

# Junos® OS

---

## Chassis Cluster User Guide for SRX Series Devices

Published  
2024-07-29

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

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

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

*Junos® OS Chassis Cluster User Guide for SRX Series Devices*  
Copyright © 2024 Juniper Networks, Inc. All rights reserved.

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

## YEAR 2000 NOTICE

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

## END USER LICENSE AGREEMENT

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

# Table of Contents

About This Guide | xiv

1

## Overview

Chassis Cluster Overview | 2

Chassis Cluster Overview | 2

Chassis Cluster Limitations | 6

Chassis Cluster Features Supported on SRX Series Devices | 9

2

## Setting Up a Chassis Cluster

SRX Series Chassis Cluster Configuration Overview | 14

Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming | 19

Preparing Your Equipment for Chassis Cluster Formation | 35

Connecting SRX Series Firewalls to Create a Chassis Cluster | 39

Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster | 45

Requirements | 46

Overview | 47

Configuration | 47

Verification | 48

Chassis Cluster Management Interfaces | 49

Understanding Management Interface on an Active Chassis Cluster | 50

Example: Configuring the Chassis Cluster Management Interface | 51

Requirements | 51

Overview | 51

Configuration | 52

Verification | 58

Chassis Cluster Fabric Interfaces | 59

Understanding Chassis Cluster Fabric Interfaces | 60

Example: Configuring the Chassis Cluster Fabric Interfaces | 65

Requirements | 66

Overview | 66

Configuration | 66

Verification | 68

Verifying Chassis Cluster Data Plane Interfaces | 69

Viewing Chassis Cluster Data Plane Statistics | 69

Clearing Chassis Cluster Data Plane Statistics | 70

## **Chassis Cluster Control Plane Interfaces | 72**

Chassis Cluster Control Plane and Control Links | 72

Example: Configure Chassis Cluster Control Ports for Control Link | 76

Requirements | 77

Overview | 77

Configuration | 77

Clear Chassis Cluster Control Plane Statistics | 81

Change from Chassis Cluster to Standalone Mode | 81

## **Chassis Cluster Redundancy Groups | 82**

Understanding Chassis Cluster Redundancy Groups | 83

Example: Configuring Chassis Cluster Redundancy Groups | 87

Requirements | 87

Overview | 87

Configuration | 88

Verification | 90

## **Chassis Cluster Redundant Ethernet Interfaces | 91**

Understanding Chassis Cluster Redundant Ethernet Interfaces | 91

Example: Configuring Chassis Cluster Redundant Ethernet Interfaces | 95

Requirements | 95

Overview | 95

Configuration | 96

Verification | 101

Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on SRX4600 | 103

Requirements | 103

Overview | 103

Configuration | 103

Verification | 108

Example: Configuring IP-over-IP Tunnel on SRX Series Firewalls | 110

Requirements | 110

Overview | 111

Topology | 111

Configuration | 111

Verification | 116

## Configuring Chassis Clustering on SRX Series Devices | 120

Example: Configure Chassis Clustering on SRX Series Firewalls | 122

Requirements | 122

Overview | 123

Configuration | 124

Verification | 133

Viewing a Chassis Cluster Configuration | 138

Viewing Chassis Cluster Statistics | 139

Clearing Chassis Cluster Statistics | 141

Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes | 142

Verifying Chassis Cluster Configuration Synchronization Status | 143

## Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Firewalls in a Chassis Cluster | 144

Requirements | 145

Overview | 145

Configuration | 147

Verification | 155

## Conditional Route Advertisement over Redundant Ethernet Interfaces on SRX Series Firewalls in a Chassis Cluster | 156

Understanding Conditional Route Advertising in a Chassis Cluster | 156

Example: Configuring Conditional Route Advertising in a Chassis Cluster | 157

Requirements | 157

Overview | 157

Configuration | 159

3

## Configuring Redundancy and Failover in a Chassis Cluster

### Chassis Cluster Dual Control Links | 163

Chassis Cluster Dual Control Links Overview | 163

Dual Control Link Connections for SRX Series Firewalls in a Chassis Cluster | 165

Upgrade the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices | 166

Example: Configure Chassis Cluster Control Ports for Dual Control Links | 169

Requirements | 169

Overview | 169

Configuration | 170

Verification | 171

### Chassis Cluster Dual Fabric Links | 172

Understanding Chassis Cluster Dual Fabric Links | 173

Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports | 174

Requirements | 174

Overview | 174

Configuration | 175

Verification | 176

Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports | 177

Requirements | 178

Overview | 178

Configuration | 179

Verification | 180

### Monitoring of Global-Level Objects in a Chassis Cluster | 181

## **Monitoring Chassis Cluster Interfaces | 185**

Understanding Chassis Cluster Redundancy Group Interface Monitoring | 186

Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring | 187

Requirements | 187

Overview | 188

Configuration | 189

Verification | 194

## **Monitoring IP Addresses on a Chassis Cluster | 227**

IP Monitoring Overview | 228

Understanding Chassis Cluster Redundancy Group IP Address Monitoring | 231

Example: Configure Chassis Cluster Redundancy Group IP Address Monitoring | 233

Requirements | 233

Overview | 234

Configuration | 234

Verification | 237

Example: Configuring IP Monitoring on SRX5000 Line Devices for IOC2 and IOC3 | 238

Requirements | 238

Overview | 238

Configuration | 239

Verification | 246

## **Configuring Cluster Failover Parameters | 248**

Understanding Chassis Cluster Control Link Heartbeats, Failure, and Recovery | 248

Example: Configuring Chassis Cluster Control Link Recovery | 251

Requirements | 251

Overview | 251

Configuration | 252

## **Understanding Chassis Cluster Resiliency | 253**

### **Chassis Cluster Redundancy Group Failover | 254**

Understanding Chassis Cluster Redundancy Group Failover | 255

Understanding Chassis Cluster Redundancy Group Manual Failover | 260

Initiating a Chassis Cluster Manual Redundancy Group Failover | 261

Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers | 264

Requirements | 264

Overview | 264

Configuration | 264

Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover | 265

Verifying Chassis Cluster Failover Status | 266

Clearing Chassis Cluster Failover Status | 268

## 4

### Chassis Cluster Operations

Aggregated Ethernet Interfaces in a Chassis Cluster | 270

Understanding Link Aggregation Groups in a Chassis Cluster | 270

Example: Configuring Link Aggregation Groups in a Chassis Cluster | 272

Requirements | 273

Overview | 273

Configuration | 274

Verification | 276

Understanding Link Aggregation Group Failover in a Chassis Cluster | 277

Understanding LACP on Chassis Clusters | 279

Example: Configuring LACP on Chassis Clusters | 282

Requirements | 283

Overview | 283

Configuration | 284

Verification | 290

Example: Configuring Chassis Cluster Minimum Links | 293

Requirements | 293

Overview | 293

Configuration | 294

Verification | 294

Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on an SRX5000 Line Device with IOC2 or IOC3 | 296



