encryption and Flexible Datapath Failure Detection.

- ICL Encryption uses IPsec protocols to secure synchronization messages between high-availability nodes, ensuring data privacy. See Example: Configure Multinode High Availability in a Layer 3 Network for configuration details.
- Flexible Datapath Failure Detection offers path monitoring with granular control through weighted features, supporting IP, Bidirectional Forwarding Detection (BFD), and interface monitoring. SeeFlexible Path Monitoring for more details.

Understanding Multinode High Availability Dual Path Interchassis Link (ICL)

Multinode High Availability (MNHA) supports dual path interchassis link (ICL) over aggregated Ethernet (AE) and loopback interfaces. This enhancement enables efficient traffic distribution and improved HA reliability across public clouds such as AWS, Azure, and Google Cloud Platform (GCP), as well as private clouds using KVM and VMware. For configuring ICL, note that:

- In public cloud settings such as AWS, Azure, and GCP, loopback interfaces are preferred due to constraints on AE interfaces.
- In private cloud environments utilizing KVM and VMware, you can use AE interfaces for establishing dual path ICLs, characterized by flexible configurations that allow you to use various network interface cards.

Benefits of Dual Path ICL in MNHA

- Enhances compatibility with public cloud environments like AWS, Azure, and GCP, enabling efficient Layer 3 high availability deployment.
- Improves traffic distribution and load balancing through the use of AE and loopback interfaces, ensuring optimal performance in both public and private cloud setups.

Support for AE Interface as Dual-Path ICL in Private Cloud Environments

In a private cloud setup using KVM or VMware, you can configure an Aggregated Ethernet (AE) interface as a dual-path ICL. This is supported with:

- Virtio NICs for KVM
- VMXNET3 NICs for VMware

For setups using SR-IOV NICs (such as Intel I40E, E810, or Mellanox MLX5), the AE interface must support virtual MAC (vMAC) functionality.

Each member (child) interface of the AE must be a Gigabit Ethernet (GE) interface and must reside on different physical functions (PFs)—typically meaning they should be on different line cards.

Efficient Distribution of ICL Traffic

Managing Traffic in vSRX with Aggregated Ethernet (AE): Using LACP and Hypervisor Bridge Configuration

In a vSRX setup, ensuring reliable and efficient traffic flow is crucial, especially when dealing with aggregated Ethernet (AE) interfaces. If one child interface within the AE group becomes out-of-order or fails, it can lead to traffic loss. To mitigate this risk, you can employ Link Aggregation Control Protocol (LACP) or configure the child interfaces to share the same bridge in the hypervisor.

Using LACP on AE Interface

LACP is a protocol that helps dynamically manage the link aggregation, ensuring that all child interfaces are working in coordination. By configuring LACP, you can automatically handle interface failures and maintain traffic flow without manual intervention. Here is how you can configure LACP on an AE interface:

```
[edit]
user@host# set interfaces ae0 aggregated-ether-options lacp active
user@host# set interfaces ae0 aggregated-ether-options lacp periodic fast
```

- Active: This setting enables the active LACP mode, where the vSRX actively sends LACP packets to the peer to form and maintain the LACP link.
- Periodic Fast: This setting reduces the LACP timeout interval, allowing quicker detection of link failures and faster response to maintain traffic flow.

Configuring Child Interfaces on the Same Bridge

In situations where LACP is hard to support or implement, another approach is to ensure that all child interfaces in the AE group are connected to the same bridge within the hypervisor. This configuration can help maintain consistent connectivity and avoid traffic loss when one interface is out-of-order.

Below is an example of XML configuration for two child interfaces (ge-0/0/0 and ge-0/0/3) using the same bridge (bridge-vsrx-mnha-icl) in a KVM hypervisor:

```
<interface type='bridge'>
      <mac address='52:54:00:c2:76:70'/>
      <source bridge='bridge-vsrx-mnha-icl'/>                                    <<<< here
bridge bridge-vsrx-mnha-icl
      <model type='virtio'/>
      <driver name='vhost' queues='4'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/> </interface>
```

```
 <interface type='bridge'>
      <mac address='52:54:00:5f:65:9f'/>
      <source bridge='bridge-vsrx-mnha-icl'/>                                        <<<< here
same bridge bridge-vsrx-mnha-icl
      <model type='virtio'/>
      <driver name='vhost' queues='4'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>  </interface>
```

In this configuration: Both interfaces are connected to the bridge-vsrx-mnha-icl bridge, ensuring they share the same network segment within the hypervisor. This setup can help manage traffic effectively even if one of the interfaces experiences issues.

Using AE Interfaces in MNHA Confihuration

The following configuration snippet sets up a MNHA environment using AE interface:

```
[edit]
user@host# set chassis high-availability local-id 2
user@host# set chassis high-availability local-id local-ip 11.1.1.11
user@host# set chassis high-availability peer-id 1 peer-ip 11.1.1.12
user@host# set chassis high-availability peer-id 1 interface ae0
user@host# set interfaces ge-0/0/0 ether-options 802.3ad ae0
user@host# set interfaces ge-0/0/1 ether-options 802.3ad ae0
user@host# set interfaces ae0 unit 0 family inet address 10.10.10.1/24
```

The above sample shows configuring local and peer chassis IDs and IPs for HA communication, associating physical interfaces with an aggregated Ethernet interface (`ae0`) using LACP, and assigning an IP address to the logical unit of the aggregated interface. In this example:

- The AE interface (ae0) is used as ICL between the nodes.
- Physical interfaces `ge-0/0/0` and `ge-0/0/1` are bundled into `ae0` using 802.3ad link aggregation.

Balanced ICL Traffic Distribution Across PFEs

We have improved the system to ensure balanced ICL traffic distribution across all PFE processing units on the receiving side in an MNHA setup.

In our MNHA setup, traffic across ICL is efficiently managed to ensure high performance. Outgoing traffic is automatically distributed across multiple flow processing units, a built-in feature of the MNHA architecture. To improve incoming traffic handling, we use a five-tuple hashing method on the ICL port.

This evenly spreads traffic across all Packet Forwarding Engine (PFE) processing units, resulting in better load balancing and overall network efficiency.

The command `set chassis high-availability peer-id <id_num> interface <interface-name>` supports GE, AE, and loopback interfaces, enabling high-availability configurations tailored to your specific deployment needs.

On the sending side, the determination of which port is used for ICL traffic in the Packet Forwarding Engine does not rely solely on this configuration. Instead, it depends on IP and route lookup results. For SRX Series Firewalls, priority queues are enabled by default on all ports help manage traffic, especially for aggregate interfaces and loopback interfaces . However, vSRX3.0 cannot utilize this approach.

For the receiving side distribution in vSRX3.0, a new CLI configuration is required to specify ICL ports for enabling 5-tuple hashing. This is done with the command:

```
user@host# set security forwarding-options receive-side-scaling nic-rss hash five-tuple ports [port_ids]
```

Example:

```
[edit]
user@host# set security forwarding-options receive-side-scaling nic-rss hash five-tuple ports 0
user@host# set security forwarding-options receive-side-scaling nic-rss hash five-tuple ports 1
```

In this configuration, ports 0 and port 1 are configured to use five-tuple hashing for receive side scaling. This ensures that incoming traffic is efficiently distributed based on source and destination IP addresses, ports, and protocol.

Multinode High Availability in AWS Deployments

SUMMARY

Read this topic to understand Multinode High Availability support for vSRX Virtual Firewall instances in Amazon Web Services (AWS) deployments.

IN THIS SECTION

Multinode High Availability in AWS

IN THIS SECTION

You can configure Multinode High Availability on the vSRX Virtual Firewall firewalls deployed on AWS. Participating nodes run both active control and data planes at the same time and the nodes 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.

Let's begin by getting familiar with the Multinode High Availability terms specific to the AWS deployment.

Terminology

Term	Description
Elastic IP address	Public IPv4 address that is routable from a specified network or from the Internet. Elastic IP addresses are dynamically bound to an interface of any node in a Multinode High Availability setup. At any given time, these addresses are bound to only one interface and are also bound to the same node. The Multinode High Availability setup uses Elastic IP addresses to control the traffic in AWS deployments. Elastic IP address acts similar to floating IP address in the Layer 3 deployment or a virtual IP address as in the default gateway deployment. The node with an active SRG1 owns the Elastic IP address and draws the traffic toward it.

(Continued)

Term	Description
Interchassis 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 the ge-0/0/0 interface to configure an ICL. The ICL uses the MAC address assigned by AWS (not the virtual MAC created by vSRX Virtual Firewall). When you configure the ICL, ensure that the IP address is a subnet of of the virtual private cloud (VPC). Note that Multibode High Availability does not support cross-VPC deployment
Juniper Services Redundancy Protocol (jsrpd) process	Process that manages activeness determination and enforcement, and provide split-brain protection.

IPsec VPN Support

Starting in Junos OS Release 24.4R1, we support IPsec VPN for active/backup Multinode High Availability in AWS deployments.

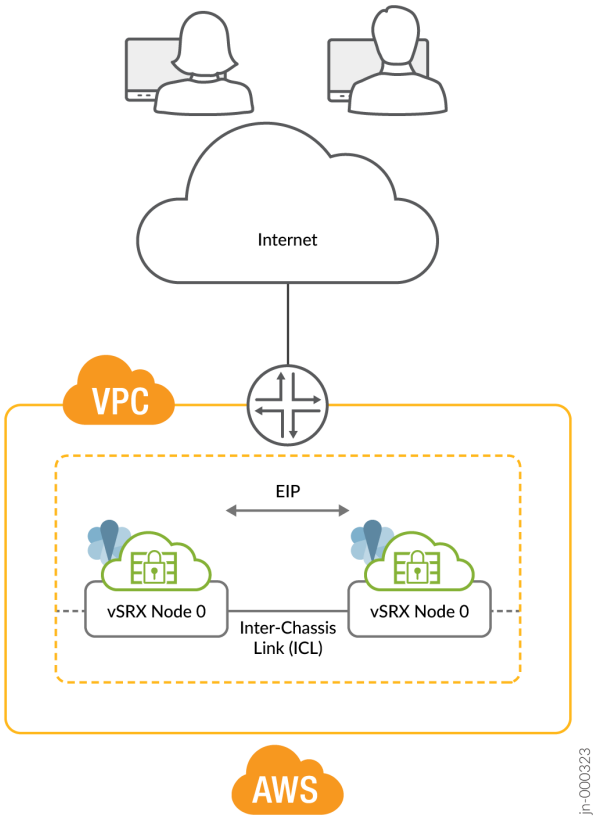
Limitation

Multinode High Availability doesn't support multiple SRG configurations (active/active) in public cloud deployments. Active/backup mode supports SRG0 and SRG1. IPsec VPN tunnel anchors at the SRG1, which works in a stateful active/backup mode. All VPN tunnels terminate on the device where the SRG1 is active.

Architecture

Figure 71 on page 1074 shows two vSRX Virtual Firewall instances form an HA pair in the Multinode High Availability deployment in AWS. One vSRX Virtual Firewall instance acts as the active node and the other as the backup node.

Figure 71: Public Cloud Deployment



In a Multinode High Availability setup, an ICL connects the two nodes (vSRX Virtual Firewall instances) and helps synchronize the control-plane and data-plane states.

In Multinode High Availability setup, two vSRX Virtual Firewall 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 Virtual Firewall instance on which the SRG1 is active hosts the Elastic IP address. The active node steers traffic toward it using the Elastic IP address. The backup node remains in standby mode and takes over on failover.

The Juniper Services Redundancy Protocol (jsrpd) process communicates with the 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 the AWS SDK API and draws traffic toward the new active node. 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. You configure the Elastic IP address 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 address using the probes. If the peer node is healthy, it responds to the probes. Otherwise, the jsrpd process communicates with the AWS infrastructure to enforce the active role for the healthy node.

Flexible ICL Interface on AWS

You can configure any ge-0/0/x interface for ICL on AWS. This is similar to ICL interface on Azure and GCP, where you can use ge-0/0/x for ICL. Previously, ICL interface on AWS was fixed and you could only use ge-0/0/0 for ICL.

For vSRX instances operating in an MNHA setup, AWS uses specific tags to ensure proper identification and management of HA pairs. Initially, two tags are used:

- LocalNodeID: This tag identifies the instance as the local node within the HA pair.
- PeerNodeID: This tag identifies the instance as peer node.

In addition to these two tags, we introduce four additional tags to enhance the configuration and management of MNHA setups by specifying interface details for both local and peer nodes:

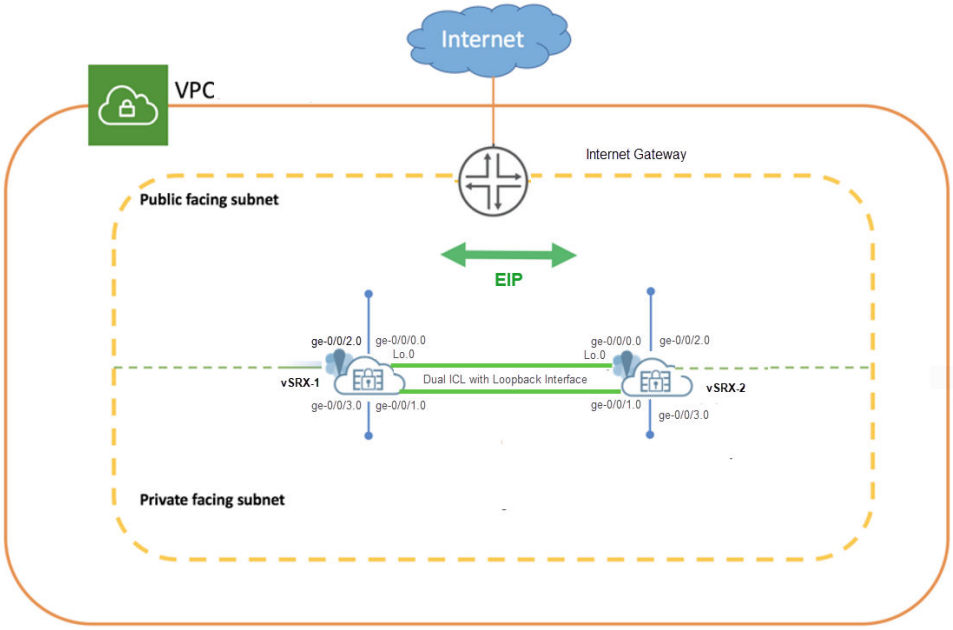
- LocalTrustInterface: Interface on the local vSRX instance that connects to the private subnet (often used for trusted internal communication).
- LocalUntrustInterface: Interface on the local vSRX instance that connects to the public subnet (typically used for untrusted external communication).
- PeerTrustInterface: Interface on the peer vSRX instance that connects to the private subnet.
- PeerUntrustInterface: Interface on the peer vSRX instance that connects to the public subnet,

Support for Loopback Interface on AWS

As the AE interface is incompatible with public cloud platforms such as AWS, GCP, and Azure, the loopback interface is used as part of the dual path ICL solution. Azure and GCP support loopback interfaces. Now, you can use loopback interface to configure dual path ICL on AWS as well. AWS requires loopback communication must be established over two or more physical interfaces (example: ge-0/0/x).

The following illustration provides the topology of vSRX firewalls in MNHA setup deployed in AWS.

Figure 72: Multinode High Availability in AWS Deployment with Dual-Path ICL



As shown in the topology, two vSRX Virtual Firewall instances (vSRX Virtual Firewall-1 and vSRX Virtual Firewall-2) are deployed in the Amazon VPC. The nodes communicate with each other using a routable IP address (Elastic IP address). The untrust side connects to a public network while the trust side connects to the protected resources

The topology uses loopback interfaces to implement dual path Inter-Chassis Link (ICL) between two vSRX firewalls within a VPC. vSRX devices use two interfaces (`ge-0/0/0` and `ge-0/0/1`) each for dual ICL connectivity. Here, loopback interface is used as the logical endpoint for ICL communication. Physical Interfaces (`ge-0/0/0` and `ge-0/0/1`) serve as the underlying transport paths for loopback communication.

Example: Configure Multinode High Availability in AWS Deployment

IN THIS SECTION

In this example, we'll show you how to configure Multinode High Availability on two vSRX Virtual Firewall instances in the Amazon Virtual Private Cloud (Amazon VPC).