Requirements | 296

Overview | 296

Configuration | 297

Verification | 300

Understanding VRRP on SRX Series Firewalls | 301

VRRP failover-delay Overview | 305

Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces | 307

Requirements | 308

Overview | 308

Configuration VRRP | 309

Verification | 317

Example: Configuring VRRP for IPv6 | 320

Requirements | 320

Overview | 320

Configuring VRRP | 321

Verification | 328

## **NTP Time Synchronization on Chassis Cluster | 332**

NTP Time Synchronization on SRX Series Devices | 333

Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP | 334

Requirements | 334

Overview | 335

Configuration | 335

Verification | 337

## **Active/Passive Chassis Cluster Deployments | 341**

Understanding Active/Passive Chassis Cluster Deployment | 341

Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Firewalls | 342

Requirements | 343

Overview | 343

Configuration | 345

Verification | 353

Example: Configuring an Active/Passive Chassis Cluster Pair (SRX1500 or SRX1600) | 361

Requirements | 361

Overview | 362

Configuration | 366

Verification | 372

Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) | 378

Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel | 380

Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel | 382

Requirements | 382

Overview | 383

Configuration | 389

Verification | 399

Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) | 404

**Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster | 407**

Requirements | 408

Overview | 408

Configuration | 411

Verification | 423

**Example: Configuring an Active/Active Layer 3 Cluster Deployment | 433**

Requirements | 433

Overview | 434

Configuration | 435

Verification | 443

**Multicast Routing and Asymmetric Routing on Chassis Cluster | 456**

Understanding Multicast Routing on a Chassis Cluster | 456

Understanding Asymmetric Routing on a Chassis Cluster | 458

Example: Configuring an Asymmetric Chassis Cluster Pair | 460

Requirements | 461

Overview | 461

Configuration | 465

Verification | 472

## Ethernet Switching on Chassis Cluster | 477

Layer 2 Ethernet Switching Capability in a Chassis Cluster Mode | 478

Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device | 479

Requirements | 479

Overview | 480

Configuration | 480

## Media Access Control Security (MACsec) on Chassis Cluster | 483

Understanding Media Access Control Security (MACsec) | 483

Configure Media Access Control Security (MACsec) | 486

Configuration Considerations When Configuring MACsec on Chassis Cluster Setup | 487

Configure MACsec Using Static Connectivity Association Key Security Mode | 488

Configure Static CAK on the Chassis Cluster Control Port | 494

Configure Static CAK on the Chassis Cluster Fabric Port | 495

Configure Static CAK on the Control Port for SRX1600, SRX2300, and SRX4300 Devices | 495

Configure Static CAK on the Control Port for SRX4600 | 499

Verify MACSEC Configuration | 500

## Understanding SCTP Behavior in Chassis Cluster | 506

Example: Encrypting Messages Between Two Nodes in a Chassis Cluster | 507

5

## Upgrading or Disabling a Chassis Cluster

Upgrading Individual Devices in a Chassis Cluster Separately | 510

Upgrading Devices in a Chassis Cluster Using ICU | 510

Upgrading Both Devices in a Chassis Cluster Using ICU | 511

Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster | 512

Upgrading ICU Using a Build Available on an FTP Server | 513

Terminating an Upgrade in a Chassis Cluster During an ICU | 514

Upgrading a Chassis Cluster Using In-Service Software Upgrade | 516

Understanding ISSU for a Chassis Cluster | 516

ISSU System Requirements | 519

- Upgrading Both Devices in a Chassis Cluster Using ISSU | 521
- Rolling Back Devices in a Chassis Cluster After an ISSU | 523
- Enabling an Automatic Chassis Cluster Node Failback After an ISSU | 523
- Log Error Messages used for Troubleshooting ISSU-Related Problems | 524
  - Chassisd Process Errors | 524
  - Understanding Common Error Handling for ISSU | 525
  - ISSU Support-Related Errors | 529
  - Initial Validation Checks Failure | 529
  - Installation-Related Errors | 531
  - Redundancy Group Failover Errors | 532
  - Kernel State Synchronization Errors | 533
- Managing Chassis Cluster ISSU-Related Problems | 533
  - Viewing ISSU Progress | 534
  - Stopping ISSU Process if it Halts During an Upgrade | 535
  - Recovering the Node in Case of a Failed ISSU | 536

## Disabling a Chassis Cluster | 538

## 6

## Troubleshooting

- Troubleshooting a Control Link Failure in an SRX Chassis Cluster | 541
- Troubleshooting a Fabric Link Failure in an SRX Chassis Cluster | 543
- Troubleshooting a Redundancy Group that Does Not Fail Over in an SRX Chassis Cluster | 546
- Troubleshooting an SRX Chassis Cluster with One Node in the Primary State and the Other Node in the Disabled State | 551
- Troubleshooting an SRX Chassis Cluster with One Node in the Primary State and the Other Node in the Lost State | 555
- Troubleshooting an SRX Chassis Cluster with One Node in the Hold State and the Other Node in the Lost State | 558
- Troubleshooting Chassis Cluster Management Issues | 562
  - Unable to Manage an SRX Series Chassis Cluster Using the Management Port or Revenue Ports | 562

Unable to Manage the Secondary Node of a Chassis Cluster Using J-Web | 574

Unable to Manage an SRX Series Chassis Cluster Using fxp0 When the Destination in the Backup Router is 0/0 | 576

Unable to Upgrade a Chassis Cluster Using In-Service Software Upgrade | 582

Configuring backup-router Command on Chassis Cluster | 584

Unable to Upgrade a Chassis Cluster Using In-Service Software Upgrade | 585

**Data Collection for Customer Support | 587**

7

## **Configuration Statements and Operational Commands**

**Junos CLI Reference Overview | 590**

8

## **Chassis Cluster Support on SRX100, SRX210, SRX220, SRX240, SRX550M, SRX650, SRX1400, SRX3400, and SRX3600 Devices**

**Chassis Cluster Support on SRX100, SRX210, SRX220, SRX240, SRX550M, SRX650, SRX1400, SRX3400, and SRX3600 Devices | 592**

Example: Configure IRB and VLAN with Members Across Two Nodes on a Security Device using Tagged | 608

Requirements | 609

Overview | 609

Configuration | 610

Example: Configure IRB and VLAN with Members Across Two Nodes on a Security Device using Untagged Traffic | 619

Requirements | 619

Overview | 619

Configuration | 620

Example: Configuring VLAN with Members Across Two Nodes on a Security Device | 629

Requirements | 629

Overview | 629

Configuration | 629

Verification | 632

# About This Guide

Use this guide to configure and operate the SRX Series Firewalls in chassis cluster mode, where a pair of devices are connected and configured to operate as a single node, providing device, interface, and service level redundancy.

# 1

CHAPTER

## Overview

---

[Chassis Cluster Overview | 2](#)

[Chassis Cluster Features Supported on SRX Series Devices | 9](#)

---

# Chassis Cluster Overview

## IN THIS SECTION

- [Chassis Cluster Overview | 2](#)
- [Chassis Cluster Limitations | 6](#)

A chassis cluster provides high availability on SRX Series Firewalls where two devices operate as a single device. Chassis cluster includes the synchronization of configuration files and the dynamic runtime session states between the SRX Series Firewalls, which are part of chassis cluster setup.