Requirements

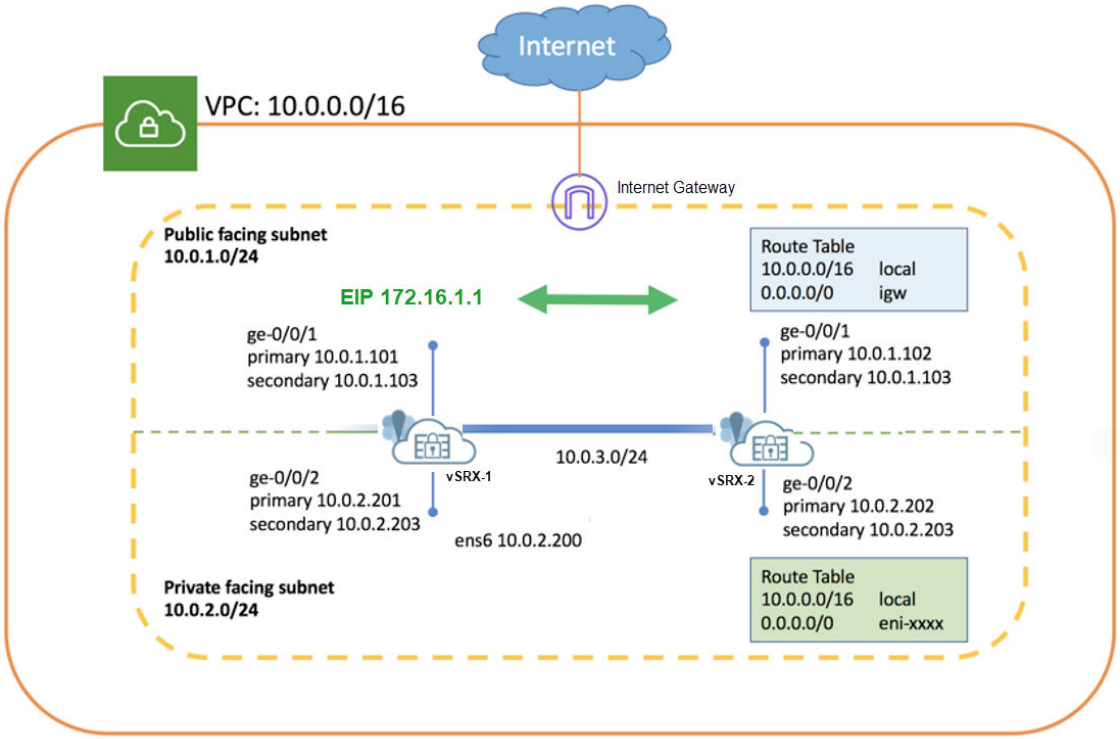
This example uses the following components:

- Two vSRX Virtual Firewall 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. See Configure an Amazon Virtual Private Cloud for vSRX for details.
- An Amazon VPC configured with its associated Internet gateway, subnets, route table, and security groups. See Configure an Amazon Virtual Private Cloud for vSRX.
- A vSRX Virtual Firewall instance launched and configured in Amazon VPC. See Launch a vSRX Instance on an Amazon Virtual Private Cloud.

Topology

Figure 73 on page 1078 shows the topology used in this example.

Figure 73: Multinode High Availability in AWS Deployment



As shown in the topology, two vSRX Virtual Firewall instances (vSRX Virtual Firewall-1 and vSRX Virtual Firewall-2) are deployed in the Amazon VPC. The nodes communicate with each other using a routable IP address (Elastic IP address). The untrust side connects to a public network while the trust side connects to the protected resources.

Complete the following configurations before configuring Multinode High Availability on the vSRX Virtual Firewall instances:

- Use instance tag in AWS to identify the two vSRX Virtual Firewall instances as Multinode High Availability peers. For example, you can use **vsrx-node-1** as the name of one peer (**Name** option) and **vsrx-node-2** as the HA peer (**ha-peer** option).
- Deploy both vSRX Virtual Firewall instances in the same Amazon VPC and availability zone.
- Assign IAM role for both the vSRX Virtual Firewall instances and launch vSRX Virtual Firewall instances as a Amazon Elastic Compute Cloud (EC2) instance with full permissions.
- Enable communication to the Internet by placing vSRX Virtual Firewall instances in the public subnet. In the Amazon VPC, public subnets have access to the Internet gateway.
- Configure a VPC with multiple subnets to host the high availability pair. The subnets are used to connect the two vSRX Virtual Firewall nodes using a logical connection (similar to the physical cable connecting ports). In this example, we have defined CIDR for VPC as 10.0.0.0/16, and created a total

of four subnets to host the vSRX Virtual Firewall traffic. You need a minimum of four interfaces for both vSRX Virtual Firewall instances. Table 61 on page 1079 provides the subnet and interface details.

Table 61: 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	ICL 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 the interface mapping with functionality mentioned in the table is for default configuration. We recommend to use the same mapping in the configuration.

- Configure interfaces with primary and secondary IP addresses. You can assign Elastic IP address as secondary IP addresses for an interface. You need the primary IP address while launching the instance. The secondary IP address is transferable from one vSRX Virtual Firewall node to another during a failover. Table 62 on page 1079 shows interface and IP address mappings used in this example.

  Table 62: Interface and IP Address Mappings

Instance	Interface	Primary IP Address	Secondary IP Address (Elastic IP Address)
vSRX Virtual Firewall-1	ge-0/0/1	10.0.1.101	10.0.1.103
	ge-0/0/2	10.0.2.201	10.0.2.203

Table 62: Interface and IP Address Mappings *(Continued)*

Instance	Interface	Primary IP Address	Secondary IP Address (Elastic IP Address)
vSRX Virtual Firewall-2	ge-0/0/1	10.0.1.102	10.0.1.103
	ge-0/0/2	10.0.2.202	10.0.2.203

- Configure neighboring routers to include vSRX Virtual Firewall in the data path and mark vSRX Virtual Firewall as the next hop for the traffic. You can use an Elastic IP address to configure the route. For example, use the command `sudo ip route x.x.x.x/x dev ens6 via 10.0.2.203`, where the 10.0.2.203 address is an Elastic IP address.

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 Virtual Firewall-1

```
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 Virtual Firewall-2

```
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 ge-0/0/0 as 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
```

We'll use the secondary IP address assigned to ge-0/0/1 and ge-0/0/2 as Elastic IP address.

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
```

   Here, you'll require 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 the peer node for interface monitoring, and configure the 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 with deployment type as cloud, assign an ID, and set preemption and activeness priority.

```
[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. Configure AWS deployment-related options. For example, specify eip-based as the service type and also, configure monitoring options such as AWS peer liveness.

```
[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
```

NOTE: In Multinode High Availability for vSRX Virtual Firewall instances in VMWare ESXi environment with VMXNET3 vNIC, configuration of virtual MAC address is not supported in the following statement:

`[set chassis high-availability services-redundancy-group <number> virtual-ip <id> use-virtual-mac`

Results

vSRX Virtual Firewall-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 Virtual Firewall-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 Virtual Firewall instance.

Action

From operational mode, run the following command:

vSRX Virtual Firewall-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 Virtual Firewall-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

Check whether Multinode High Availability is deployed in AWS cloud.

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 the the AWS deployment uses the EIP service type (for Elastic IP address) to control traffic.
- The field `Cloud Service Status: Bind to Local Node` indicates the binding of the Elastic IP address to the 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 Virtual Firewall-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 Virtual Firewall-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 including ID, IP address, interface.
- 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. The field `Status: ACTIVE` indicates that the particular SRG1 is in active role. You can also view activeness priority and preemption state in the output.
- 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
```

Meaning

In `Services Redundancy Group: 1` section, you can see the `Status: BACKUP`. This field indicates that the SRG-1 is in the backup mode.

Action

Initiate the failover on the active node (vSRX Virtual Firewall-1) and again run the command on the backup node (vSRX Virtual Firewall-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

In the `Services Redundancy Group: 1` section, the status of SRG1 changes from `BACKUP` to `ACTIVE`. The change in the field value indicates that the node has transitioned into the active role and the other node (previously active) has transitioned to the backup role. You can see the other node's status in the `Peer Information` option, which shows `BACKUP`.

SEE ALSO

- Two-Node Multinode High Availability | **573**
- Multinode High Availability Services | **624**
- Prepare Your Environment for Multinode High Availability Deployment | **620**
- Example: Configure Multinode High Availability in a Default Gateway Deployment | **743**
- Example: Configure Multinode High Availability in a Layer 3 Network | **698**
- Example: Configure Multinode High Availability in a Hybrid Deployment | **778**

Change History Table

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

Release	Description
change-completed	Starting in Junos OS Release 22.3R1, we support MNHA in Amazon Web Services (AWS) platform.
change-completed	Starting in Junos OS Release 25.4R1, you can configure any ge-0/0/x interface for ICL on AWS similar to Azure and GCP, where ICL interface is flexible and can be on any ge-0/0/x. Previously, ICL interface on AWS was fixed and you could only use ge-0/0/0 for ICL.

Multinode High Availability in Azure Cloud

SUMMARY

Read this document to understand how to configure Multinode High Availability on vSRX instances deployed on Azure cloud.

IN THIS SECTION

Overview

IN THIS SECTION

You can configure Multinode High Availability on vSRX Virtual Firewalls deployed in the Microsoft Azure Cloud. Microsoft Azure is Microsoft's application platform for the public cloud. It is an open, flexible, enterprise-grade cloud computing platform for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers.

You can configure a pair of vSRX virtual firewalls on Azure to operate as in an active/backup Multinode high availability configuration. Participating nodes run both active control and data planes at the same

time. The nodes backup each other to ensure a fast synchronized failover in case of a system or hardware failure. The interchassis link (ICL) connection between the two devices synchronizes and maintains the state information and handles device failover scenarios.

You can configure Multinode High Availability on vSRX Virtual Firewall VMs by customizing firewall deployment settings in Microsoft Azure Cloud.

IPsec VPN Support

Starting in Junos OS Release 24.4R1, we support IPsec VPN for active/backup Multinode High Availability in Azure Cloud deployments.

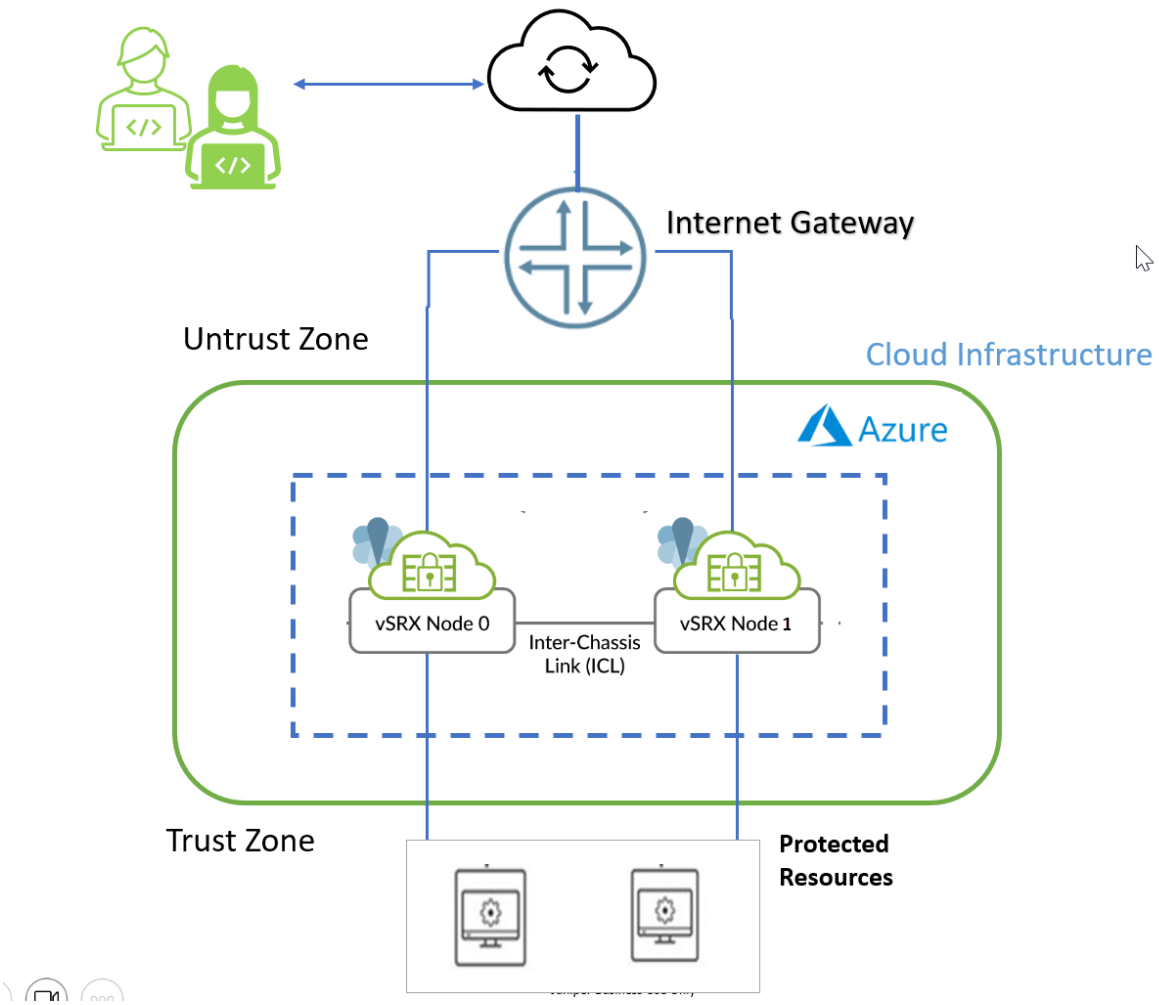
Limitation

Multinode High Availability doesn't support multiple SRG configurations (active/active) in public cloud deployments. Active/backup mode supports SRG0 or SRG1. IPsec VPN tunnel anchors at the SRG1, which works in a stateful active/backup mode. All VPN tunnels terminate on the device where the SRG1 is active.

Architecture

Figure 74 on page 1102 shows two vSRX Virtual Firewall instances form a Multinode high availability pair in the in the Azure cloud. One vSRX Virtual Firewall instance acts as the active node and the other instance acts as the backup node. Both nodes connect to each other using an ICL to synchronize and maintain state information and to handle device failover scenarios.

Figure 74: vSRX in Multinode High Availability Deployments in Azure Cloud



vSRX Virtual Firewall requires two public IP addresses and one or more private IP address for each individual instance group. The public subnets consist of one for the management interface (fxp0) and one for a revenue (data) interface. You can use any four revenue interfaces for the subnet configuration. The private interface is connected to the protected resources. It ensures that all traffic between applications on the private subnet and the Internet must pass through the vSRX Virtual Firewall instance.

For Multinode High Availability on Azure, you must deploy both the firewalls within the same Azure Resource Group. An Azure Resource Group is a logical container that holds related resources for an Azure solution. It can include all the resources for the solution, or only those resources that you want to manage as a group.

You must allocate a node-specific primary address to each node and a common secondary or floating IP address to both the nodes. The secondary IP address, which acts as a floating IP address, is always attached to the active node. In case of failure on the current active node, the secondary IP address

transitions from the failed active node to the current active node. The new active node ensures the continue flow of traffic.

Initially both the nodes are launched with predefined tags stating which one is the owner of the secondary IP address during boot up. That particular node starts operating as the active node and other one starts as a backup node.

Split-Brain Protection

The split-brain scenario refers to a situation where both nodes of the Multinode High Availability system become stuck in the same state, either active or backup, when the inter-chassis link (ICL) between the nodes is down. To prevent this state, both nodes attempt to probe the primary IP address of the trust or the untrust interface, based on the configuration.

When an Interchassis Link (ICL) experiences a failure along with a probe failure, the node that does not receive a reply from its peer will take on the active role. However, if the probe succeeds and confirms that the peer node is still operational, the node will maintain its current state. This probing process persists until the ICL is restored.

Example: Configure Multinode High Availability in Azure Cloud Deployment

IN THIS SECTION

You can configure Multinode High Availability on vSRX Virtual Firewalls deployed in the Microsoft Azure Cloud. Microsoft Azure is Microsoft's application platform for the public cloud. It is an open, flexible, enterprise-grade cloud computing platform for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers.

You can configure a pair of vSRX virtual firewalls on Azure to operate as in an active/backup Multinode high availability configuration. Participating nodes run both active control and data planes at the same time. The nodes backup each other to ensure a fast synchronized failover in case of a system or

hardware failure. The interchassis link (ICL) connection between the two devices synchronizes and maintains the state information and handles device failover scenarios.

TIP:

Table 63: Readability Score and Time Estimates

Reading Time	1 hour
Configuration Time	2 hour

Example Prerequisites

Read this topic to understand how to configure the Multinode High Availability solution on SRX Series Firewalls.

In this example, we'll show you how to configure Multinode High Availability on two vSRX Virtual Firewall instances deployed in the Azure Cloud.

VMs requirements	Two vSRX Virtual Firewalls deployed on Azure Cloud
Software requirements	Junos OS Release 23.4R1 or later releases

Licensing requirements	• Review the requirements for deploying a vSRX Virtual Firewall VM in Microsoft Azure Cloud. Requirements for vSRX Virtual Firewall on Microsoft Azure .
	• Microsoft account (username and password) to log into the Microsoft Azure portal
	• Microsoft Azure subscription (see Microsoft Azure). Ensure that your Azure subscription includes the following for your vSRX Virtual Firewall VM:
	  • Resource group (Create a Resource Group).
	  • Storage account (Create a Storage Account)
	  • Virtual network (Create a Virtual Network).
	Use vSRX Virtual Firewall license or request an evaluation license. Licenses can be procured from the Juniper Networks License Management System (LMS).