## Chassis Cluster Overview

### IN THIS SECTION

- [Benefits of Chassis Cluster | 3](#)
- [Chassis Cluster Functionality | 3](#)
- [Chassis Cluster Modes | 3](#)
- [How Chassis Clustering Works? | 4](#)
- [IPv6 Clustering Support | 4](#)
- [Use case for SRX Chassis Clusters | 5](#)

The Junos OS provides high availability on SRX Series Firewall by using chassis clustering. SRX Series Firewalls can be configured to operate in cluster mode, where a pair of devices can be connected together and configured to operate like a single node, providing device, interface, and service level redundancy.

For SRX Series Firewalls, which act as stateful firewalls, it is important to preserve the state of the traffic between two devices. In a chassis cluster setup, in the event of failure, session persistence is required so that the established sessions are not dropped even if the failed device was forwarding traffic.



When configured as a *chassis cluster*, the two nodes back up each other, with one node acting as the primary device and the other as the secondary device, ensuring stateful failover of processes and services in the event of system or hardware failure. If the primary device fails, the secondary device takes over the processing of traffic. The cluster nodes are connected together with two links called control link and fabric link and devices in a chassis cluster synchronize the configuration, kernel, and PFE session states across the cluster to facilitate high availability, failover of stateful services, and load balancing.

There is no separate license required to enable chassis cluster. However, some Junos OS software features require a license to activate the feature. For more information, see [Understanding Chassis Cluster Licensing Requirements](#), [Installing Licenses on the SRX Series Devices in a Chassis Cluster](#) and [Verifying Licenses on an SRX Series Device in a Chassis Cluster](#). Please refer to the Juniper Licensing Guide for general information about License Management. Please refer to the product Data Sheets at [SRX Series Services Gateways](#) for details, or contact your Juniper Account Team or Juniper Partner.

## Benefits of Chassis Cluster

- Prevents single device failure that results in a loss of connectivity.
- Provides high availability between devices when connecting branch and remote site links to larger corporate offices. By leveraging the chassis cluster feature, enterprises can ensure connectivity in the event of device or link failure.

## Chassis Cluster Functionality

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple Packet Forwarding Engines. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.

## Chassis Cluster Modes

A chassis cluster can be configured in an active/active or active/passive mode.

- **Active/passive mode:** In active/passive mode, transit traffic passes through the primary node while the backup node is used only in the event of a failure. When a failure occurs, the backup device becomes primary and takes over all forwarding tasks.
- **Active/active mode:** In active/active mode, has transit traffic passing through both nodes of the cluster all of the time.

## How Chassis Clustering Works?

The control ports on the respective nodes are connected to form a control plane that synchronizes configuration and kernel state to facilitate the high availability of interfaces and services.

The data plane on the respective nodes is connected over the fabric ports to form a unified data plane.

When creating a chassis cluster, the control ports on the respective nodes are connected to form a control plane that synchronizes the configuration and kernel state to facilitate the high availability of interfaces and services.

Similarly, the data plane on the respective nodes is connected over the fabric ports to form a unified data plane.

The fabric link allows for the management of cross-node flow processing and for the management of session redundancy.

The control plane software operates in active or backup mode. When configured as a chassis cluster, the two nodes back up each other, with one node acting as the primary device and the other as the secondary device, ensuring stateful failover of processes and services in the event of system or hardware failure. If the primary device fails, the secondary device takes over processing of traffic.

The data plane software operates in active/active mode. In a chassis cluster, session information is updated as traffic traverses either device, and this information is transmitted between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, it is possible for traffic to ingress the cluster on one node and egress from the other node. When a device joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

At any given instant, a cluster can be in one of the following states: hold, primary, secondary-hold, secondary, ineligible, and disabled. A state transition can be triggered because of any event, such as interface monitoring, SPU monitoring, failures, and manual failovers.

## IPv6 Clustering Support

SRX Series Firewalls running IP version 6 (IPv6) can be deployed in active/active (failover) chassis cluster configurations in addition to the existing support of active/passive (failover) chassis cluster configurations. An interface can be configured with an IPv4 address, IPv6 address, or both. Address book entries can include any combination of IPv4 addresses, IPv6 addresses, and Domain Name System (DNS) names.

Chassis cluster supports Generic Routing Encapsulation (GRE) tunnels used to route encapsulated IPv4/IPv6 traffic by means of an internal interface, `gr-0/0/0`. This interface is created by Junos OS at system bootup and is used only for processing GRE tunnels. See the [Interfaces User Guide for Security Devices](#).

## Use case for SRX Chassis Clusters

Enterprise and service provider networks employ various redundancy and resiliency methods at the customer edge network tier. As this tier represents the entrance or peering point to the Internet, its stability and uptime are of great importance. Customer transactional information, email, Voice over IP (VoIP), and site-to-site traffic can all utilize this single entry point to the public network. In environments where a site-to-site VPN is the only interconnect between customer sites and the headquarter site, this link becomes even more vital.

Traditionally, multiple devices with discreet configurations have been used to provide redundancy at this network layer with mixed results. In these configurations, the enterprise relies on routing and redundancy protocols to enable a highly available and redundant customer edge. These protocols are often slow to recognize failure and do not typically allow for the synchronization required to properly handle stateful traffic. Given that a fair amount of enterprise traffic passing through the edge (to/from the Internet, or between customer sites) is stateful, a consistent challenge in the configuration of this network tier has been ensuring session state is not lost when failover or reversion occurs.

Another challenge in configuration of redundant devices is the need to configure, manage, and maintain separate physical devices with different configurations. Synchronizing those configurations can also be a challenge because as the need and complexity of security measures increase, so too does the probability that configurations are mismatched. In a secure environment, a mismatched configuration can cause something as simple as a loss of connectivity or as complex and costly as a total security breach. Any anomalous event on the customer edge can affect uptime, which consequently impacts the ability to service customers, or possibly the ability to keep customer data secure.

An answer to the problem of redundant customer edge configuration is to introduce a state-aware clustering architecture that allows two or more devices to operate as a single device. Devices in this type of architecture are able to share session information between all devices to allow for near instantaneous failover and reversion of stateful traffic. A key measure of success in this space is the ability of the cluster to fail over and revert traffic while maintaining the state of active sessions.

Using the SRX Chassis Cluster configuration described in ["Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster" on page 407](#) will reduce system downtime.

Devices in an effective clustering architecture can also be managed as a single device; sharing a single control plane. This function is vital as it reduces the OpEx associated with managing multiple devices. Rather than managing and operating separate devices with different configurations and management portals, you can manage multiple devices that serve the same function through a single management point.

Finally, in a cluster configuration, devices have the ability to monitor active interfaces to determine their service state. An effective cluster proactively monitors all revenue interfaces and should fail over to backup interfaces if a failure is detected. This should be done at nearly instantaneous intervals to minimize the impact of a service failure (dropped customer calls, and so on).

## Chassis Cluster Limitations

The SRX Series Firewalls have the following chassis cluster limitations:

### Chassis Cluster

- Group VPN is not supported.
- On all SRX Series Firewalls in a chassis cluster, flow monitoring for version 5 and version 8 is supported. However, flow monitoring for version 9 is not supported.
- When an SRX Series Firewall is operating in chassis cluster mode and encounter any IA-chip access issue in an SPC or a I/O Card (IOC), a minor FPC alarm is activated to trigger redundancy group failover.
- On SRX5400, SRX5600, and SRX5800 devices, screen statistics data can be gathered on the primary device only.
- On SRX4600, SRX5400, SRX5600, and SRX5800 devices, in large chassis cluster configurations, if more than 1000 logical interfaces are used, the cluster heartbeat timers are recommended to be increased from the default wait time before triggering failover. In a full-capacity implementation, we recommend increasing the wait to 8 seconds by modifying `heartbeat-threshold` and `heartbeat-interval` values in the `[edit chassis cluster]` hierarchy.

The product of the `heartbeat-threshold` and `heartbeat-interval` values defines the time before failover. The default values (`heartbeat-threshold` of 3 beats and `heartbeat-interval` of 1000 milliseconds) produce a wait time of 3 seconds.

To change the wait time, modify the option values so that the product equals the desired setting. For example, setting the `heartbeat-threshold` to 8 and maintaining the default value for the `heartbeat-interval` (1000 milliseconds) yields a wait time of 8 seconds. Likewise, setting the `heartbeat-threshold` to 4 and the `heartbeat-interval` to 2000 milliseconds also yields a wait time of 8 seconds.

- On SRX5400, SRX5600, and SRX5800 devices, eight-queue configurations are not reflected on the chassis cluster interface.

### Flow and Processing

- If you use packet capture on reth interfaces, two files are created, one for ingress packets and the other for egress packets based on the reth interface name. These files can be merged outside of the device using tools such as Wireshark or Mergecap.
- If you use port mirroring on reth interfaces, the reth interface cannot be configured as the output interface. You must use a physical interface as the output interface. If you configure the reth interface as an output interface using the `set forwarding-options port-mirroring family inet output` command, the following error message is displayed.

Port-mirroring configuration error.

Interface type in reth1.0 is not valid for port-mirroring or next-hop-group config

- When an SRX Series Firewall is operating in chassis cluster mode and encounter any IA-chip (IA-chip is part of Juniper SPC1 and IOC1. It has direct impact on SPC1/IOC1 control plane) access issue in an SPC or a I/O Card (IOC), a minor FPC alarm is activated to trigger redundancy group failover.
- On SRX Series Firewalls in a chassis cluster, when two logical systems are configured, the scaling limit crosses 13,000, which is very close to the standard scaling limit of 15,000, and a convergence time of 5 minutes results. This issue occurs because multicast route learning takes more time when the number of routes is increased.
- On SRX4600, SRX5400, SRX5600, and SRX5800 devices in a chassis cluster, if the primary node running the LACP process (lacpd) undergoes a graceful or ungraceful restart, the lacpd on the new primary node might take a few seconds to start or reset interfaces and state machines to recover unexpected synchronous results. Also, during failover, when the system is processing traffic packets or internal high-priority packets (deleting sessions or reestablishing tasks), medium-priority LACP packets from the peer (switch) are pushed off in the waiting queues, causing further delay.

Flowd monitoring is supported on SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, SRX1600, SRX2300, and SRX4300 devices.

### Installation and Upgrade

- For SRX300, SRX320, SRX340, SRX345, and SRX380 devices, the reboot parameter is not available, because the devices in a cluster are automatically rebooted following an in-band cluster upgrade (ICU).

### Interfaces

- On the lsq-0/0/0 interface, Link services MLPPP, MLFR, and CRTP are not supported.
- On the lt-0/0/0 interface, CoS for RPM is not supported.
- The 3G dialer interface is not supported.
- Queuing on the ae interface is not supported.

### Layer 2 Switching

- On SRX Series Firewall failover, access points on the Layer 2 switch reboot and all wireless clients lose connectivity for 4 to 6 minutes.

### MIBs

- The Chassis Cluster MIB is not supported.

### Monitoring

- The maximum number of monitoring IPs that can be configured per cluster is 64 for SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, SRX1600, SRX2300, and SRX4300 devices.
- On SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, SRX1600, SRX2300, and SRX4300 devices, logs cannot be sent to NSM when logging is configured in the stream mode. Logs cannot be sent because the security log does not support configuration of the source IP address for the fxp0 interface and the security log destination in stream mode cannot be routed through the fxp0 interface. This implies that you cannot configure the security log server in the same subnet as the fxp0 interface and route the log server through the fxp0 interface.

### IPv6

- Redundancy group IP address monitoring is not supported for IPv6 destinations.

### GPRS

- On SRX5400, SRX5600, and SRX5800 devices, an APN or an IMSI filter must be limited to 600 for each GTP profile. The number of filters is directly proportional to the number of IMSI prefix entries. For example, if one APN is configured with two IMSI prefix entries, then the number of filters is two.

### MIBs

- The Chassis Cluster MIB is not supported.

### Nonstop Active Routing (NSR)

- NSR can preserve interface and kernel information and saves routing protocol information by running the routing protocol process (RPD) on the backup Routing Engine. However, most SRX platforms do not support NSR yet. So on the secondary node, there is no existing RPD daemon. After RGO failover happens, the new RGO master will have a new RPD and need to re-negotiate with peer device. Only SRX5000 platforms with version 17.4R2 or higher can support NSR.

Starting with Junos OS Release 12.1X45-D10 and later, sampling features such as flow monitoring, packet capture, and port mirroring are supported on reth interfaces.

### SEE ALSO

| [Preparing Your Equipment for Chassis Cluster Formation](#) | 35

### 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
12.1X45	Starting with Junos OS Release 12.1X45-D10 and later, sampling features such as flow monitoring, packet capture, and port mirroring are supported on reth interfaces.

### RELATED DOCUMENTATION

[SRX Series Chassis Cluster Configuration Overview | 14](#)

[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming | 19](#)

[Example: Configure Chassis Clustering on SRX Series Firewalls | 122](#)

## Chassis Cluster Features Supported on SRX Series Devices

### IN THIS SECTION

- [Supported Features on Standalone and Chassis Cluster | 9](#)

To determine if a feature is supported by a specific platform or Junos OS release, refer to [Feature Explorer](#).

### Supported Features on Standalone and Chassis Cluster

All features are supported in both chassis cluster and standalone mode on the same platform, except what is indicated in the below table.

[Table 1 on page 10](#) lists the features that are not supported in standalone or chassis cluster.