Before You Begin

Benefits	Increases the availability of vSRX Series firewalls deployed in Azure that results in improved reliability and reduced downtime.
Know more	vSRX Virtual Firewall with Microsoft Azure Cloud
Learn more	Deploying Juniper Security in AWS and Azure

Functional Overview

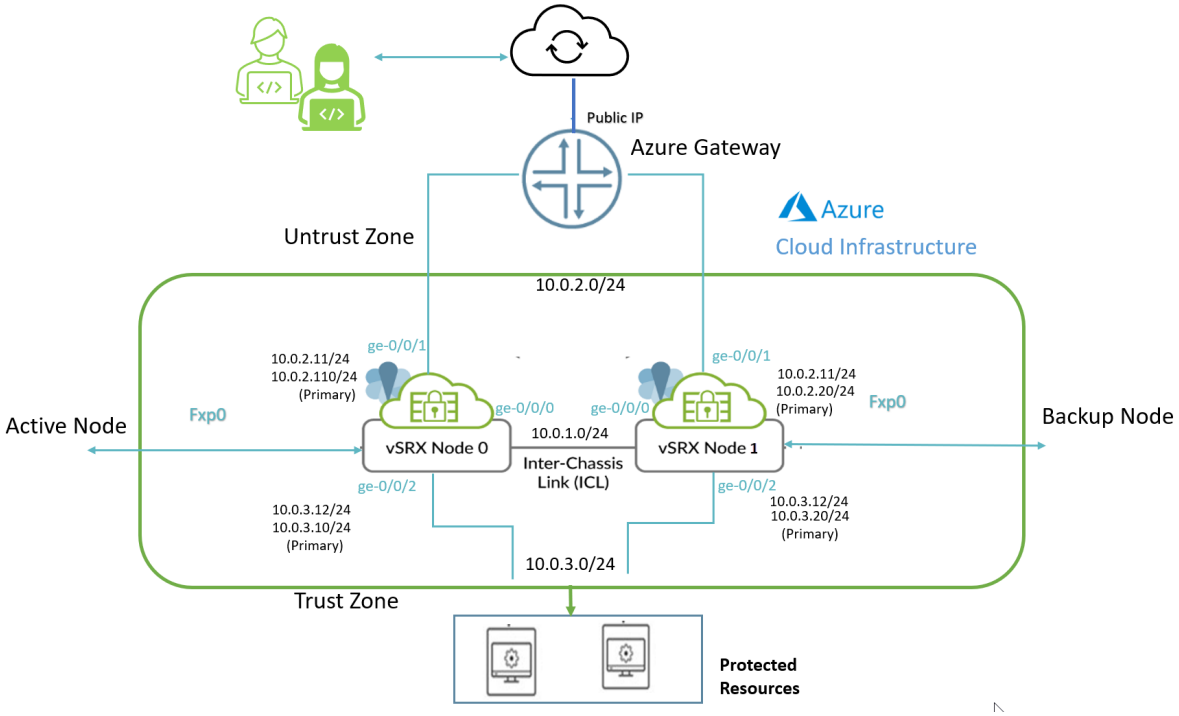
Technologies used	• Interfaces and zones • Multinode high availability • Monitoring options • Routing policy, protocols, and routing options

Primary verification tasks	• High Availability information about both nodes • Multinode High Availability status on Azure Cloud

Topology Overview

Figure 75 on page 1106 shows the topology used in this example.

Figure 75: Multinode High Availability Configuration on vSRX VMs Deployed in Azure



As shown in the topology, two vSRX Virtual Firewall instances (vSRX Node 0 and vSRX Node 1) are deployed in the Azure Cloud. The untrust side connects to a public network while the trust side connects to the protected resources.

Two vSRX Virtual Firewall instances are placed in the public subnet as public subnets have access to the Internet gateway.

In a Multinode High-Availability setup, a floating IP address moves between two SRX Series Firewalls, if a link or firewall fails. Since the primary interface's IP address on the Azure firewall cannot be moved,

you assign a secondary IP address as a floating IP. When the active node fails, the floating IP address shifts to the backup node, allowing it to seamlessly handle traffic as the new active peer.

In addition, you must set up an interchassis link (ICL) for data synchronization and maintaining state information. The nodes communicate with each other using a routable IP address that is assigned to ICL.

The following table provides interface and IP address details used in this example.

Table 64: Interfaces and IP Addresses

Interface	Function	Primary Node (vSRX Node 0)	Backup Node (vSRX Node 1)
ge-0/0/0	ICL to connect to peer node	10.0.1.10/24	10.0.1.11/24
ge-0/0/1	Untrust interface	• 10.0.2.110/24 (primary) • 10.0.2.11/24	• 10.0.2.20/24 (primary)
ge-0/0/2	Trust interface	• 10.0.3.10/24 (primary) • 10.0.3.12/24	• 10.0.3.20/24 (primary)

You need to select the vSRX Virtual Firewall image from Azure Marketplace and customize the vSRX Virtual Firewall VM deployment settings and dependencies based on your network requirements in Microsoft Azure Cloud. This deployment approach might be required for configuring Multinode High Availability on vSRX VMs. Note that this deployment scenario is outside of the use cases offered in the vSRX Virtual Firewall VM solution templates available from Juniper Networks.

Let's dive into the details of each step for deploying the vSRX Virtual Firewall on Microsoft Azure.

Configuration in Azure Portal

IN THIS SECTION

In this example, you deploy two virtual machines (VMs) for vSRX Firewall instance-vsrx3.0-node0 and vsrx3.0-node1. You must configure Resource Group, Virtual Networks, Public IP address, Network Security Groups for the VMs. Following tables provide the details of the resources used in this example.

Table 65: Resources Details in Azure for vSRX VMs

Name	Type
azure-vsrx3.0	Resource Group
azure-vsrx3.0-vnet	Virtual network
vsrx3.0-node0	Virtual machine for Node 0
vsrx3.0-node1	Virtual machine for Node 1
vsrx3.0-node0-ip	Public IP address for Node 0
vsrx3.0-node1-ip	Public IP address for Node 1
vsrx3.0-node0-nsg	Network security group for Node 0
vsrx3.0-node1-nsg	Network security group for Node 1
vsrx3.0-node172	Network Interface for Node 0
vsrx3.0-node719	Network Interface for Node 1

Table 66: Network Interface Details for vSRX VMs

Name	Primary Private IP
L3HA-ge-0	10.0.1.10

Table 66: Network Interface Details for vSRX VMs *(Continued)*

Name	Primary Private IP
L3HALink-node1	10.0.1.11
node0-ge-1	10.0.2.110
node0-ge-2	10.0.3.10
node1-ge-1	10.0.2.20
node1-ge-2	10.0.3.20

Create a Resource Group

A **resource group** is a logical container for resources deployed in Azure. It helps you manage and organize related resources. The vSRX VM is a resource within an Azure resource group. All components related to the vSRX (such as virtual network, storage account, public IP, etc.) are part of the same resource group.

1. Sign into the Azure portal.
2. Click on **Create a resource**.
3. Search for **Resource group** and create a new one.
4. Choose a name, enter subscription, and select region for the resource group.
5. Click **Review + create** and then **Create**. The following image shows a sample of Resource Group. See Create a Resource Group for details.

Assign IAM (Identity and Access Management) Role

IAM roles control access to Azure resources. Assigning roles ensures that only authorized users can manage specific resources. You must grant the "service principal" role to the user or user groups to manage IAM.

1. Navigate to your resource group.
2. Click on **Access control (IAM)**.
3. Add a new role assignment.
4. Select the desired role (Example: Service Principal Contributor, Owner, and so on).

5. Choose the user or user group to assign the role to.
6. Click **Save**.

Create a Storage Account

A **storage account** provides a unique name space to store and access data objects in Azure.

1. In your resource group, click on **Create (+)**.
2. Search for "Storage account" and create a new one.
3. Specify the name, deployment model, performance, and other settings.
4. Click **Review + create** and then **Create**. See Create a Storage Account for details.

Create a Virtual Network

A virtual network (VNet) is the foundation for your Azure infrastructure. It allows you to securely connect Azure resources.

1. In your resource group, click on **Create (+)**.
2. Search for **Virtual network** and create a new one.
3. Define the **name**, **address space**, and **subnet configuration**.
4. Click **Review + create** and then **Create**.
5. Click **Settings > Subnets**. The subnets are used to connect the two vSRX Virtual Firewall nodes using a logical connection (like the physical cable connecting ports).
   Following table shows a sample configuration used in this example.

   Table 67: Subnet Configuration in Azure Portal

Function	CIDR	Role
Management Subnet	10.0.3.0/24	Management traffic
ICL subnet	10.0.1.0/24	RTO, synchronization, and probes-related traffic
Untrust Subnet	10.0.2.0/24	External traffic
Trust Subnet	10.0.3.0/24	Internal traffic

See Create a Virtual Network for details.

Assign IAM Role

1. Enable permissions to use Azure API by navigating **Home > Managed Identities**.
2. Select your Resource Group and select **Azure role assignments** and click the role that you want to assign permissions.

   You need to enable the following permissions:

   - Microsoft.Authorization/*/read
   - Microsoft.Compute/virtualMachines/read
   - Microsoft.Network/networkSecurityGroups/join/action
   - Microsoft.Network/networkInterfaces/*
   - Microsoft.Network/virtualNetworks/join/action
   - Microsoft.Network/publicIPAddresses/read
   - Microsoft.Network/publicIPAddresses/write
   - Microsoft.Network/publicIPAddresses/join/action
   - Microsoft.Authorization/*/read
   - Microsoft.Compute/virtualMachines/read
   - Microsoft.Network/routeTables/*
   - Microsoft.Network/networkInterfaces/*

Now you are ready to deploy vSRX VMs.

Deploy vSRX VMs

IN THIS SECTION

Use the following steps to deploy two vSRX VM instances. You'll use these two instances to setup Multinode High Availability. For details, see Deploy the vSRX Virtual Firewall Image from Azure Marketplace.

1. Sign into the Azure portal using your Microsoft account credentials.
2. Search for vSRX in the Azure Marketplace by clicking on **Create a resource** and search for **vSRX Virtual Firewall**.
3. Select the vSRX VM image from the Azure Marketplace.
4. Configure deployment settings by providing the following details:
   a. Name for your vSRX VM.
   b. Resource Group. (You can create a new one or use an existing group). In this example, use the **azure-vsrx3.0** which you created in previously.

      NOTE: Deploy both vSRX Virtual Firewall instances in the same resource group. The resource group will hold all the resources associated with the vSRX Virtual Firewalls for this deployment.

   c. Subscription
   d. VM Disk Type
   e. Region where you want to deploy the VM.
   f. Authentication Type
   g. Username and password for VM access.
5. Configure the storage, networking, and monitoring settings for the vSRX Virtual Firewall VM. This includes specifying the storage account, virtual network, subnet, public IP address, network security group, VM extensions, availability set, and monitoring options.
6. Review your settings and click **Create**.

Azure will start provisioning the vSRX VM based on your configuration. After the vSRX Virtual Firewall VMs are created, the Azure portal dashboard lists the new vSRX Virtual Firewall VMs under Resource Groups.

NOTE: Remember to customize these steps based on your specific requirements and network design.

Complete the following configurations for the vSRX Virtual Firewall instances you just deployed in Azure portal:

Create Tags

On your vSRX VM page, select **Tags** from left-navigation bar.

Create tags in Azure for both VMs to identify the trust and untrust interfaces on two vSRX Virtual Firewall instances. The following tables shows sample tags used in this example.

Table 68: Interface Tags

Tag Name	Value
local_trust_interface	node0-ge-002
local_untrust_interface	node0-ge-001
peer_trust_interface	node1-ge-002
peer_untrust_interface	node1-ge-001

Note that the tag names mentioned in the table is for default configuration. We recommend to use the same tag names in the configuration.

Create a Public IP Address

In the Azure portal, go to the **Create a resource** section. Locate and select **Networking** and then go to **Public IP address**.

1. Click on **Create** to start setting up a new public IP address.
2. Configure IP address settings with following details:

- Name: Enter a unique name for the public IP resource.
- SKU: Choose between Basic and Standard offerings.
- IP Version: Select IPv4 or IPv6 based on your requirements.
- IP address assignment: Choose Static or Dynamic.
- Select or create a resource group where this IP will reside.
- Location: Choose the Azure region closest to your users.

3. Once configured, review the settings and then click **Create** to allocate the public IP address.

Create a Network Interface

Plan the network interface configuration on the vSRX Series firewalls on Azure.

1. Navigate to **Network interfaces** in the Azure portal under the **Networking** section.
2. Click on **Create network interface**.
3. Enter the following details for the new network interface:
   - Name: Provide a unique name for your NIC.
   - Region: Select the same region as your VNet, VMs, and IP addresses.
   - Virtual network: Choose the virtual network that you want your NIC to be associated with.
   - Subnet: Select the appropriate subnet.
4. Attach the public IP address you created earlier, if required.
5. Choose to create a new network security group or associate with an existing one.
6. Review the settings and click **Create** to provision the new NIC.

Create a Network Security Group (NSG)

1. Select **Network security groups** in the **Networking** category on the Azure portal.
2. Select **Create network security group**.
3. Set up your NSG with the following details:
   - Name: Create a name for your NSG.
   - Subscription: Verify you're working within the right subscription.

- Resource group: Select an existing one or create a new one.
- Location: Match it to the location of the resources you're protecting.

4. Add rules: After creation, define inbound and outbound security rules to control traffic to and from your NIC and VMs.
5. Go back to your NIC or subnet and associate it with the new NSG.
6. Check the configurations and then create the NSG.

Remember to define appropriate security rules for your NSG to manage the traffic flow as per your requirements.

Once you have created a network interface, a public IP address, and a network security group, you can proceed to attach the NIC to a virtual machine and the NSG to the NIC or a subnet. This complets the setup required for network connectivity and security for your Azure environment.

Set up Interfaces and IP Addresses

1. Navigate to your deployed vSRX VM and click **Settings > Networking**.
2. Locate the attached network interface.
3. Click the network interface name to open its details. In the IP configurations section, you'll find the assigned IP address (if any) and you can also configure IP address. For this example, use IP address configuration as mentioned in the following table:

Table 69: Configuration of Interfaces and IP Addresses on vSRX VMs

VMs	vSRX VM Node 0 (vsrx3.0-node0) (Active Node)			vSRX VM Node 1 (vsrx3.0-node1) (Backup Node)		
	Untrust Interface	Trust Interface	ICL	Untrust Interface	Trust Interface	ICL
Interfaces	ge-0/0/1	ge-0/0/2	ge-0/0/0	ge-0/0/1	ge-0/0/2	ge-0/0/0
Primary IP Address	10.0.2.110	10.0.3.10	10.0.1.10/24	10.0.2.20	10.0.3.20	10.0.1.11/24

Table 69: Configuration of Interfaces and IP Addresses on vSRX VMs *(Continued)*

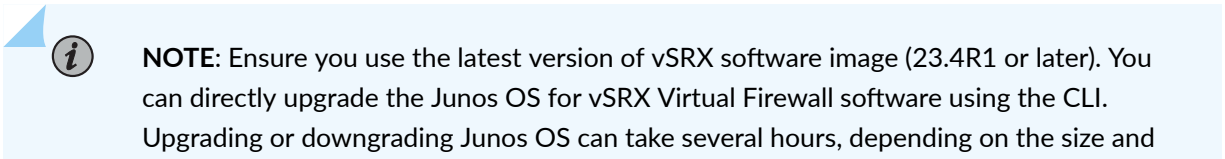
VMs	vSRX VM Node 0 (vsrx3.0-node0) (Active Node)			vSRX VM Node 1 (vsrx3.0-node1) (Backup Node)		
Secondary IP address (static IP address from subnet)	10.0.2.11	10.0.3.12	-	The node acting as backup node receives the same IP address when it transitions into active role.-	The node acting as backup node receives the same IP address when it transitions into active role.-	-
Associate public IP address to secondary to reach Internet	172.16.0.0	(Not Applicable in this example)	-	Note: The node acting as backup node receives the same IP address when it transitions into active role.	(Not Applicable in this example)	-

Ensure you enable IP forwarding on control link interface and configure default routes on both trust and untrust sides.

Click **Settings > Networking** to display interfaces, subnet, and IP configurations of your VM.

After vSRX Virtual Firewall deployment is completed, the vSRX Virtual Firewall VM is automatically powered on and launched. At this point you can use an SSH client to log in to the vSRX Virtual Firewall VM.

Now that all configurations required on Azure portal are complete, let's start configuration on vSRX Virtual Firewall using CLI.



NOTE: Ensure you use the latest version of vSRX software image (23.4R1 or later). You can directly upgrade the Junos OS for vSRX Virtual Firewall software using the CLI. Upgrading or downgrading Junos OS can take several hours, depending on the size and

configuration of the network. You download the desired Junos OS Release for **vSRX Virtual Firewall.tgz** file from the Juniper Networks website. See Migration, Upgrade, and Downgrade Instructions.

Configure vSRX Virtual Firewalls

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

- "Set Commands on all Devices" on page 1127
- "Show Configuration Output" on page 1130

Junos IKE package is recommended on your SRX Series Firewalls for Multinode High Availability configuration. This package is available as a default package or as an optional package on SRX Series Firewalls. See Support for Junos IKE Package for details.

If the package is not installed by default on your SRX Series firewall, use the following command to install it. 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 ...

......
Restart cli using the new version ? [yes,no] (yes)
```

1. Configure interfaces for ICL, internal and external traffic.
   - Node 0

```
[edit]
user@srx-00# set interfaces ge-0/0/0 unit 0 family inet address 10.0.1.10/24
user@srx-00# set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.110/24 primary
user@srx-00# set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.11/24
user@srx-00# set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.10/24 primary
user@srx-00# set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.12/24
```

```
user@srx-00# set interfaces lo0 description HA_LOOPBACK
user@srx-00# set interfaces lo0 unit 0 family inet address 10.11.1.10/32 primary
```

- Node 1

```
[edit]
user@srx-01# set interfaces ge-0/0/0 unit 0 family inet address 10.0.1.11/24
user@srx-01# set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.20/24 primary
user@srx-01# set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.11/24
user@srx-01# set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.20/24 primary
user@srx-01# set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.12/24
user@srx-01# set interfaces lo0 description HA_LOOPBACK
user@srx-01# set interfaces lo0 unit 0 family inet address 10.11.1.11/32 primary
```

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

- Node 0

```
[edit]
user@srx-00# set security zones security-zone icl host-inbound-traffic system-services all
user@srx-00# set security zones security-zone icl host-inbound-traffic protocols all
user@srx-00# set security zones security-zone icl interfaces ge-0/0/0.0
user@srx-00# set security zones security-zone untrust host-inbound-traffic system-services
ike
user@srx-00# set security zones security-zone untrust host-inbound-traffic system-services
ping
user@srx-00# set security zones security-zone untrust host-inbound-traffic protocols bfd
user@srx-00# set security zones security-zone untrust host-inbound-traffic protocols bgp
user@srx-00# set security zones security-zone untrust interfaces ge-0/0/1.0
user@srx-00# set security zones security-zone trust host-inbound-traffic system-services
all
user@srx-00# set security zones security-zone trust host-inbound-traffic protocols all
user@srx-00# set security zones security-zone trust interfaces ge-0/0/2.0
```

- Node 1

```
[edit]
user@srx-01# set security zones security-zone icl host-inbound-traffic system-services all
user@srx-01# set security zones security-zone icl host-inbound-traffic protocols all
user@srx-01# set security zones security-zone icl interfaces ge-0/0/0.0
```

```
user@srx-01# set security zones security-zone untrust host-inbound-traffic system-services
ike
user@srx-01# set security zones security-zone untrust host-inbound-traffic system-services
ping
user@srx-01# set security zones security-zone untrust host-inbound-traffic protocols bfd
user@srx-01# set security zones security-zone untrust host-inbound-traffic protocols bgp
user@srx-01# set security zones security-zone untrust interfaces ge-0/0/1.0
user@srx-01# set security zones security-zone trust host-inbound-traffic system-services
all
user@srx-01# set security zones security-zone trust host-inbound-traffic protocols all
user@srx-01# set security zones security-zone trust interfaces ge-0/0/2.0
```

3. Configure local node and peer node details.

   - Node 0

```
[edit]
user@srx-00# set chassis high-availability local-id 1
user@srx-00# set chassis high-availability local-id local-ip 10.0.1.10
```

   - Node 1

```
[edit]
user@srx-01# set chassis high-availability local-id 2
user@srx-01# set chassis high-availability local-id local-ip 10.0.1.11
```

4. Associate the interface to the peer node for interface monitoring, and configure the liveness detection details.

   - Node 0

```
[edit]
user@srx-00# set chassis high-availability peer-id 2 peer-ip 10.0.1.11
user@srx-00# set chassis high-availability peer-id 2 interface ge-0/0/0.0
user@srx-00# set chassis high-availability peer-id 2 liveness-detection minimum-interval
400
user@srx-00# set chassis high-availability peer-id 2 liveness-detection multiplier 5
```

- Node 1

```
[edit]
user@srx-01# set chassis high-availability peer-id 1 peer-ip 10.0.1.10
user@srx-01# set chassis high-availability peer-id 1 interface ge-0/0/0.0
user@srx-01# set chassis high-availability peer-id 1 liveness-detection minimum-interval
400
user@srx-01# set chassis high-availability peer-id 1 liveness-detection multiplier 5
```

5. Configure SRG1 with deployment type as cloud, assign an ID, and set preemption and activeness priority.

   - Node 0

```
[edit]
user@srx-00# set chassis high-availability services-redundancy-group 1 mode active-backup
user@srx-00# set chassis high-availability services-redundancy-group 1 deployment-type
cloud
user@srx-00# set chassis high-availability services-redundancy-group 1 peer-id 2
user@srx-00# set chassis high-availability services-redundancy-group 1 prefix-list pref1
routing-instance s1-router
user@srx-00# set chassis high-availability services-redundancy-group 1 managed-services
ipsec
user@srx-00# set chassis high-availability services-redundancy-group 1 activeness-priority
200
```

   - Node 1

```
[edit]
user@srx-01# set chassis high-availability services-redundancy-group 1 mode active-backup
user@srx-01# set chassis high-availability services-redundancy-group 1 deployment-type
cloud
user@srx-01# set chassis high-availability services-redundancy-group 1 peer-id 1
user@srx-01# set chassis high-availability services-redundancy-group 1 prefix-list pref1
routing-instance s1-router
user@srx-01# set chassis high-availability services-redundancy-group 1 managed-services
ipsec
user@srx-01# set chassis high-availability services-redundancy-group 1 activeness-priority
100
```

6. Configure Azure deployment-related options.

- Node 0

```
[edit]
user@srx-00# set security cloud high-availability azure peer-liveliness probe-ip 10.0.2.20
user@srx-00# set security cloud high-availability azure peer-liveliness probe-ip source-ip
10.0.2.110
user@srx-00# set security cloud high-availability azure peer-liveliness probe-ip routing-
instance s1-router
```

- Node 1

```
[edit]
user@srx-01# set security cloud high-availability azure peer-liveliness probe-ip 10.0.2.110
user@srx-01# set security cloud high-availability azure peer-liveliness probe-ip source-ip
10.0.2.20
user@srx-01# set security cloud high-availability azure peer-liveliness probe-ip routing-
instance s1-router
```

7. Configure the security policy.

   Node 0 and Node 1

```
[edit]
user@srx-01# set security policies default-policy permit-all
```

NOTE: The security policy shown in this example is only for demonstration. You should configure security policies as per your network needs. Ensure that your security policies allow only the applications, users, and devices that you trust.

8. Configure routing instance.

- Node 0

```
[edit]
user@srx-00# set routing-instances s1-router instance-type virtual-router
user@srx-00# set routing-instances s1-router routing-options static route 0.0.0.0/0 next-
hop 10.0.2.1
```

```
user@srx-00# set routing-instances s1-router interface ge-0/0/1.0
user@srx-00# set routing-instances s1-router interface ge-0/0/2.0
```

- Node 1

```
[edit]
user@srx-01# set routing-instances s1-router instance-type virtual-router
user@srx-01# set routing-instances s1-router routing-options static route 0.0.0.0/0 next-
hop 10.0.2.1

user@srx-01# set routing-instances s1-router interface ge-0/0/1.0
user@srx-01# set routing-instances s1-router interface ge-0/0/2.0
```

9. Configure policy options.

- Node 0

```
[edit]
user@srx-00# set policy-options prefix-list pref1 10.0.2.0/24
```

- Node 1

```
[edit]
user@srx-01# set policy-options prefix-list pref1 10.0.2.0/24
```

For encrypting the ICL, use the following sample configuration:

```
user@host# set chassis high-availability peer-id <peer-id> vpn-profile IPSEC_VPN_ICL
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
```

```
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
```

See Example: Configure Multinode High Availability in a Layer 3 Network for details.

Verification

IN THIS SECTION

Use the show commands to confirm that the configuration is working properly.

Table 70: Show Commands for Verification

Command	Verification Task
`show chassis high-availability information`	Display details of the Multinode High Availability status on your security device including health status of the peer node.
`show security cloud high-availability information`	Display status about Multinode High Availability deployment on public cloud (AWS or Azure).

Check Multinode High Availability Details

Purpose

View and verify the details of the Multinode High Availability setup configured on your vSRX Virtual Firewall instance.

Action

From operational mode, run the following command on both the devices.

On Node 0 (Active Node)

```
user@srx-00> 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: 10.0.1.10
HA Peer Information:

    Peer Id: 2        IP address: 10.0.1.11     Interface: ge-0/0/0.0
    Routing Instance: default
    Encrypted: NO     Conn State: UP
    Configured BFD Detection Time: 5 * 400ms
    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: DISABLED
```

```
        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 Node 1 (Backup Node)

```
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: 2
Local-IP: 10.0.1.11
HA Peer Information:

    Peer Id: 1        IP address: 10.0.1.10     Interface: ge-0/0/0.0
    Routing Instance: default
    Encrypted: NO     Conn State: UP
    Configured BFD Detection Time: 5 * 400ms
    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: CLOUD
        Status: BACKUP
        Activeness Priority: 100
        Preemption: DISABLED
        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 Azure

Purpose

Check the status of Multinode High Availability deployment in Azure Cloud.

Action

From operational mode, run the following command:

```
user@srx-00> show security cloud high-availability information

Cloud HA Information:

    Cloud Type      Cloud Service Type   Cloud Service Status
    AZURE           Secondary IP         Bind to Peer Node
```

Meaning

Verify these details from the command output:

- The field Cloud Type: Azure indicates the deployment is for Azure.
- The field Cloud Service Type: Secondary IP indicates that the Azure deployment uses the secondary IP to control traffic.
- The field Cloud Service Status: Bind to Peer Node indicates the binding of the secondary IP address to the peer node meaning the current node is backup node.

Basic Troubleshooting Checklist

1. Check secondary IP for untrust interface and trust interface are on the same vsrx3.0 VM instance.
2. Check the four tag values to match the interface names.
3. Check inbound rule is correct to permit the traffic.
4. Check IP forwarding is enabled in Azure portal.
5. Check Azure portal route and vSRX CLI route are synced.
6. Check untrust interface of the active node to see if the floating IP addresses attached to it in Azure portal.

Set Commands on all Devices

IN THIS SECTION

vSRX Virtual Firewall (Node 0)

```
set chassis high-availability local-id 1
set chassis high-availability local-id local-ip 10.0.1.10
set chassis high-availability peer-id 2 peer-ip 10.0.1.11
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 mode active-backup
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 prefix-list pref1 routing-instance s1-
router
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 activeness-priority 200
set security policies default-policy permit-all
set security zones security-zone icl host-inbound-traffic system-services all
set security zones security-zone icl host-inbound-traffic protocols all
set security zones security-zone icl interfaces ge-0/0/0.0
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-0/0/1.0
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 azure peer-liveliness probe-ip 10.0.2.20
set security cloud high-availability azure peer-liveliness probe-ip source-ip 10.0.2.110
set security cloud high-availability azure peer-liveliness probe-ip routing-instance s1-router
set interfaces ge-0/0/0 unit 0 family inet address 10.0.1.10/24
set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.110/24 primary
set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.11/24
set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.10/24 primary
set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.12/24
set interfaces lo0 description HA_LOOPBACK
set interfaces lo0 unit 0 family inet address 10.11.1.10/32 primary
set policy-options prefix-list pref1 10.0.2.0/24
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 10.0.2.1
```

```
set routing-instances s1-router interface ge-0/0/1.0
set routing-instances s1-router interface ge-0/0/2.0
```

vSRX Virtual Firewall (Node 1)

```
set chassis high-availability local-id 2
set chassis high-availability local-id local-ip 10.0.1.11
set chassis high-availability peer-id 1 peer-ip 10.0.1.10
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 mode active-backup
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 prefix-list pref1 routing-instance s1-
router
set chassis high-availability services-redundancy-group 1 managed-services ipsec
set chassis high-availability services-redundancy-group 1 activeness-priority 100
set security policies default-policy permit-all
set security zones security-zone icl host-inbound-traffic system-services all
set security zones security-zone icl host-inbound-traffic protocols all
set security zones security-zone icl interfaces ge-0/0/0.0
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-0/0/1.0
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 azure peer-liveliness probe-ip 10.0.2.110
set security cloud high-availability azure peer-liveliness probe-ip source-ip 10.0.2.20
set security cloud high-availability azure peer-liveliness probe-ip routing-instance s1-router
set interfaces ge-0/0/0 unit 0 family inet address 10.0.1.11/24
set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.20/24 primary
set interfaces ge-0/0/1 unit 0 family inet address 10.0.2.11/24
set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.20/24 primary
set interfaces ge-0/0/2 unit 0 family inet address 10.0.3.12/24
set interfaces lo0 description HA_LOOPBACK
set interfaces lo0 unit 0 family inet address 10.11.1.11/32 primary
set policy-options prefix-list pref1 10.0.2.0/24
```

```
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 10.0.2.1

set routing-instances s1-router interface ge-0/0/1.0
set routing-instances s1-router interface ge-0/0/2.0
```

Show Configuration Output

IN THIS SECTION

From configuration mode, confirm your configuration by entering the `show high availability`, `show security zones`, and `show interfaces` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

vSRX Virtual Firewall (Node 0)

```
user@srx-00# show chassis high-availability
local-id {
    1;
    local-ip 10.0.1.10;
}
peer-id 2 {
    peer-ip 10.0.1.11;
    interface ge-0/0/0.0;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 1 {
```

```
    mode active-backup;
    deployment-type cloud;
    peer-id {
        2;
    }
    prefix-list pref1 {
        routing-instance s1-router;
    }
    managed-services ipsec;
    activeness-priority 200;
}
```

```
user@srx-00# show security zones
security-zone icl {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/1.0;
```

```
    }
}

security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
```

```
user@srx-00# show security cloud
high-availability {
    azure {
        peer-liveliness {
            probe-ip 10.0.2.20 source-ip 10.0.2.110 routing-instance s1-router;
        }
    }
}
```

```
user@srx-00# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.0.1.10/24;
        }
    }
}
ge-0/0/1 {
```

```
    unit 0 {
        family inet {
            address 10.0.2.110/24 {
                primary;
            }
            address 10.0.2.11/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.0.3.10/24 {
                primary;
            }
            address 10.0.3.12/24;
        }
    }
}
lo0 {
    description HA_LOOPBACK;
    unit 0 {
        family inet {
            address 10.11.1.10/32 {
                primary;
            }
        }
    }
}
```

```
user@srx-00# show routing-instances
s1-router {
    instance-type virtual-router;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.0.2.1;
        }

    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
}
```

vSRX Virtual Firewall (Node 1)

```
user@srx-01# show chassis high-availability
local-id {
    2;
    local-ip 10.0.1.11;
}
peer-id 1 {
    peer-ip 10.0.1.10;
    interface ge-0/0/0.0;
    liveness-detection {
        minimum-interval 400;
        multiplier 5;
    }
}
services-redundancy-group 1 {
    mode active-backup;
    deployment-type cloud;
    peer-id {
        1;
    }
    prefix-list pref1 {
        routing-instance s1-router;
    }
    managed-services ipsec;
    activeness-priority 100;
}
```

```
user@srx-01# show security zones
 security-zone icl {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
```

```
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            ike;
            ping;
        }
        protocols {
            bfd;
            bgp;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/2.0;
    }
}
```

```
user@srx-01# show security cloud
 high-availability {
    azure {
        peer-liveliness {
            probe-ip 10.0.2.110 source-ip 10.0.2.20 routing-instance s1-router;
        }
```

```
    }
}
```

```
user@srx-01# show interfacesge-0/0/0 {
    unit 0 {
        family inet {
            address 10.0.1.11/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.0.2.20/24 {
                primary;
            }
            address 10.0.2.11/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.0.3.20/24 {
                primary;
            }
            address 10.0.3.12/24;
        }
    }
}
lo0 {
    description HA_LOOPBACK;
    unit 0 {
        family inet {
            address 10.11.1.11/32 {
                primary;
            }
        }
```

```
    }
}
```

```
user@srx-01# show routing-instances
s1-router {
    instance-type virtual-router;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.0.2.1;
        }