**Table 1: Features Not Supported in Standalone or Chassis Cluster**

Category	Features	Standalone	Chassis cluster
<b>Ethernet Link Aggregation</b>	LACP (port priority) Layer 3 Mode  LACP (port priority) Layer 2 Mode		Yes  Not supported in Active/Backup and Active/Active modes.
<b>Diagnostics Tools</b>	J-Flow  Ping MPLS	Yes	Yes  Not supported in Active/Active and Active/Active Failover mode. But state is synched to backup node.
<b>Ethernet Interfaces</b>	Promiscuous mode on Ethernet interface		Yes
<b>Chassis Management</b>	Chassis cluster SPC insert  IEEE 802.3af/ 802.3at support	Yes	Yes  Not supported in Active/Backup, Active/Backup Failover, Active/Active, Active/Active Failover.
<b>Class of Service</b>	Simple filters		Yes

**Table 1: Features Not Supported in Standalone or Chassis Cluster (Continued)**

<b>Flow-Based and Packet-Based Processing-</b>	End-to-end packet debugging  Express Path support  Host bound fragmented traffic  Packet-based processing  Selective stateless packet-based services		Yes
<b>GPRS</b>	GPRS (transparent mode and route mode)		Yes



<b>Multicast VPN</b>	<p>Basic multicast features in C-instance</p> <p>Multicast VPN membership discovery with BGP</p> <p>P2MP LSP support</p> <p>P2MP OAM to P2MP LSP ping</p> <p>Reliable multicast VPN routing information exchange</p>	Yes
<b>GTPv2</b>	<p>IMSI prefix and APN filtering</p> <p>Message-length filtering</p> <p>Message-rate limiting</p> <p>Message-type filtering</p> <p>Packet sanity check</p> <p>Policy-based inspection</p> <p>Restart GTPv2 path</p> <p>Sequence-number and GTP-U validation</p> <p>Stateful inspection</p> <p>Traffic logging</p> <p>Tunnel cleanup</p>	Yes
<b>IDP</b>	<p>Cryptographic key handling</p> <p>DSCP marking</p> <p>IDP class-of-service action</p> <p>IDP inline tap mode</p> <p>IDP SSL inspection</p> <p>Jumbo frames</p> <p>Performance and capacity tuning for IDP</p>	Yes
<b>Layer 2 Mode</b>	Q-in-Q tunneling	Yes
<b>SNMP v1,v2,v3</b>		Yes

<b>Stateless Firewall Filters</b>	Stateless firewall filters(ACLs)		Yes
<b>Transparent Mode</b>	Bridge domain		Yes
<b>UTM</b>	Antivirus-Express Antivirus-Full Stateful active/active cluster mode Web filtering-Websense redirect		Yes  Not supported in Active/Backup Failover and Active/Active Failover modes.
<b>User Interfaces</b>	J-Web user interface Junos XML protocol Session and Resource Control (SRC) application		Yes
<b>Upgrading and Rebooting</b>	ISSU	No	Yes

## RELATED DOCUMENTATION

[Chassis Cluster Overview | 2](#)

# 2

CHAPTER

## Setting Up a Chassis Cluster

---

[SRX Series Chassis Cluster Configuration Overview | 14](#)

[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming | 19](#)

[Preparing Your Equipment for Chassis Cluster Formation | 35](#)

[Connecting SRX Series Firewalls to Create a Chassis Cluster | 39](#)

[Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster | 45](#)

[Chassis Cluster Management Interfaces | 49](#)

[Chassis Cluster Fabric Interfaces | 59](#)

[Chassis Cluster Control Plane Interfaces | 72](#)

[Chassis Cluster Redundancy Groups | 82](#)

[Chassis Cluster Redundant Ethernet Interfaces | 91](#)

[Configuring Chassis Clustering on SRX Series Devices | 120](#)

[Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Firewalls in a Chassis Cluster | 144](#)

[Conditional Route Advertisement over Redundant Ethernet Interfaces on SRX Series Firewalls in a Chassis Cluster | 156](#)

---

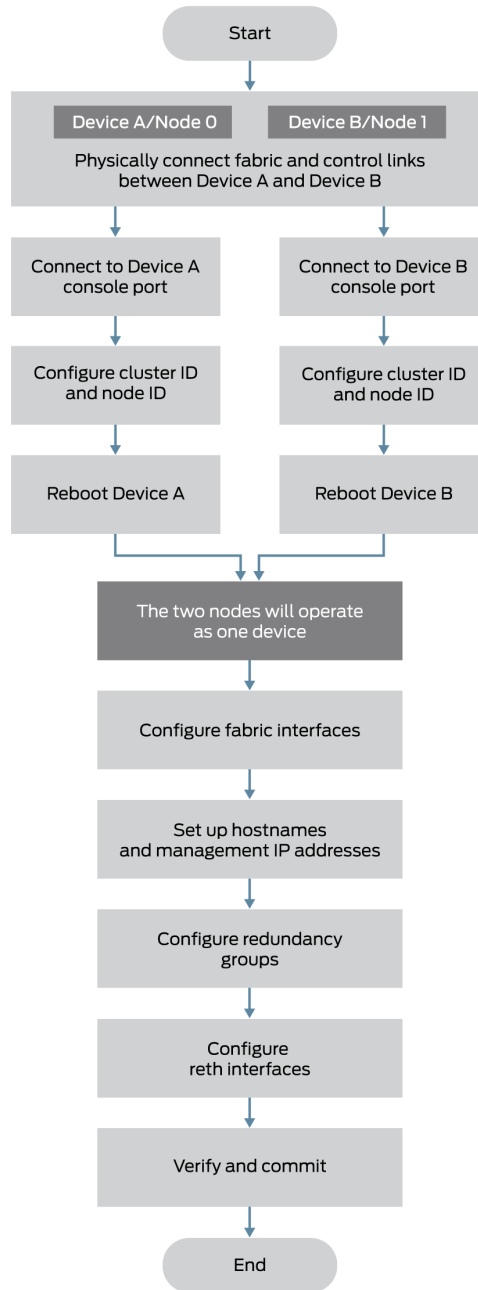
# SRX Series Chassis Cluster Configuration Overview

Following are the prerequisites for configuring a chassis cluster:

- On SRX300, SRX320, SRX340, SRX345, and SRX380 any existing configurations associated with interfaces that transform to the fxp0 management port and the control port should be removed. For more information, see ["Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming"](#) on page 19.
- Confirm that hardware and software are the same on both devices.
- Confirm that license keys are the same on both devices.
- For SRX300, SRX320, SRX340, SRX345, and SRX380 chassis clusters, the placement and type of GPIMs, XGPIMs, XPIMs, and Mini-PIMs (as applicable) must match in the two devices.
- For SRX5000 line chassis clusters, the placement and type of SPCs must match in the two devices.