    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
}
```

Change History Table

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

Release	Description
24.4R1	Starting in Junos OS Release 24.4R1, we support IPsec VPN for active/backup Multinode High Availability in Azure Cloud deployments.
23.4R1	Starting in Junos OS Release 23.4R1, we support MNHA in Microsoft Azure Cloud platform.

Multinode High Availability in Google Cloud Platform

IN THIS SECTION

Understanding High Availability on Google Cloud Platform

IN THIS SECTION

You can configure Multinode High Availability on vSRX Virtual Firewalls deployed in the Google Cloud Platform (GCP) Marketplace. You can use the GCP Marketplace to set up your vSRX as a VM running on a Google Compute Engine instance. You can configure a pair of vSRX virtual firewalls on GCP to operate as in an active/backup Multinode high availability configuration. Participating nodes run both active control and data planes at the same time. The nodes backup each other to ensure a fast synchronized failover in case of a system or hardware failure. The interchassis link (ICL) connection between the two devices synchronizes and maintains the state information and handles device failover scenarios.

Overview

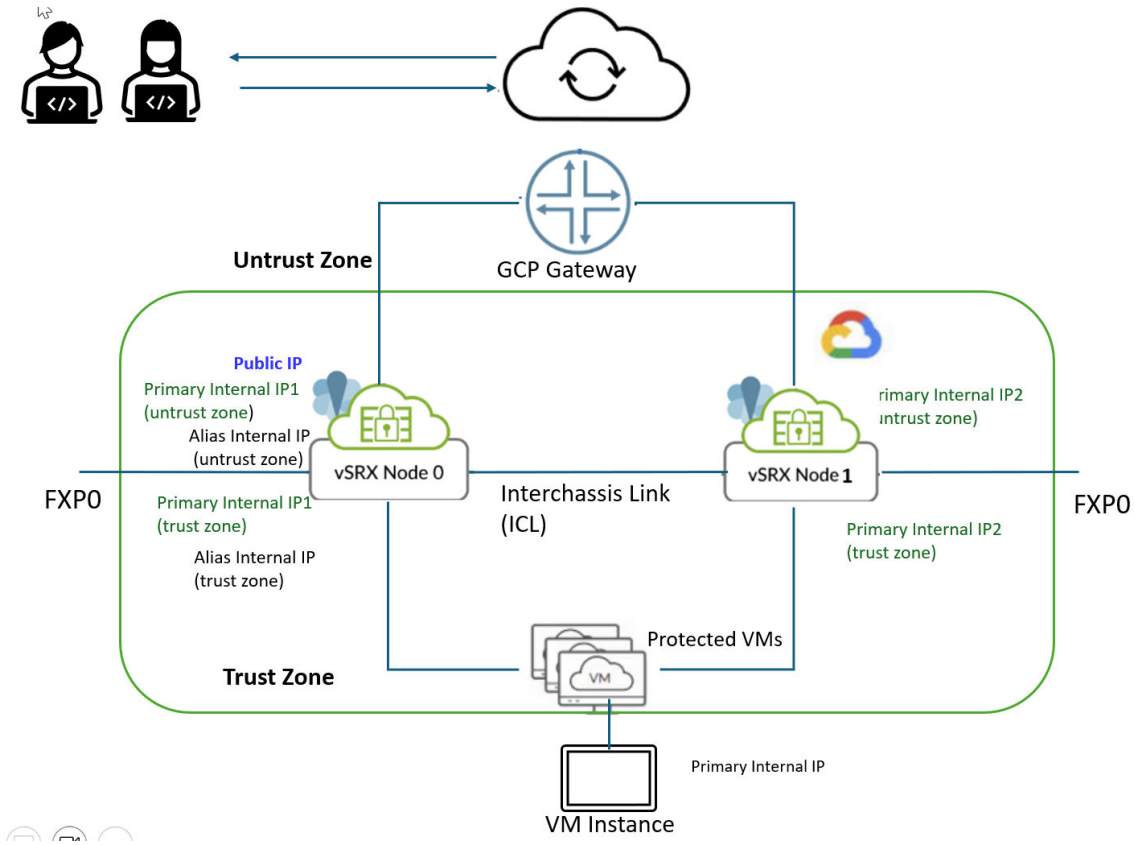
Multinode High Availability on Google Cloud Platform (GCP) is designed to ensure that your network traffic and security services remain uninterrupted by enabling two instances to function as a single device in an active/backup configuration. This setup synchronizes both configuration and stateful session information between instances, ensuring seamless failover when an instance fails. When a failure occurs, the system automatically triggers a failover, redirecting traffic to the healthy node to maintain continuous service availability.

About GCP Alias Internal IP Addresses

- GCP permits each NIC interface to have a single public external IP which can be transferred between instances.
- Google Cloud alias IP ranges let you assign internal IP addresses as aliases to a VM's network interfaces. This is useful when running multiple services on a VM and needing distinct IP addresses for each. Alias IP ranges can come from a subnet's primary or secondary CIDR ranges and are routed automatically within the virtual network.
- Incoming packets don't contain the GCP alias internal IP, but it can be utilized at the egress interface for outgoing packets.
- The GCP NIC alias internal IP is used in Multinode High Availability setup for session installation, not the primary internal IP.

Following figure shows two vSRX Virtual Firewall instances form a Multinode high availability pair in the in the GCP. One vSRX Virtual Firewall instance acts as the active node and the other instance acts as the backup node. Both nodes connect to each other using an ICL to synchronize and maintain state information and to handle device failover scenarios.

Figure 76: vSRX in Multinode High Availability Deployments in GCP



As shown in the illustration, each node has assigned one private IP address for both untrust (Internet) interface and trust (protected) interface. These IP addresses are referred as private untrust and private trust IPs in the illustration.

Active node assigns public IP address and alias internal IP address for untrust interface and assign alias internal IP address for trust interface.

When high-availability failover happens, the system moves:

- The public IP and alias internal IP of the untrust interface from one node to another node.
- The alias internal IP address of the trust interface from one node to another node.

The Cloud HA manager process and GCP SDK APIs manage the setup, synchronizing session states and configurations between two instances to provide seamless failover. The Cloud HA manager process

manages state enforcement and split-brain prevention, the use of alias IPs to maintain traffic continuity, and the provision of Terraform templates to automate deployment.

IPsec VPN Support

IPsec VPN support is available for active/backup Multinode High Availability in GCP deployments. IPsec VPN tunnels are secure, encrypted connection between different networks or endpoints. In the Multinode High Availability setup, the system establishes secure tunnels between nodes in high availability setup and VPN peer devices. The IPsec VPN tunnel anchors at an active SRG1, which manages their termination and failover actions.

Limitation

Multinode High Availability doesn't support multiple SRG configurations (active/active) in public cloud deployments. Active/backup mode supports SRG0 or SRG1.

Configuration Options

To create Multinode High Availability setup on vSRX instances in GCP, you can use CLI configuration or automate using Terraform templates.

By leveraging Terraform templates, you can automate the deployment process, significantly reducing the time and complexity involved in setting up the HA solution on GCP. These templates guide you through creating VPCs, spinning up instances, assigning IPs, and configuring HA, streamlining the deployment process.

Change History Table

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

Release	Description
24.4R1	Starting in Junos OS Release 24.4R1, we support MNHA in Google Cloud Platform (GCP) platform.

Multinode High Availability Monitoring Options

IN THIS SECTION

Monitoring Types

IN THIS SECTION

A high availability failure detection monitors both system, software, and hardware for internal failures. The system can also monitor network connectivity problems or link connectivity using interface monitoring, BFD path monitoring and IP monitoring to detect reachability of targets further away.

Table 71 on page 1142 provides details on different monitoring types used in Multinode High Availability.

Table 71: Multinode High Availability Monitoring Types

Montitoring Type	What is Does	Detection Type	Scope
BFD Monitoring	Monitors reachability to the next hop by examining the link layer along with the actual link.	• Path failures • Link failures	• Detects failure within its routing connectivity • Not intended to detect failures beyond direct connections/ next-hops.
IP monitoring	Monitors the connectivity to hosts or services located beyond directly connected interfaces or next-hops.	• Path failures • Link failures	• Detects failure occurring at more distant hosts or services. • Not intended for detecting failures occurring in directly connected links or next-hop failures.
Interface monitoring	Examines whether the link layer is operational or not.	Link failures	• Detects failure in directly connected links or next-hops, and connectivity to hosts or services located farther away. • Not intended for monitoring path

In Multinode High Availability, when monitoring detects a connectivity failure to a host or service, it marks the affected path as down/unavailable, and marks the corresponding Service Route Groups (SRGs) at the impacted node as Ineligible. The affected SRGs will transition in a stateful manner to the other node without causing any disruption to traffic.

To prevent any traffic from being lost, Multinode High Availability takes following precautions:

- Layer 3 mode—Routes will be redrawn so that the traffic is redirected correctly

- Default gateway or hybrid mode—The new active node for the SRG sends a GARP (Gratuitous ARP) to the connected switch to ensure the re-routing of traffic

Multinode High Availability Failure Scenarios

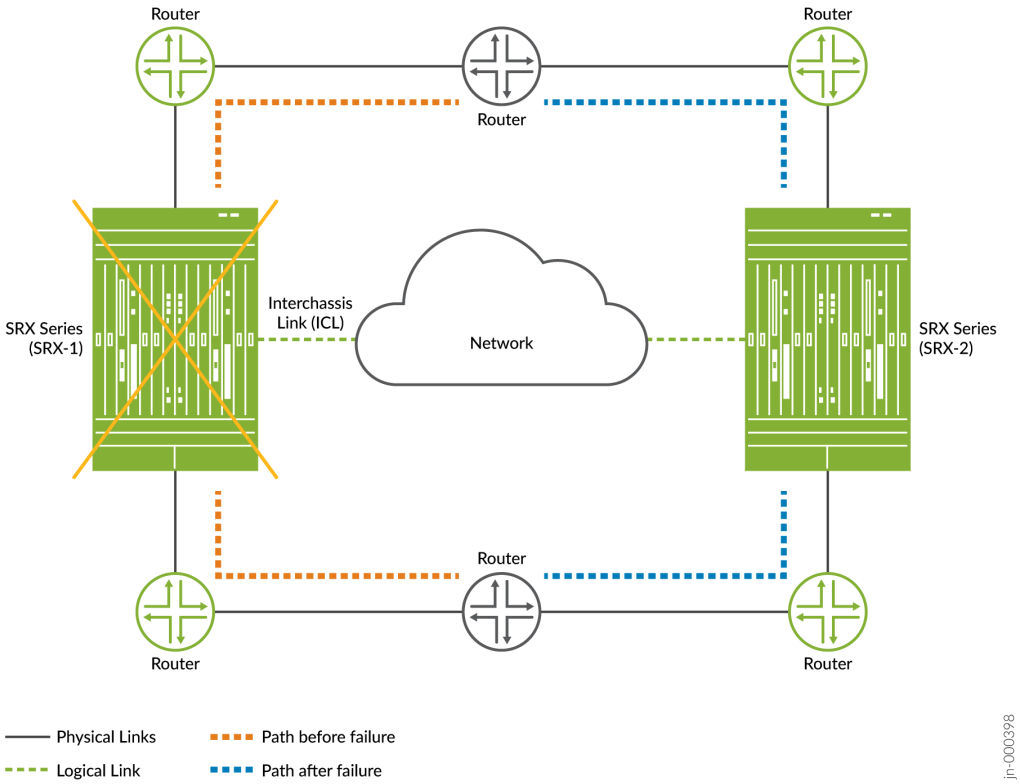
The following sections describe possible failure scenarios: how a failure is detected, what recovery action to take, and if applicable, the impact on the system caused by the failure.

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 the node with hardware failure.
- **Impact** —Traffic will failover to the other node (if healthy) as shown in Figure 77 on page 1144. .

Figure 77: Hardware Failure in Multinode High Availability



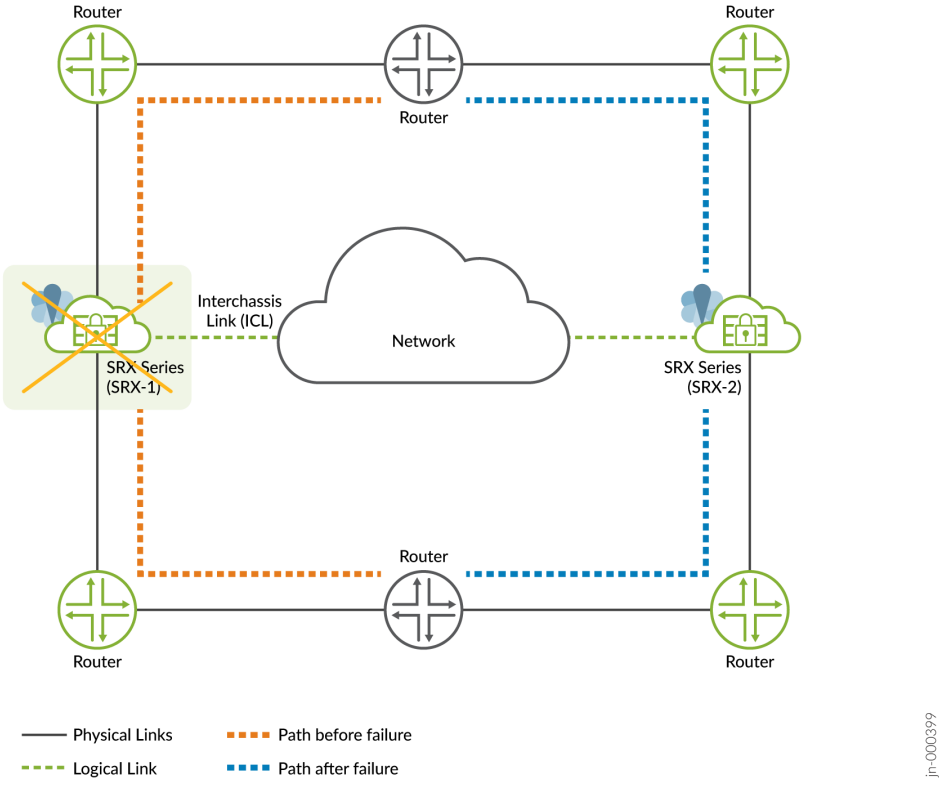
- **Recovery**—Recovery of failure takes place when you clear chassis hardware failure (ex: replace or repair the failed hardware component.
- **Results**—Check status using the following commands:
  - *show chassis high-availability information detail*
  - `show chassis hardware`
  - `show chassis alarms`

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
  - Changes system state to `INELIGIBLE` on the affected node with system/software failure.

- **Impact** —Traffic will failover to the other node if healthy as shown in Figure 78 on page 1145

Figure 78: 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* command.

Network/Connectivity Failure

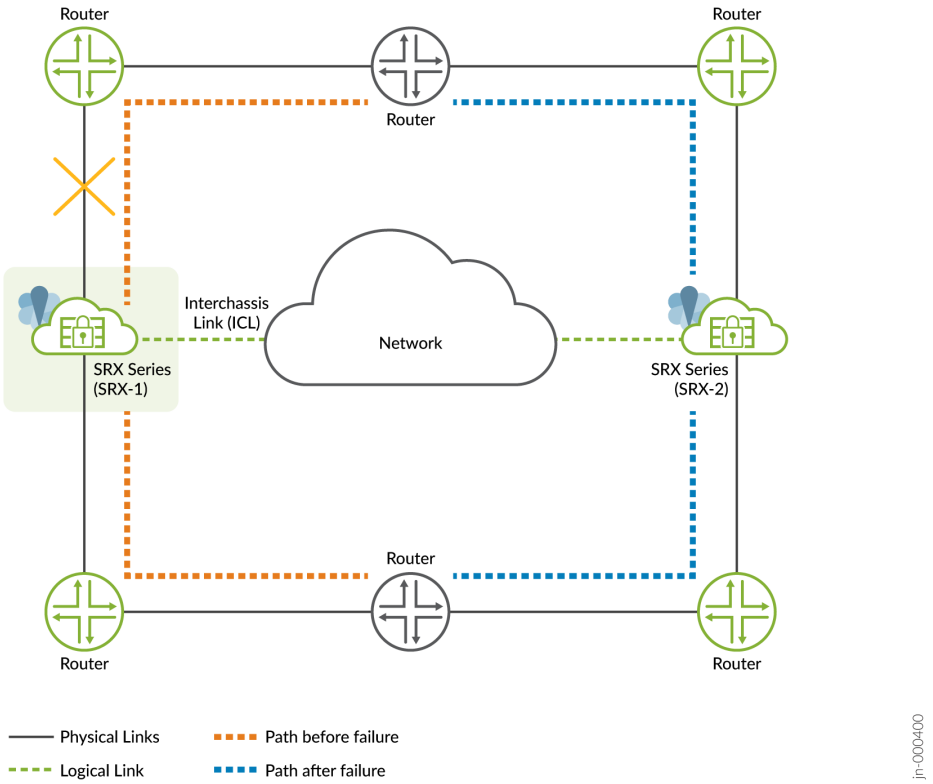
Physical Interfaces (Link) Failure

- **Cause**—A failure in interfaces could be due to network equipment 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 with network or connectivity failure (if the interface-monitor is configured). Path connectivity could also be detected with BFD or IP-monitoring and trigger an event based on configured action.

- **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 79 on page 1146.

Figure 79: 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 "Example: Configure Multinode High Availability in a Layer 3 Network" on page 698 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 in MNHA, see "Example: Configure Multinode High Availability in a Layer 3 Network" on page 698. For troubleshooting interfaces, see Troubleshooting Interfaces.

Interchassis Link (ICL) Failure

- **Cause**—A failure in ICL could be due to network outages, or inconsistent configurations.

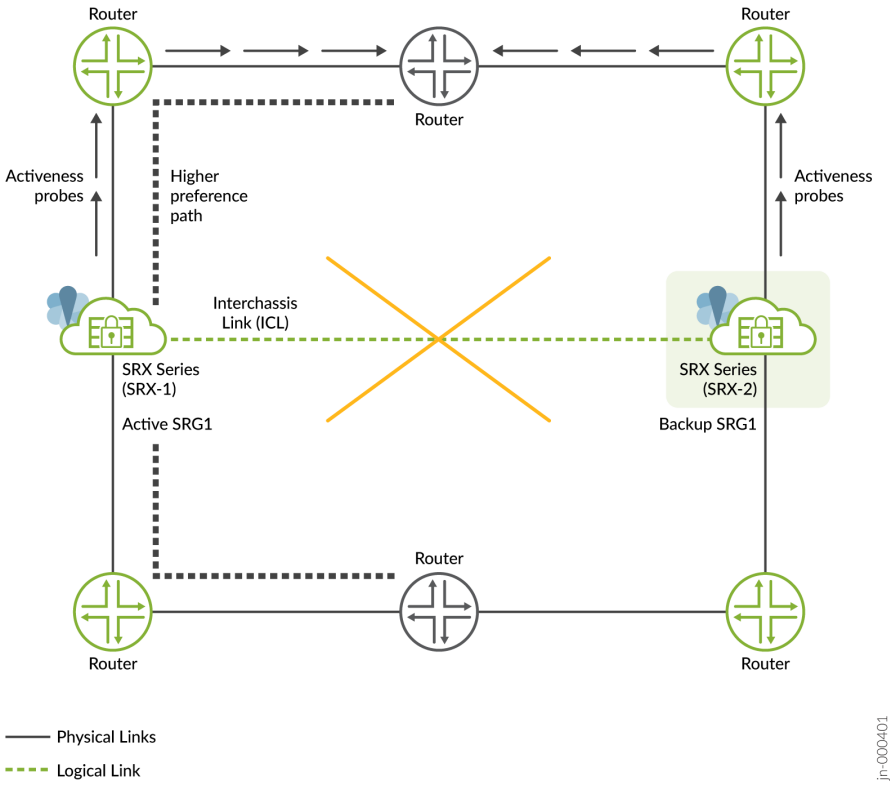
- **Detection**— In Multinode High Availability, nodes cannot reach each other and they initiate a activeness determination probe (ICMP probe).

- **Impact**— In a Multinode High Availability system, ICL connects active and backup nodes; if the ICL goes down, both devices will notice this change and start the activeness probe (ICMP probe). Activeness probe is done to determine the node that can take active role for each SRG1+. Based on the probe result, one of the node transitions to the active state.

  As shown in Figure 80 on page 1148, 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 80: 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.

Node Remains in Isolated State

- **Cause**—In a Multinode High Availability setup, the node remains in isolated state after a reboot and associated interfaces continue to remain down when:
    - Inter chassis link (ICL) has no connectivity to the other node after booting up until the cold-sync complete

      and

    - The `shutdown-on-failure` option is configured on SRG0

      NOTE: The above cause could also happen if the other device is out of service.

- **Detection**—SRG0 status displayed as `ISOLATED` in command output.
- **Recovery**—The node automatically recovers when the other node comes online and the ICL can exchange system information or when you remove the `shutdown-on-failure` statement and commit the configuration.

  Use the `delete chassis high-availability services-redundancy-group 0 shutdown-on-failure` to remove the statement.

  If the above solution is not suitable for your environment, you can use the `install-on-failure-route` option. In this option, the Multinode High Availability setup uses a defined signal route for more graceful handling of the above situation using routing policy options, which is similar to active-signal-route and backup-signal-route approach available in SRG1+.

Flexible Path Monitoring

IN THIS SECTION

Starting in Junos OS Release 23.4R1, we have added new enhancements for the following existing path monitoring features:

- IP monitoring
- BFD monitoring
- Interface monitoring

The enhancements add more granular control for the path monitoring feature by:

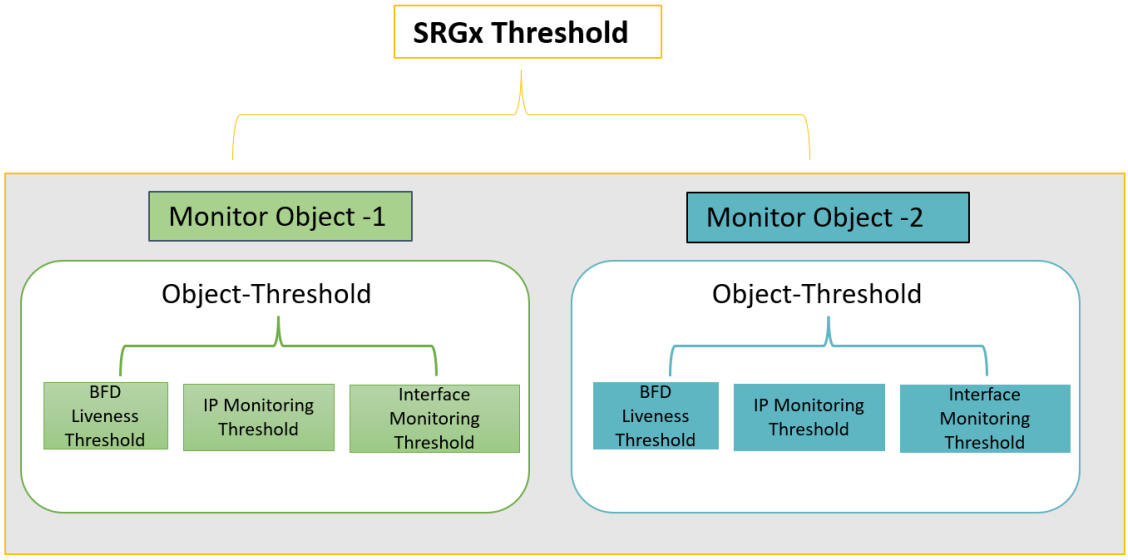
- Extension of monitoring for SRG0 in addition to SRG1+
- Grouping of monitoring functions
- Support monitoring based on the direction associated with an service redundancy group (SRG) path
- Adding weights associated with each monitoring functions

By grouping related functions together, the system can process them as a unit, which can lead to more efficient computation and resource utilization.

SRG Monitoring Objects

Lets understand the concept of monitoring objects with the following illustration.

Figure 81: SRG Monitoring Objects



You can configure the monitoring options on a per-service-redundancy-group basis. That is, if specific items in the SRG were to fail, that SRG can failover to the other node. Each SRG includes one or more monitoring objects.

The monitoring features available in monitoring objects are—BFD liveness, interface monitoring, and IP monitoring. Each of these feature has an associated threshold value and weight attributes.

Within a monitor-object, whenever the particular object fails to trigger a failover as result of IP/interface/BFD monitoring, the system considers the event as monitoring failure. The software adds the count based on the weight of the failed object.

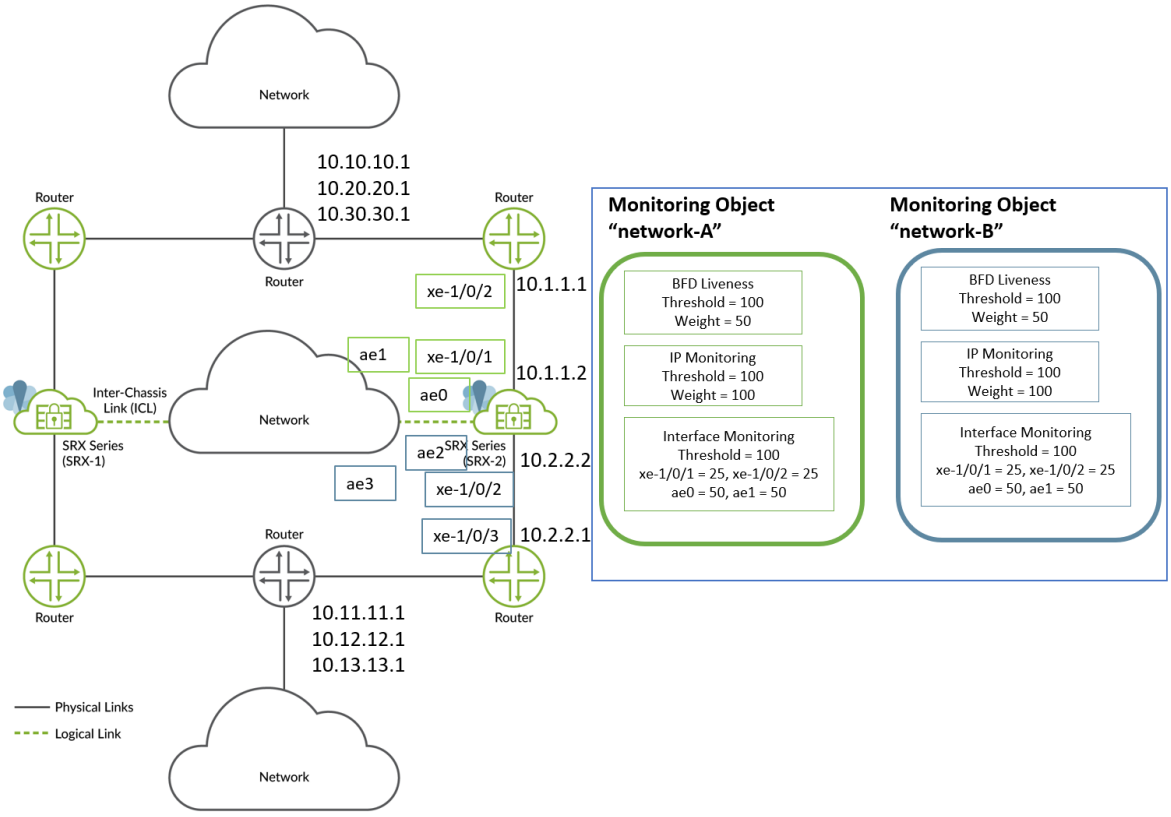
When the count exceeds the threshold value of IP/interface/BFD, the system adds the count to parent monitoring-object's threshold value.

When the sum of the thresholds of all monitoring-objects bound to the SRG is equal to or greater than the threshold value configured on the SRG, the system triggers a monitor failure for that SRG. SRG fails over to the other node.

Path Monitoring Configuration

Lets consider the following example for the topology shown in Figure 82 on page 1152. In this setup, we are configuring path monitoring options for SRG1 on Node 2 device.

Figure 82: Path Monitoring Configuration Sample



In this example, to configure path monitoring options:

- Use an aggregated Ethernet interface (ae) for inter-chassis link (ICL) and use xe-1/0/x interfaces for connecting to neighboring routers.
- Create two monitor-objects "**network-A**" and "**network-B**". Both the **network-A** and **network-B** monitor-objects include all IP addresses and interfaces configured between the SRX Series device and neighboring routers.
- Configure BFD to monitor the neighboring routes.
- Configure IP monitoring to monitor the routes not directly connected to SRG1.
- Configure interface monitoring on directly connected links or next-hops.

The following table shows sample weights and threshold assignments.

Table 72: Weights and Threshold for Monitor-Objects (Example)

Monitor Objects	BFD		IP		Interface		Monitor-Object Threshold	SRG Threshold
	Threshold	**Weight**	**Threshold**	**Weight**	**Threshold**	**Weight**		
network-A	100	50	100	50 (10.10.10.1, 10.20.20.1, 10.30.30.1)	100	25 (xe-1/0/1 and xe-1/0/2) 50 (ae0 and ae1)	100	100
network-B	100	50	100	50 (10.11.11.1, 10.12.12.1, 10.13.13.1)	100	25 (xe-1/0/3 and xe-1/0/4) 50 (ae2 and ae3)	200	

NOTE:

- You can configure up to 10 monitoring objects per SRG.
- You can configure SRG monitoring as in Junos OS 23.4 (with SRG threshold and monitoring-objects) or configure monitoring options as supported before Junos OS Release 23.4R1. Combining both the styles of configuration is not supported.
- Configuring monitor-objects is same as on SRG 0 and SRG1+.

Configuration Samples:

In the following configuration snippet, the service redundancy group (SRGx) includes two monitor-objects—**network-A** and **network-B**. Each of these monitoring objects have IP monitoring, interface monitoring, and BFD detection configured with respective weights and threshold values.

- Set SRG threshold value.

```
set chassis high-availability services-redundancy-group x monitor srg-threshold 100
```

- Configure monitor-object `network-A`.
  - Set monitor object threshold value.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
object-threshold 100
```

  - Configure BFD monitoring options.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
bfd-liveliness threshold 100
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
bfd-liveliness dst-ip 10.1.1.1 src-ip 10.1.1.2 session-type multi-hop weight 100
```

  - Configure weight and threshold values for IP monitoring.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
ip threshold 100
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
ip destination-ip 10.10.10.1 weight 50
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
ip destination-ip 20.20.20.1 weight 50
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
ip destination-ip 30.30.30.1 weight 50
```

  - Configure weight and threshold values for interface monitoring.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
interface threshold 100
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
interface interface-name xe-1/0/1 weight 25
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
```

```
interface interface-name xe-1/0/2 weight 25
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
interface interface-name ae0 weight 50
set chassis high-availability services-redundancy-group x monitor monitor-object network-A
interface interface-name ae1 weight 50
```

- Configure monitor-object `network-B`.
  - Set monitor object threshold value.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
object-threshold 200
```

  - Configure BFD monitoring in the monitor-object.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
bfd-liveliness threshold 100
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
bfd-liveliness dst-ip 10.2.2.1 src-ip 10.2.2.2 session-type multi-hop weight 100
```

  - Configure weight and threshold values for IP monitoring.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
ip threshold 100
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
ip destination-ip 10.11.11.1 weight 50
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
ip destination-ip 10.21.21.1 weight 50
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
ip destination-ip 10.31.31.1 weight 50
```

  - Configure weight and threshold values for interface monitoring.

```
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
interface threshold 100
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
```

```
interface interface-name xe-1/0/3 weight 25
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
interface interface-name xe-1/0/4 weight 25
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
interface interface-name ae2 weight 50
set chassis high-availability services-redundancy-group x monitor monitor-object network-B
interface interface-name ae3 weight 50
```

Let take the case of **network-B** monitor-object in the sample.

The system has a threshold value of 100 for interface monitoring and assigned weights for the member interfaces (50, 50, 25, and 25). If an interface of weight 50 goes down, the weight value of the interface (50) is added to the count and compared to the threshold value of the interface-monitoring. That is—count is 50 and interface threshold is 100. The count is still less than interface threshold value.

If another interface of weight 50 goes down, the count is incremented by 50 and compared to the threshold value of the interface-monitoring. The count is now equal to the interface threshold value 100. As the count equals the threshold value, the system adds this value (100) to monitor-object (network-B)'s count. The threshold value of monitor-object network-B is 200. The count (100) is still less than object-monitor's threshold value.

Similarly, if IP monitor or BFD monitor also reach their respective threshold values and add to the object-monitor's count, the count is incremented and compared against object-monitor's threshold value. Once the count suppresses object-monitor's threshold value, the system adds the count to service-redundancy-group (SRG-1)'s count. If the sum of both network-A and network-B object-monitor counts exceeds SRG-1's threshold value, the system triggers failover to another node.

Check Monitoring Objects Configuration

Use the `show chassis high-availability services-redundancy-group 1` or `show chassis high-availability services-redundancy-group <id> monitor-object <name>` commands.

The following sample shows the output of `show chassis high-availability services-redundancy-group 1` command.

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

```
.........................................
        SRG Path Monitor Info:
          SRG Monitor Status: UP
          SRG Monitor Threshold: 100
          SRG Monitor Weight: 0
          SRG Monitor Failed Objects: [ NONE ]



          Object Name: Network-B
          Object Status: UP
          Object Monitored Entries: [ IP IF BFD ]
          Object Failures: [ IP ]
          Object Threshold: 200
          Object Current Weight: 0

          Object Name: Network-A
          Object Status: UP
          Object Monitored Entries: [ IP IF BFD]
          Object Failures: NONE
          Object Threshold: 100
          Object Current Weight: 0
```

In the command output, you can see the status of both monitoring objects `Network-B` and `Network-A`. You can also notice that the failure object details in the output along with their threshold values and weight.

SEE ALSO

- Two-Node Multinode High Availability | **573**
- Example: Configure Multinode High Availability in a Default Gateway Deployment | **743**
- Example: Configure Multinode High Availability in a Layer 3 Network | **698**
- Example: Configure Multinode High Availability in a Hybrid Deployment | **778**

PART 16

Administration

Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade (CLI Procedure)

IN THIS SECTION

You can use nonstop software upgrade (NSSU) to upgrade the software on standalone EX6200 or EX8200 switches with redundant Routing Engines. NSSU upgrades the software running on the Routing Engines and line cards with minimal traffic disruption during the upgrade. NSSU is supported on EX8200 switches running Junos OS Release 10.4 or later and on EX6200 switches running Junos OS Release 12.2 or later.

This topic covers:

Preparing the Switch for Software Installation

Before you begin software installation using NSSU:

- (Optional) Configure line-card upgrade groups as described in Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade. By default, an NSSU upgrades line cards one at a time to allow aggregated Ethernet links that have members on different line cards to remain up through the upgrade process. Configuring line-card upgrade groups reduces the time an upgrade takes because the line cards in each upgrade group are upgraded at the same time rather than sequentially.

- Verify that the Routing Engines are running the same version of the software. Enter the following command:

```
{master}
user@switch> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: switch
Model: ex8208
JUNOS Base OS boot [11.3-20110429.1]
JUNOS Base OS Software Suite [11.3-20110429.1]
JUNOS Kernel Software Suite [11.3-20110429.1]
JUNOS Crypto Software Suite [11.3-20110429.1]
JUNOS Online Documentation [11.3-20110429.1]
JUNOS Enterprise Software Suite [11.3-20110429.1]
LC JUNOS Installation Software [11.3-20110429.1]
JUNOS Routing Software Suite [11.3-20110429.1]
JUNOS Web Management [11.3-20110429.1]

re1:
--------------------------------------------------------------------------
Hostname: switch
Model: ex8208
JUNOS Base OS boot [11.3-20110429.1]
JUNOS Base OS Software Suite [11.3-20110429.1]
JUNOS Kernel Software Suite [11.3-20110429.1]
JUNOS Crypto Software Suite [11.3-20110429.1]
JUNOS Online Documentation [11.3-20110429.1]
JUNOS Enterprise Software Suite [11.3-20110429.1]
LC JUNOS Installation Software [11.3-20110429.1]
JUNOS Routing Software Suite [11.3-20110429.1]
JUNOS Web Management [11.3-20110429.1]
```

  If the Routing Engines are not running the same version of the software, use the `request system software add` command to upgrade the Routing Engine that is running the earlier software version.

- Ensure that nonstop active routing (NSR) and graceful Routing Engine switchover (GRES) are enabled. To verify that they are enabled, you need to check only the state of nonstop active routing—if nonstop active routing is enabled, then graceful Routing Engine switchover is enabled.