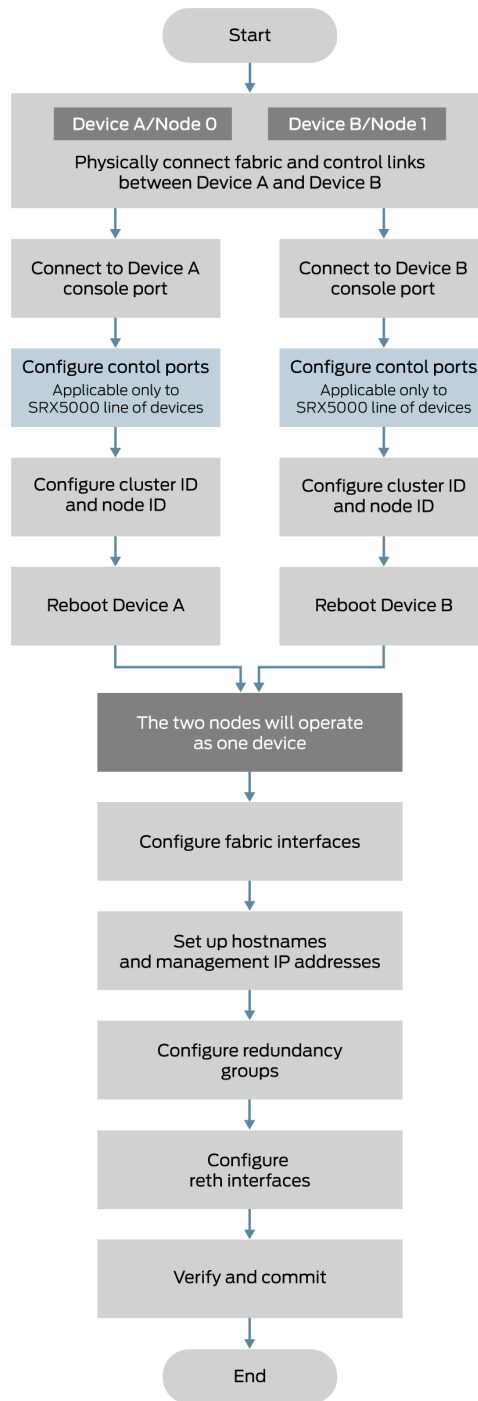
[Figure 1 on page 15](#) shows a chassis cluster flow diagram for SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, SRX1600, SRX2300, SRX4300, SRX4100, SRX4200, and SRX4600 devices.

Figure 1: Chassis Cluster Flow Diagram (SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, SRX1600, SRX2300, SRX4100, SRX4200, SRX4300 and SRX4600 Devices)



g043313

Figure 2: Chassis Cluster Flow Diagram (SRX5800, SRX5600, SRX5400 Devices)



This section provides an overview of the basic steps to create an SRX Series chassis cluster. To create an SRX Series chassis cluster:

1. Prepare the SRX Series Firewalls to be used in the chassis cluster. For more information, see ["Preparing Your Equipment for Chassis Cluster Formation"](#) on page 35.

2. Physically connect a pair of the same kind of supported SRX Series Firewalls together. For more information, see ["Connecting SRX Series Devices to Create a Chassis Cluster" on page 39](#).
  - a. Create the fabric link between two nodes in a cluster by connecting any pair of Ethernet interfaces. For most SRX Series Firewalls, the only requirement is that both interfaces be Gigabit Ethernet interfaces (or 10-Gigabit Ethernet interfaces).

When using dual fabric link functionality, connect the two pairs of Ethernet interfaces that you will use on each device. See ["Understanding Chassis Cluster Dual Fabric Links" on page 173](#).

- b. Configure the control ports (SRX5000 line only). See ["Example: Configuring Chassis Cluster Control Ports" on page 76](#).
3. Connect the first device to be initialized in the cluster to the console port. This is the node (node 0) that forms the cluster and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster by giving it the cluster ID.
  - b. Identify the node by giving it its own node ID and then reboot the system.

See ["Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster " on page 45](#). For connection instructions, see the Getting Started Guide for your device

- a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.
4. Connect to the console port on the other device (node 1) and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.
5. Configure the management interfaces on the cluster. See ["Example: Configuring the Chassis Cluster Management Interface" on page 51](#).
6. Configure the cluster with the CLI. See the following topics:
  - a. [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster](#)
  - b. ["Example: Configuring the Chassis Cluster Fabric Interfaces" on page 65](#)
  - c. ["Example: Configuring Chassis Cluster Redundancy Groups" on page 87](#)
  - d. ["Example: Configuring Chassis Cluster Interface Monitoring" on page 187](#)
  - e. ["Example: Configuring Chassis Clustering on an SRX Series Devices" on page 122](#)
7. (Optional) Initiate manual failover. See ["Initiating a Chassis Cluster Manual Redundancy Group Failover" on page 261](#).
8. (Optional) Configure conditional route advertisement over redundant Ethernet interfaces. See ["Understanding Conditional Route Advertising in a Chassis Cluster" on page 156](#).
9. Verify the configuration. See ["Viewing a Chassis Cluster Configuration" on page 138](#).

If two nodes are connected in cluster, one node is elected as primary mode and its Routing Engine is running as primary. The Routing Engine in secondary node running as client. All FPCs in the cluster, regardless in primary node or secondary node, connect to the primary Routing Engine. The FPCs on secondary node connect to primary Routing Engine through the HA control link. If the cluster has two primaries, IOC receives a message from a different primary and reboot itself to recover from this error state.

To prevent the IOC card from rebooting, secondary node has to be powered off before connecting into the cluster.

To preserve the traffic on primary while connecting the secondary node into cluster, it is recommended to configure cluster mode on node 1 and power down before connecting it to the cluster to avoid any impact to the primary. The reason here is that control networks are different for a HA cluster or a single node system. When the control ports are connected, these two join the same network and they exchange messages.

This section provides an overview of the basic steps to restore the backup node after a failure when there is a running primary node:

1. Connect to the console port on the other device (node 1) and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.

See "[Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster](#)" on page 45. For connection instructions, see the Getting Started Guide for your device

2. Power off the secondary node.
3. Connect the HA control ports between two nodes.
4. Power on the secondary node.
5. The cluster is re-formed and the session is synced to the secondary node.
  - When using dual fabric link functionality, connect the two pairs of Ethernet interfaces that you will use on each device. See "[Understanding Chassis Cluster Dual Fabric Links](#)" on page 173.
  - When using dual control link functionality (SRX5600 and SRX5800 devices only), connect the two pairs of control ports that you will use on each device.

See "[Dual Control Link Connections for SRX Series Firewalls in a Chassis Cluster](#)" on page 165.

For SRX5600 and SRX5800 devices, control ports must be on corresponding slots in the two devices. [Table 2 on page 19](#) shows the slot numbering offsets:



Table 2: Slot Numbering Offsets

Device	Offset
SRX5800	12 (for example, <b>fpc3</b> and <b>fpc15</b> )
SRX5600	6 (for example, <b>fpc3</b> and <b>fpc9</b> )
SRX5400	3 (for example, <b>fpc3</b> and <b>fpc6</b> )
SRX4600	7 (for example, <b>fpc1</b> and <b>fpc8</b> )

- On SRX3400 and SRX3600 devices, the control ports are dedicated Gigabit Ethernet ports.
- On SRX4600 devices, the control ports and fabric ports are dedicated 10-Gigabit Ethernet ports.