To verify that nonstop active routing is enabled, execute the following command:

```
{master}
user@switch> show task replication
        Stateful Replication: Enabled
        RE mode: Master

    Protocol                Synchronization Status
    OSPF                    Complete
    RIP                     Complete
    PIM                     Complete
    RSVP                    Complete
```

If nonstop active routing is not enabled (**Stateful Replication** is **Disabled**), see Configuring Nonstop Active Routing on Switches for information on how to enable it.

- (Optional) Enable nonstop bridging (NSB). Enabling NSB ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover that is part of the NSSU.
- (Optional) Back up the system software on each Routing Engine to an external storage device with the `request system snapshot` command.

Upgrading Both Routing Engines Using NSSU

This procedure describes how to upgrade both Routing Engines using NSSU. When the upgrade completes, both Routing Engines are running the new version of the software, and the backup Routing Engine is the new primary Routing Engine.

To upgrade both Routing Engines using NSSU:

1. Download the software package.
2. Copy the software package to the switch. We recommend that you use FTP to copy the file to the **/var/tmp** directory.
3. Log in to the primary Routing Engine using the console connection. You can perform an NSSU from the management interface, but a console connection allows you to monitor the progress of the primary Routing Engine reboot.

4. Install the new software package:

```
{master}
user@switch> request system software nonstop-upgrade reboot
 /var/tmp/package-name-m.nZx-distribution.tgz
```

where *package-name-m.nZx-distribution*.tgz is, for example, **jinstall-ex-8200-10.4R1.5-domestic-signed.tgz**.

The switch displays the following status messages as the upgrade executes:

```
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Pushing bundle to re1
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 3          Offline                 Offlined by CLI command
  FPC 4          Online (ISSU)
  FPC 5          Online (ISSU)
  FPC 6          Online (ISSU)
  FPC 7          Online (ISSU)
Resolving mastership...
```

```
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
ISSU: Upgrading Old Master RE
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
ISSU: Old Master Upgrade Done
ISSU: IDLE


*** FINAL System shutdown message from user@switch ***
System going down IMMEDIATELY

Shutdown NOW!
[pid 2635]
```

NOTE: If you omit the **reboot** option in this step when using an EX8200 switch, you must manually reboot the original primary Routing Engine with the `request system reboot` command for the upgrade to complete.

The original primary Routing Engine reboots automatically after updating the new primary Routing Engine when an NSSU is used to upgrade an EX6200 switch with dual Routing Engines.

5. Log in after the reboot completes. To verify that both Routing Engines have been upgraded, enter the following command:

```
{backup}
user@switch> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------
Hostname: switch
Model: ex8208
JUNOS Base OS boot [12.1-20111229.0]
JUNOS Base OS Software Suite [12.1-20111229.0]
JUNOS Kernel Software Suite [12.1-20111229.0]
JUNOS Crypto Software Suite [12.1-20111229.0]
JUNOS Online Documentation [12.1-20111229.0]
JUNOS Enterprise Software Suite [12.1-20111229.0]
LC JUNOS Installation Software [12.1-20111229.0]
JUNOS Routing Software Suite [12.1-20111229.0]
JUNOS Web Management [12.1-20111229.0]
```

```
re1:
--------------------------------------------------------------------
Hostname: switch
Model: ex8208
JUNOS Base OS boot [12.1-20111229.0]
JUNOS Base OS Software Suite [12.1-20111229.0]
JUNOS Kernel Software Suite [12.1-20111229.0]
JUNOS Crypto Software Suite [12.1-20111229.0]
JUNOS Online Documentation [12.1-20111229.0]
JUNOS Enterprise Software Suite [12.1-20111229.0]
LC JUNOS Installation Software [12.1-20111229.0]
JUNOS Routing Software Suite [12.1-20111229.0]
JUNOS Web Management [12.1-20111229.0]
```

6. To verify that the line cards that were online before the upgrade are online after the upgrade, log in to the primary Routing Engine and enter the `show chassis nonstop-upgrade` command:

```
{backup}
user@switch> request routing-engine login master

{master}
user@switch> show chassis nonstop-upgrade
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 3          Offline                 Offlined by CLI command
  FPC 4          Online (ISSU)
  FPC 5          Online (ISSU)
  FPC 6          Online (ISSU)
  FPC 7          Online (ISSU)
```

7. If you want to make **re0** the primary Routing Engine again, enter the following command:

```
{master}
user@switch> request chassis routing-engine master switch
Toggle mastership between routing engines ? [yes,no] (no) yes
```

You can verify that **re0** is the primary Routing Engine by executing the `show chassis routing-engine` command.

8. To ensure that the resilient dual-root partitions feature operates correctly, execute the following command to copy the new Junos OS image into the alternate root partition on each Routing Engine:

```
user@switch> request system snapshot slice alternate routing-engine both
```

Resilient dual-root partitions allow the switch to boot transparently from the alternate root partition if the system fails to boot from the primary root partition.

Upgrading One Routing Engine Using NSSU (EX8200 Switch Only)

This procedure describes how to upgrade one of the Routing Engines using NSSU on an EX8200 switch. When the upgrade completes, the backup Routing Engine is running the new software version and is the new primary. The original primary Routing Engine, now the backup Routing Engine, continues to run the previous software version.

NOTE: NSSU always upgrades the software on both Routing Engines on an EX6200 switch. Therefore, you cannot upgrade software on one Routing Engine using NSSU on an EX6200 switch.

To upgrade one Routing Engine using NSSU:

1. Download the software package.
2. Copy the software package to the switch. We recommend that you use FTP to copy the file to the **/var/tmp** directory.
3. Log in to the primary Routing Engine.
4. Request an NSSU. On an EX8200 switch, specify the **no-old-master-upgrade** option when requesting the NSSU:

```
{master}
user@switch> request system software nonstop-upgrade
no-old-master-upgrade /var/tmp/package-name-m.nZx-distribution.tgz
```

where *package-name-m.nZx-distribution*.tgz is, for example, **jinstall-ex-8200-10.4R2.5-domestic-signed.tgz**.

The switch displays the following status messages as the upgrade executes:

```
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Pushing bundle to re1
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
Backup upgrade done
Rebooting Backup RE

Rebooting re1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 3          Offline                 Offlined by CLI command
  FPC 4          Online (ISSU)
  FPC 5          Online (ISSU)
  FPC 6          Online (ISSU)
  FPC 7          Online (ISSU)
Resolving mastership...
Complete. The other routing engine becomes the master.
ISSU: RE switchover Done
Skipping Old Master Upgrade
ISSU: IDLE
```

When the upgrade is complete, the original primary Routing Engine (**re0**) becomes the backup Routing Engine.

5. To verify that the original backup Routing Engine (**re1**) has been upgraded, enter the following command:

```
{backup}
user@switch> show version invoke-on all-routing-engines
re0:
--------------------------------------------------------------------------
Hostname: switch
Model: ex8208
JUNOS Base OS boot [11.3-20110429.1]
JUNOS Base OS Software Suite [11.3-20110429.1]
JUNOS Kernel Software Suite [11.3-20110429.1]
JUNOS Crypto Software Suite [11.3-20110429.1]
JUNOS Online Documentation [11.3-20110429.1]
JUNOS Enterprise Software Suite [11.3-20110429.1]
LC JUNOS Installation Software [11.3-20110429.1]
JUNOS Routing Software Suite [11.3-20110429.1]
JUNOS Web Management [11.3-20110429.1]

re1:
--------------------------------------------------------------------------
Hostname: switch
Model: ex8208
JUNOS Base OS boot [12.1-20111229.0]
JUNOS Base OS Software Suite [12.1-20111229.0]
JUNOS Kernel Software Suite [12.1-20111229.0]
JUNOS Crypto Software Suite [12.1-20111229.0]
JUNOS Online Documentation [12.1-20111229.0]
JUNOS Enterprise Software Suite [12.1-20111229.0]
LC JUNOS Installation Software [12.1-20111229.0]
JUNOS Routing Software Suite [12.1-20111229.0]
JUNOS Web Management [12.1-20111229.0]
```

6. To verify that the line cards that were online before the upgrade are online after the upgrade, log in to the new primary Routing Engine and enter the `show chassis nonstop-upgrade` command:

```
{backup}
user@switch> request routing-engine login master

--- JUNOS 12.1-20111229.0 built 2011-12-29 04:12:22 UTC
{master}
```

```
user@switch> show chassis nonstop-upgrade
  Item           Status                  Reason
  FPC 0          Online
  FPC 1          Online
  FPC 2          Online
  FPC 3          Offline                 Offlined by CLI command
  FPC 4          Online
  FPC 5          Online
  FPC 6          Online
  FPC 7          Online
```

7. To ensure that the resilient dual-root partitions feature operates correctly, copy the new Junos OS image into the alternate root partition of the Routing Engine:

```
user@switch> request system snapshot slice alternate
```

Resilient dual-root partitions allow the switch to boot transparently from the alternate root partition if the system fails to boot from the primary root partition.

Upgrading the Original Primary Routing Engine (EX8200 Switch Only)

This procedure describes how to upgrade the original primary Routing Engine after you have upgraded the original backup Routing Engine as described in "Upgrading One Routing Engine Using NSSU (EX8200 Switch Only)" on page 1165 for an EX8200 switch.

1. Log in to the current primary Routing Engine (**re1**).
2. Enter configuration mode and disable nonstop active routing:

```
{master}[edit]
user@switch# delete routing-options nonstop-routing
```

3. Deactivate graceful Routing Engine switchover and commit the configuration:

```
{master}[edit]
user@switch# deactivate chassis redundancy graceful-switchover
```

```
{master}[edit]
user@switch# commit
```

4. Log in to the current backup Routing Engine (**re0**) using a console connection.
5. Request a software installation:

```
user@switch> request system software add reboot /var/tmp/package-name-m.nZx-distribution.tgz
```

> **NOTE**: When you use NSSU to upgrade only one Routing Engine, the installation package is not automatically deleted from **/var/tmp**, leaving the package available to be used to upgrade the original primary Routing Engine.

6. After the upgrade completes, log in to the current primary Routing Engine (**re1**) and enter CLI configuration mode.
7. Re-enable nonstop active routing and graceful Routing Engine switchover:

```
[edit]
user@switch# activate chassis redundancy graceful-switchover

[edit]
user@switch# set routing-options nonstop-routing

[edit]
user@switch# commit
```

8. To ensure that the resilient dual-root partitions feature operates correctly, exit the CLI configuration mode and copy the new Junos OS image into the alternate root partition of the Routing Engine:

```
user@switch> request system snapshot slice alternate
```

Resilient dual-root partitions allow the switch to boot transparently from the alternate root partition if the system fails to boot from the primary root partition.

9. (Optional) To return control to the original primary Routing Engine (**re0**), enter the following command:

```
{master}
user@switch> request chassis routing-engine master switch
 Toggle mastership between routing engines ? [yes,no] (no) yes
```

You can verify that **re0** is the primary Routing Engine by executing the `show chassis routing-engine` command.

RELATED DOCUMENTATION

Understanding Nonstop Software Upgrade on EX Series Switches | **558**

Configuring Dual-Root Partitions

Troubleshooting Software Installation on EX Series Switches

Upgrading Software on an EX8200 Virtual Chassis Using Nonstop Software Upgrade (CLI Procedure)

IN THIS SECTION

You can use nonstop software upgrade (NSSU) to upgrade the software on an EX8200 Virtual Chassis. NSSU upgrades the software running on all Routing Engines with minimal traffic disruption during the upgrade. NSSU is supported on EX8200 Virtual Chassis with redundant XRE200 External Routing Engines running Junos OS Release 11.1 or later.

NOTE: NSSU upgrades all Routing Engines on all members of the Virtual Chassis and on the XRE200 External Routing Engines. Using NSSU, you cannot choose to upgrade the

backup Routing Engines only, nor can you choose to upgrade a specific member of the Virtual Chassis. If you need to upgrade a specific member of the Virtual Chassis, see *Installing Software for a Single Device in an EX8200 Virtual Chassis*.

This topic covers:

Preparing the Switch for Software Installation

Before you begin software installation using NSSU:

- (Optional) Configure line-card upgrade groups as described in Configuring Line-Card Upgrade Groups for Nonstop Software Upgrade. By default, NSSU upgrades line cards one at a time, starting with the line card in slot 0 of member 0. This permits aggregated Ethernet links that have members on different line cards remain up through the upgrade process. Configuring line-card upgrade groups reduces the time an upgrade takes because the line cards in each upgrade group are upgraded at the same time rather than sequentially.

- Verify that the members are running the same version of the software:

```
{master:8}
user@external-routing-engine> show version all-members
```

  If the Virtual Chassis members are not running the same version of the software, use the `request system software add` command to upgrade the software on the inconsistent members. For instructions, see *Installing Software for a Single Device in an EX8200 Virtual Chassis*.

- Ensure that nonstop active routing (NSR) and graceful Routing Engine switchover (GRES) are enabled. To verify that they are enabled, you need to check only the state of nonstop active routing—if nonstop active routing is enabled, then graceful Routing Engine switchover is enabled.

  To verify that nonstop active routing is enabled:

```
{master:8}
user@switch> show task replication
        Stateful Replication: Enabled
        RE mode: Master
```

```
    Protocol                Synchronization Status
        PIM                     Complete
```

If nonstop active routing is not enabled (**Stateful Replication** is **Disabled**), see Configuring Nonstop Active Routing on Switches for information on how to enable it.

Upgrading the Software Using NSSU

This procedure describes how to upgrade the software running on all Routing Engines using NSSU. When the upgrade completes, all Routing Engines are running the new version of the software. The backup external Routing Engine is now the primary external Routing Engine, and the internal backup Routing Engines in the member switches are now the internal primary Routing Engines in those member switches.

To upgrade all Routing Engines using NSSU:

1. Download the software package for the XRE200 External Routing Engine by following one of the procedures in Downloading Software. The name of the software package for the XRE200 External Routing Engine contains the term **xre200**.
2. Copy the software package to the switch. We recommend that you use FTP to copy the file to the **/var/tmp** directory.
3. Log in to the primary external Routing Engine using the console connection. You can perform an NSSU from the management interface, but a console connection allows you to monitor the progress of the primary Routing Engine reboot.
4. Install the new software package:

```
{master:8}
user@external-routing-engine> request system software nonstop-upgrade reboot
 /var/tmp/package-name-m.nZx-distribution.tgz
```

where ***package-name-m.nZx-distribution*.tgz** is, for example, **jinstall-ex-xre200-11.1R2.5-domestic-signed.tgz**.

> **NOTE**: You can omit **reboot** option. When you include the **reboot** option, NSSU automatically reboots the original primary Routing Engines after the new image has been installed on them. If you omit the **reboot** option, you must manually reboot the original primary Routing Engines (now the backup Routing Engines) to complete the

upgrade. To perform the reboot, you must establish a connection to the console port on the Switch Fabric and Routing Engine (SRE) module or Routing Engine (RE) module.

The switch displays status messages similar to the following messages as the upgrade executes:

```
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing LCC Backup REs
ISSU: Preparing Backup RE
Pushing bundle /var/tmp/jinstall-ex-xre200-11.1-20110208.0-domestic-signed.tgz to member9
member9:
--------------------------------------------------------------------------
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
VC Backup upgrade done
Rebooting VC Backup RE

Rebooting member9
ISSU: Backup RE Prepare Done
Waiting for VC Backup RE reboot
Pushing bundle to member0-backup
Pushing bundle to member1-backup
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately

Rebooting member0-backup
Rebooting LCC [member0-backup]

Rebooting member1-backup
Rebooting LCC [member1-backup]
ISSU: LCC Backup REs Prepare Done
GRES operational
Initiating Chassis Nonstop-Software-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking Nonstop-Upgrade status
```

```
member0:
--------------------------------------------------------------------------
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 5          Online (ISSU)

member1:
--------------------------------------------------------------------------
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 5          Online (ISSU)

member0:
--------------------------------------------------------------------------
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 5          Online (ISSU)

member1:
--------------------------------------------------------------------------
  Item           Status                  Reason
  FPC 0          Online (ISSU)
  FPC 1          Online (ISSU)
  FPC 2          Online (ISSU)
  FPC 5          Online (ISSU)
ISSU: Upgrading Old Master RE
Pushing bundle /var/tmp/incoming-package-8200.tgz to member0-master
Pushing bundle /var/tmp/incoming-package-8200.tgz to member1-master
ISSU: RE switchover Done
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
ISSU: Old Master Upgrade Done
ISSU: IDLE

*** FINAL System shutdown message from root@ ***
System going down
IMMEDIATELY
```

```
Shutdown NOW!
```

NOTE: If you omit the **reboot** option in this step, you must complete the upgrade by separately rebooting the original primary Routing Engine on each Virtual Chassis member and the original primary external Routing Engine. To reboot the original primary Routing Engine on a Virtual Chassis member, you must establish a connection to the console port on the Switch Fabric and Routing Engine (SRE) module or Routing Engine (RE) module.