#### RELATED DOCUMENTATION

[Chassis Cluster Overview](#) | 2

## Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming

#### IN THIS SECTION

- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, and SRX1600 devices.](#) | 20
- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX4600 Devices](#) | 25
- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX2300, SRX4100, SRX4200, and SRX4300 Devices](#) | 27

- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX5800, SRX5600, and SRX5400 Devices | 32](#)
- [FPC Slot Numbering in SRX Series Firewall Cards | 34](#)

See the hardware documentation for your particular model ([SRX Series Services Gateways](#)) for details about SRX Series Firewalls. See [Interfaces User Guide for Security Devices](#) for a full discussion of interface naming conventions.

After the devices are connected as a cluster, the slot numbering on the SRX acting as node 1 changes and thus the interface numbering will change. The slot number for each slot in both nodes is determined using the following formula:

**cluster slot number = (node ID \* maximum slots per node) + local slot number**

In chassis cluster mode, the interfaces on the SRX acting as node 1 are renumbered internally.

This topic describes the slot numbering and physical port and logical interface naming conventions for SRX Series Firewalls in a chassis cluster and includes following sections:

## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, and SRX1600 devices.

For SRX340 and SRX345 devices, the fxp0 interface is a dedicated port. For SRX300 and SRX320 devices, after you enable chassis clustering and reboot the system, the built-in interface named ge-0/0/0 is repurposed as the management interface and is automatically renamed fxp0.

For SRX300, SRX320, SRX340, SRX345, and SRX380 devices, after you enable chassis clustering and reboot the system, the built-in interface named ge-0/0/1 is repurposed as the control interface and is automatically renamed fxp1. The SRX300, SRX320, SRX340, SRX345 and SRX380 devices support only Gigabit Ethernet ports as fabric link.

SRX1500 devices have 16 GE interfaces and 4 XE ports.

SRX1600 devices have 16 (1G), 4 (10G), 2 (25G), 2 (1G) ports. The 2 (1G) ports are HA control ports and are not used for forwarding traffic.

[Table 3 on page 21](#) shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series Firewalls that become node 0 and node 1 of the chassis cluster after the cluster is formed.

**Table 3: Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming**

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX1600	Node 0	3	0	fxp0	Dedicated dual Control links with MACsec support	Dual fabric links
					em0/em1	fab0
	Node 1		7	fxp0	Dedicated dual Control links with MACsec support	Dual fabric links
					em0/em1	fab1
SRX1500	Node 0	3	0	fxp0	Dedicated Control port	Any Ethernet port
					em0	fab0
	Node 1		7	fxp0	Dedicated Control port	Any Ethernet port
					em0	fab1
SRX340,SRX 345, and SRX380	Node 0	5 (PIM slots)	0–4	fxp0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0

**Table 3: Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming (Continued)**

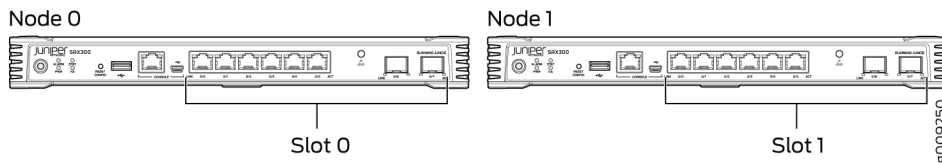
Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
	Node 1		5–9	fxp0	ge-5/0/1	Any Ethernet port
				fxp0	fxp1	fab1
SRX320	Node 0	3 (PIM slots)	0–2	ge-0/0/0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		3–5	ge-3/0/0	ge-3/0/1	Any Ethernet port
				fxp0	fxp1	fab1
SRX300	Node 0	1(PIM slot)	0	ge-0/0/0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		1	ge-1/0/0	ge-1/0/1	Any Ethernet port
				fxp0	fxp1	fab1

**Table 4: Chassis Cluster Fabric Interface Details for SRX1600**

Interfaces	Used as Fabric Port?	Supports Z-Mode Traffic?	Supports MACsec?
16X1Gigabit Ethernet Interface -BASE-T RJ45	Yes	Yes	No
2x 25G SFP28	Yes	Yes	No
4x 10G SFP+	Yes	Yes	No

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many slots. (See [Figure 3 on page 23](#), [Figure 4 on page 23](#), [Figure 5 on page 24](#), [Figure 6 on page 24](#), and [Figure 8 on page 24](#).)