5. Log in after the reboot completes. To verify that the software on all Routing Engines in the Virtual Chassis members has been upgraded, enter the following command:

```
{backup:8}
user@external-routing-engine> show version all-members
```

6. Verify that the line cards that were online before the upgrade are online after the upgrade by entering the `show chassis nonstop-upgrade` command:

```
{backup:8}
user@external-routing-engine> show chassis nonstop-upgrade
member0:
--------------------------------------------------------------------------
  Item           Status                  Reason
  FPC 0          Online
  FPC 1          Online
  FPC 2          Online
  FPC 5          Online

member1:
--------------------------------------------------------------------------
  Item           Status                  Reason
  FPC 0          Online
  FPC 1          Online
  FPC 2          Online
  FPC 5          Online
```

RELATED DOCUMENTATION

Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure) | 1176

Upgrading Software on an EX6200 or EX8200 Standalone Switch Using Nonstop Software Upgrade (CLI Procedure)

Understanding Nonstop Software Upgrade on EX Series Switches | 558

Configuring Dual-Root Partitions

Troubleshooting Software Installation on EX Series Switches

Understanding Nonstop Software Upgrade on EX Series Switches | 558

Understanding Software Installation on EX Series Switches

Upgrading Software Using Nonstop Software Upgrade on EX Series Virtual Chassis and Mixed Virtual Chassis (CLI Procedure)

IN THIS SECTION

You can use nonstop software upgrade (NSSU) to upgrade the software running on all member switches in most EX Series Virtual Chassis with minimal traffic disruption during the upgrade.

Nonstop software upgrade (NSSU)lists the EX Series switches and Virtual Chassis that support NSSU and the Junos OS release at which they began supporting it.

This topic covers:

Preparing the Switch for Software Installation

Before you begin software installation using NSSU:

- Ensure that the Virtual Chassis is configured correctly to support NSSU. Verify that:
  - The Virtual Chassis members are connected in a ring topology. A ring topology prevents the Virtual Chassis from splitting during an NSSU.
  - The Virtual Chassis primary and backup are adjacent to each other in the ring topology. Adjacency permits the primary and backup to always be in sync, even when the switches in linecard roles are rebooting.
  - The Virtual Chassis is preprovisioned so that the linecard role has been explicitly assigned to member switches acting in the linecard role. During an NSSU, the Virtual Chassis members must maintain their roles—the primary and backup must maintain their primary and backup roles (although primary role will change), and the other member switches must maintain their linecard roles.
  - A two-member Virtual Chassis has `no-split-detection` configured so that the Virtual Chassis does not split when an NSSU upgrades a member.
- Verify that the members are running the same version of the software:

  ```
  user@switch> show version
  ```

  If the Virtual Chassis members are not running the same version of the software, use the `request system software add` command to upgrade the software on the inconsistent members.

- Ensure that nonstop active routing (NSR) and graceful Routing Engine switchover (GRES) are enabled. To verify that they are enabled, you need to check only the state of nonstop active routing—if nonstop active routing is enabled, then graceful Routing Engine switchover is enabled.

  To verify that nonstop active routing is enabled:

  ```
  user@switch> show task replication
          Stateful Replication: Enabled
          RE mode: Master

      Protocol                Synchronization Status
      OSPF                    Complete
      BGP                     Complete
      PIM                     Complete
  ```

If nonstop active routing is not enabled (`Stateful Replication` is `Disabled`), see Configuring Nonstop Active Routing on Switches for information on how to enable it.

- For the EX4300 Virtual Chassis, you should enable the `vcp-no-hold-time` statement at the [`edit virtual-chassis`] hierarchy level before performing a software upgrade using NSSU. If you do not enable the `vcp-no-hold-time` statement, the Virtual Chassis may split during the upgrade. A split Virtual Chassis can cause disruptions to your network, and you may have to manually reconfigure your Virtual Chassis after the NSSU if the split and merge feature was disabled. For more information about a split Virtual Chassis, see *Understanding Split and Merge in a Virtual Chassis*.
- (Optional) Enable nonstop bridging (NSB). Enabling NSB ensures that all NSB-supported Layer 2 protocols operate seamlessly during the Routing Engine switchover that is part of the NSSU.
- (Optional) Back up the system software—Junos OS, the active configuration, and log files—on each member to an external storage device with the `request system snapshot` command.

Upgrading the Software Using NSSU

This procedure describes how to upgrade the software running on all Virtual Chassis members using NSSU. When the upgrade completes, all members are running the new version of the software. Because a graceful Routing Engine switchover occurs during the upgrade, the original Virtual Chassis backup is the new primary.

To upgrade all members using NSSU:

1. Download the software package. If you are upgrading the software running on a mixed Virtual Chassis, download the software packages for both switch types.
2. Copy the software package or packages to the Virtual Chassis. We recommend that you copy the file to the `/var/tmp` directory on the primary.
3. Log in to the Virtual Chassis using the console connection or the virtual management Ethernet (VME) interface. Using a console connection allows you to monitor the progress of the primary switch reboot.
4. Start the NSSU:
   - On an EX3400 Virtual Chassis, enter:

```
user@switch> request system software nonstop-upgrade  /var/tmp/package-name.tgz
```

   where *package-name*.tgz is, for example, `jinstall-ex4200-12.1R2.5-domestic-signed.tgz`.

- On a mixed Virtual Chassis, enter:

```
user@switch> request system software nonstop-upgrade set [/var/tmp/package-
name.tgz /var/tmp/package-name.tgz]
```

  where [/var/tmp/*package-name*.tgz /var/tmp/*package-name*.tgz] specifies the EX4200 and EX4500 software packages.

The switch displays status messages similar to the following messages as the upgrade executes:

```
Chassis ISSU Check Done
ISSU: Validating Image
ISSU: Preparing Backup RE
Installing image on other FPC's along with the backup

Checking pending install on fpc1
Pushing bundle to fpc1
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
Completed install on fpc1

Checking pending install on fpc2
Pushing bundle to fpc2
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
Completed install on fpc2

Rebooting fpc1
ISSU: Backup RE Prepare Done
Waiting for Backup RE reboot
GRES operational
Initiating Chassis In-Service-Upgrade
Chassis ISSU Started
ISSU: Preparing Daemons
ISSU: Daemons Ready for ISSU
ISSU: Starting Upgrade for FRUs
ISSU: Preparing for Switchover
ISSU: Ready for Switchover
Checking In-Service-Upgrade status
  Item           Status                  Reason
  FPC 0          Online
```

```
  FPC 1          Online
  FPC 2          Online (ISSU)
Going to install image on master
WARNING: A reboot is required to install the software
WARNING:     Use the 'request system reboot' command immediately
relinquish mastership
ISSU: IDLE

*** FINAL System shutdown message from user@switch ***

System going down IMMEDIATELY


Shutdown NOW!
[pid 9336]
```

5. Log in after the reboot of the original primary switch completes. To verify that the software on all Routing Engines in the Virtual Chassis members has been upgraded, enter the following command:

```
user@switch> show version
```

6. To ensure that the resilient dual-root partitions feature operates correctly, copy the new Junos OS image into the alternate root partitions of all members:

```
user@switch> request system snapshot slice alternate all-members
```

Resilient dual-root partitions allow the switch to boot transparently from the alternate root partition if the system fails to boot from the primary root partition.

RELATED DOCUMENTATION

Understanding Nonstop Software Upgrade on EX Series Switches | 558

Configuring Dual-Root Partitions

Understanding Software Installation on EX Series Switches

Troubleshooting Software Installation on EX Series Switches

Understanding Nonstop Software Upgrade on EX Series Switches | 558

PART 17

Verification Tasks

Verifying Power Configuration and Use

IN THIS SECTION

Purpose

Verify on a device that supports this feature:

- The power redundancy and line card priority settings
- The PoE power budgets for line cards that support PoE
- Whether the *N*+1 or *N*+*N* power requirements are being met
- Whether the switch has sufficient power for a new line card or an *N*+*N* configuration

Action

Enter the following command:

```
user@switch> show chassis power-budget-statistics
```

Example output:

```
        PSU  0     (EX6200-PWR-AC2500)                :    2500 W   Online
        PSU  1     (EX6200-PWR-AC2500)                :    2500 W   Online
        PSU  2     (EX6200-PWR-AC2500)                :    2500 W   Online
        PSU  3     (EX6200-PWR-AC2500)                :    2500 W   Online
        Total Power supplied by all Online PSUs  :   10000 W
```

```
      Power Redundancy Configuration          :     N+1
      Power Reserved for the Chassis          :     500 W
Fan Tray Statistics           Base power    Power Used
FTC  0                        :     300 W      43.04 W
FPC Statistics                 Base power   Power Used   PoE power   Priority
FPC  1   (EX6200-48P)         :     220 W      49.47 W      1440 W       1
FPC  2   (EX6200-48P)         :     220 W      47.20 W       800 W       2
FPC  3   (EX6200-48P)         :     220 W    1493.57 W      1440 W       0
FPC  4   (EX6200-SRE64-4XS)   :     100 W      51.38 W         0 W       0
FPC  5   (EX6200-SRE64-4XS)   :     100 W      50.28 W         0 W       0
FPC  6   (EX6200-48P)         :     220 W      49.38 W       800 W       6
FPC  8   (EX6200-48P)         :     220 W      61.41 W      1440 W       9
FPC  9   (EX6200-48T)         :     150 W      12.49 W         0 W       9

      Total (non-PoE) Power allocated         :    1750 W
      Total Power allocated for PoE           :    5920 W
      Power Available (Redundant case)        :    5750 W
      Total Power Available                   :    2515 W
```

Meaning

- Example output —The online power supplies can supply a total of 10,000 W to the switch. The switch is configured for *N*+1 redundancy, which means 7500 W of redundant power can be supplied. The **Power Available (Redundant case)** field shows that the switch is meeting the *N*+1 power requirements, with an additional 5750 W available. This value is calculated by subtracting all power allocations except PoE power allocations from redundant power (7500 W).

  The total amount of power available on the switch is 2515 W. This value is calculated by subtracting all power allocations, including PoE power allocations, from the total power (10,000 W). On a switch with PoE line cards, if **Total Power Available** is 0, some or all of the PoE line cards might not be allocated their configured PoE power budgets, which means power to some or all PoE ports might be disabled.

  The power priority order of the line cards, from highest priority line card to the lowest priority line card, is 4, 5, 3, 1, 2, 6, 8, 9. Slots 4 and 5, which contain the Switch Fabric and Routing Engine (SRE) modules, always have highest priority, even if a lower-numbered slot, such as slot 3 in this example, has a priority of 0. Should two or more 2500 W power supplies fail, power management will remove or reduce the PoE power allocations from the PoE line cards in the following order to balance the power budget: 8, 6, 2, 1, and 3.

The **Power Used** values for the fan tray and line cards shows the actual power being consumed for these components at the time the command was executed. These values are for your information only; power management uses allocated power, which is based on the maximum power the component might consume, and not actual power consumed, in determining its power budget.

PART 18

Troubleshooting

Tracing Nonstop Active Routing Synchronization Events

To track the progress of nonstop active routing synchronization between Routing Engines, you can configure nonstop active routing trace options flags for each supported protocol and for BFD sessions and record these operations to a log file.

To configure nonstop active routing trace options for supported routing protocols, include the `nsr-synchronization` statement at the `[edit protocols` *`protocol-name`* `traceoptions flag]` hierarchy level and optionally specify one or more of the **detail**, **disable**, **receive**, and **send** options:

```
[edit protocols]
bgp {
    traceoptions {
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
isis {
    traceoptions {
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
ldp {
    traceoptions {
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
mpls {
    traceoptions {
        flag nsr-synchronization;
        flag nsr-synchronization-detail;
    }
}
msdp {
    traceoptions {
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
(ospf | ospf3) {
    traceoptions {
```

```
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
rip {
    traceoptions {
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
ripng {
    traceoptions {
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
pim {
    traceoptions {
        flag nsr-synchronization <detail> <disable> <receive> <send>;
    }
}
```

To configure nonstop active routing trace options for BFD sessions, include the **nsr-synchronization** and `nsr-packet` statements at the `[edit protocols bfd traceoptions flag]` hierarchy level.

```
[edit protocols]
bfd {
    traceoptions {
        flag nsr-synchronization;
        flag nsr-packet;
    }
}
```

To trace the Layer 2 VPN signaling state replicated from routes advertised by BGP, include the `nsr-synchronization` statement at the `[edit routing-options traceoptions flag]` hierarchy level. This flag also traces the label and logical interface association that VPLS receives from the kernel replication state.

```
[edit routing-options]
traceoptions {
    flag nsr-synchronization;
}
```

RELATED DOCUMENTATION

Configuring Nonstop Active Routing | 280

Troubleshooting the EX Series Redundant Power System Power On and Power Backup Issues

IN THIS SECTION

This topic provides troubleshooting information for problems related to the EX Series Redundant Power System (RPS).

The EX Series RPS Is Not Powering On

IN THIS SECTION

Problem

Description

The RPS does not power on even though it has a power supply installed and is connected to an AC power source outlet.

Environment

The RPS with one EX-PWR3-930-AC power supply installed in it is connected to a switch.

Symptoms

The SYS LED on the power supply side of the RPS is off, and when you check the RPS status using the CLI command **show chassis redundant-power-system**, the message **No RPS connected** is displayed.

Cause

A power supply must be installed in the middle slot on the RPS to power on the RPS.

Solution

Install a power supply in the middle slot on the power supply side of the RPS and verify that the AC power source outlet is properly connected to it. See Installing a Power Supply in the EX Series Redundant Power System.

Verify that the **AC OK** LED and the **DC OK** LED on the power supply in the RPS are lit green.

A Switch Is Not Recognized by the RPS

IN THIS SECTION

Problem

Description

I cannot set up the RPS.

Cause

A switch must be active to be recognized by the RPS.

Solution

Activate the switch by configuring it and issuing a commit statement.

An Error Message Indicates That an RPS Power Supply is Not Supported

IN THIS SECTION

Problem

Description

An RPS error message indicates that an RPS power supply is not supported.

Cause

RPS supports only one power supply, the EX-PWR3-930-AC. If you install another similar power supply, it may fit in the slot but it is not compatible with RPS.

Solution

The power supply shipped with your RPS (in a separate box) is an EX-PWR3-930-AC. If you installed more power supplies, you ordered them separately. Replace any other power supply model (such as the EX-PWR2-930-AC) with an EX-PWR3-930-AC model.

The EX Series Redundant Power System Is Not Providing Power Backup to a Connected Switch

IN THIS SECTION

Problem

Description

The RPS does not provide power backup to a connected switch.

Environment

The RPS has an EX-PWR3-930-AC power supply installed in the middle power supply slot and is connected to two switches with power loss, one connected to RPS switch connector port 1 and the other on port 2.

Symptoms

The status LED on the associated switch connector port is not blinking green–it is either solid green (connected) or not lit (off).

Cause

The RPS provides backup power based on the power priority assigned to each switch.

Solution

If the status LED on a switch connector port is off, ensure that the RPS cable is properly connected to both the RPS and the switch, and ensure that the priority configured for the switch is not 0. See *show redundant-power-system status*.

If the status LED on switch connector port 1 is on and is steadily green, check the backup priority configured for the switch and assign it a higher priority. See Determining and Setting Priority for Switches Connected to an EX Series RPS

If the status LED on switch connector port 1 is amber, check if the RPS has enough power supplies installed in it to provide backup power. if it does not, install a power supply in an empty power supply slot on the RPS. See Installing a Power Supply in the EX Series Redundant Power System.

If the status LED on switch connector port 1 is still off, check the priority configured for the switch. Ensure that the is not set to 0, which means off. See *show redundant-power-system status*. The priority assigned must be from 1 through 6. See Determining and Setting Priority for Switches Connected to an EX Series RPS.

Verify that a dedicated power supply is installed in the switch. The RPS cannot boot a switch that does not have a dedicated power supply. See Installing a Power Supply in the EX Series Redundant Power System.

Also keep in mind that when the command *request redundant-power-system multi-backup* has been set, support for switches that supply PoE is not guaranteed. To reverse this setting, use the command `request redundant-power-system no-multi-backup`.

The Wrong Switches Are Being Backed Up

IN THIS SECTION

Problem

Description

Four or more switches are connected to an RPS with three power supplies. When all four switches fail, the wrong three switches have .

Environment

Four or more switches are connected to an RPS with three power supplies. One or more switches provide PoE to other devices.

Symptoms

When all four switches fail, the wrong three switches have .

Cause

The RPS provides backup power based on the power priority assigned to each switch. This is derived from two configurations, one of which has precedence over the other one. Initial is derived from the location of the port used to attach a switch—the leftmost connector has lowest priority and the rightmost connector has highest priority. The second, dominant priority configuration is derived from a CLI priority setting on the switch itself. With this CLI configuration, 6 is highest priority and 1 is the lowest priority.

Solution

Connect the three switches to the three rightmost connectors on the RPS. Then, using the CLI on each switch, set each switch's priority to 1 using the *redundant-power-system* configuration command **redundant-power-system 1**. Now, physical connection location is determining .

If you do not want to change the cabling on the switches, you can use the configuration statement **redundant-power-system** on all four switches, assigning priority **6** (highest), **5**, **4** and **3** to the appropriate switches. Priority configuration on the switch always overcomes set by connector location.

Six Switches That Do Not Require PoE Are Not All Being Backed Up

IN THIS SECTION

Problem

Description

Only three switches out of six are simultaneously backed up when all switches experience power supply failure. None of these switches supply PoE power to any device.

Environment

The RPS with three EX-PWR3-930-AC power supplies installed in it is connected to six switches, none of which is connected to a non-PoE device.

Symptoms

Only three switches out of six are simultaneously backed up when all switches experience power supply failure. None of these switches supply PoE power to any device.

Cause

Each power supply can support two switches that do not need enough power for PoE, as long as you configure the RPS to do so.

Solution

From any of the attached switches, issue the *request redundant-power-system multi-backup* command from the operational mode. Now standard power will be supplied to two non-PoE switches per power supply.

19

PART

Knowledge Base