**Figure 3: Slot Numbering in SRX300 Chassis Cluster**



**Figure 4: Slot Numbering in SRX320 Chassis Cluster**

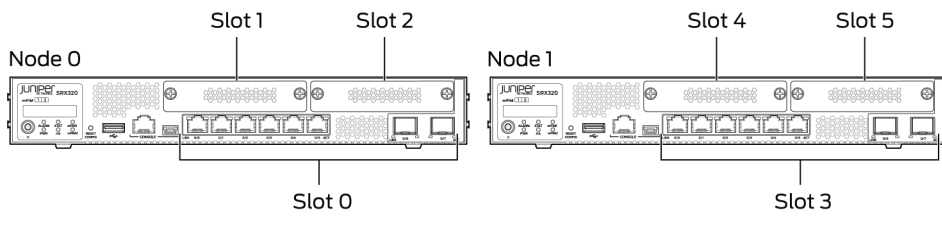


Figure 5: Slot Numbering in SRX340 Chassis Cluster

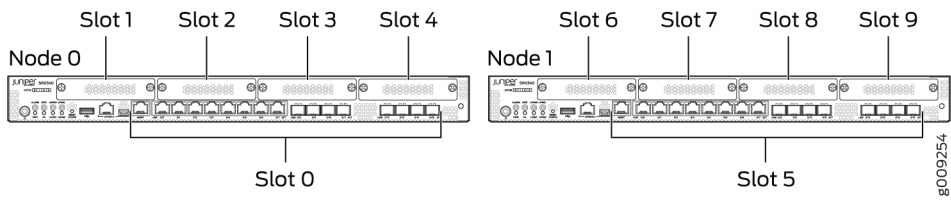


Figure 6: Slot Numbering in SRX345 Chassis Cluster

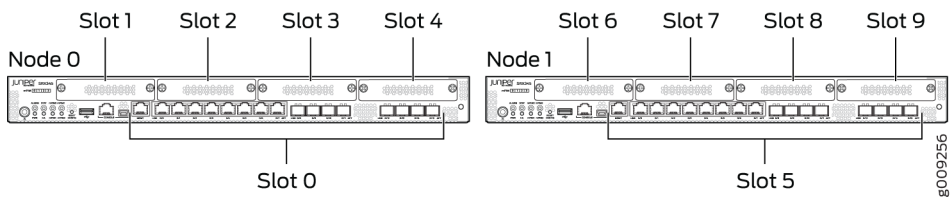


Figure 7: Slot Numbering in SRX380 Chassis Cluster

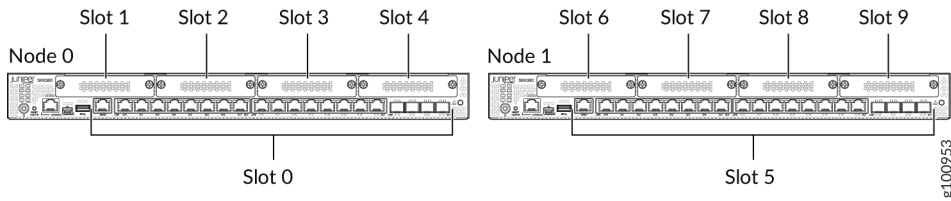


Figure 8: Slot Numbering in SRX1500 Chassis Cluster

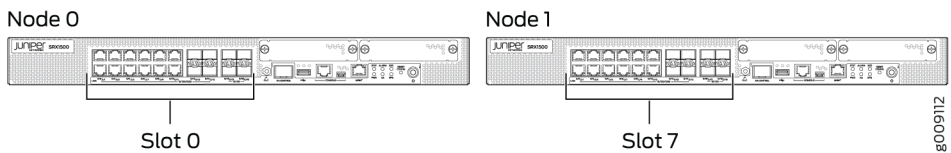
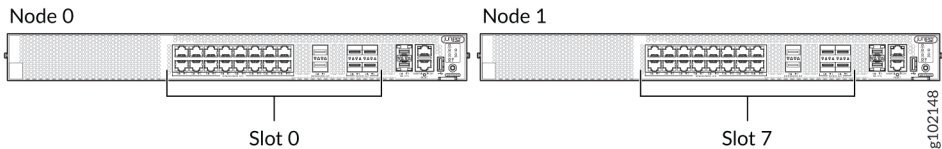


Figure 9: Slot Numbering in SRX1600 Chassis Cluster



## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX4600 Devices

The SRX4600 devices use dedicated HA control and fabric ports.

Table 5 on page 25 and Table 6 on page 26 show the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series Firewalls that become node 0 and node 1 of the chassis cluster after the cluster is formed.

For information on SRX3400 and SRX3600 devices, see *Chassis Cluster support on SRX3400 and SRX3600 devices*.

Table 5: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX4600 Devices

Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/ Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX4600	Node 0	1	0-6	fxp0	Dual (redundant) MACsec-enabled HA control ports (10GbE) are xe-0/0/0 and xe-0/0/1  It uses 1-Gigabit Ethernet SFP as control port.	Dual (redundant) MACsec-enabled HA fabric ports (10GbE)  Dual Fabric ports with macsec enabled are xe-0/0/2 and xe-0/0/3
	Node 1		7-13			

**Table 6: Chassis Cluster Interface Renumbering for SRX4600**

Device	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX4600	7	xe-1/0/0	xe-8/0/0

**Table 7: Chassis Cluster Fabric Interface Details for SRX4600**

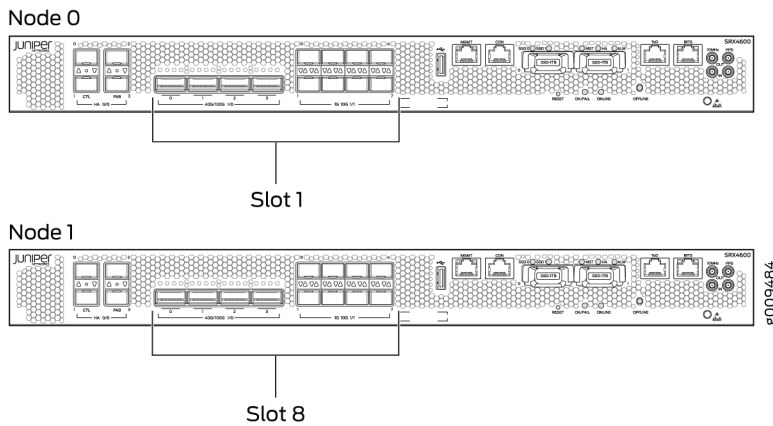
Interfaces	Used as Fabric Port?	Supports Z-Mode Traffic?	Supports MACsec?
Dedicated fabric ports	Yes	Yes	Yes
8X10-Gigabit Ethernet Interface SFPP ports	Yes	Yes	No
4X40-Gigabit Ethernet Interface QSFP28 ports	Yes	Yes	No
4x10-Gigabit Ethernet Interface SFPP ports	Yes	Yes	No
2X100-Gigabit Ethernet Interface QSFP28 slots	No	No	No

Mix and match of fabric ports are not supported. That is, you cannot use one 10-Gigabit Ethernet interface and one 40-Gigabit Ethernet interface for fabric links configuration. Dedicated fabric link supports only 10-Gigabit Ethernet Interface.

[Figure 10 on page 27](#) shows the slot numbering for both of the SRX Series Firewalls that become node 0 and node 1 of the chassis cluster after the cluster is formed.



Figure 10: Slot Numbering in SRX4600 Chassis Cluster



## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX2300, SRX4100, SRX4200, and SRX4300 Devices

The SRX4100 and SRX4200 devices use two 1-Gigabit Ethernet/10-Gigabit Ethernet ports, labeled as **CTL** and **FAB** as control port and fabric port respectively.

The SRX4300 devices supports 1-Gigabit Ethernet labeled as **CTL** control port.

Supported fabric interface types for SRX4100 and SRX4200 devices are 10-Gigabit Ethernet (xe) (10-Gigabit Ethernet Interface SFP+ slots).

Table 8: Chassis Cluster Fabric Interface Details for SRX2300

Interfaces	Used as Fabric Port?	Supports Z-Mode Traffic?	Supports MACsec?
8X10-Gigabit Ethernet Interface SFPP ports	Yes	Yes	Yes
4X25-Gigabit Ethernet Interface SFP28 ports	Yes	Yes	Yes
8x10-Gigabit Ethernet Interface BASE-T RJ45 ports	Yes	Yes	Yes

**Table 8: Chassis Cluster Fabric Interface Details for SRX2300 (Continued)**

Interfaces	Used as Fabric Port?	Supports Z-Mode Traffic?	Supports MACsec?
2X100-Gigabit Ethernet Interface QSFP28 slots	No	No	Yes

**Table 9: Chassis Cluster Fabric Interface Details for SRX4300**

Interfaces	Used as Fabric Port?	Supports Z-Mode Traffic?	Supports MACsec?
8X10-Gigabit Ethernet Interface SFP ports.	Yes	Yes	Yes
8x10-Gigabit Ethernet Interface BASE-T RJ45 ports.	Yes	Yes	Yes
6X100-Gigabit Ethernet Interface QSFP28 slots.	No	No	Yes
2X1-Gigabit Ethernet Interface SFP HA slots.	No	No	Yes
4X25-Gigabit Ethernet Interface SFP28 ports.	Yes	Yes	Yes

SRX4100, SRX4200, and SRX4300 devices do not support direct attach copper (DAC) cables for chassis cluster control.

[Table 10 on page 29](#) shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series Firewalls that become node 0 and node 1 of the chassis cluster after the cluster is formed

**Table 10: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX2300, SRX4100, SRX4200, and SRX4300 Devices**

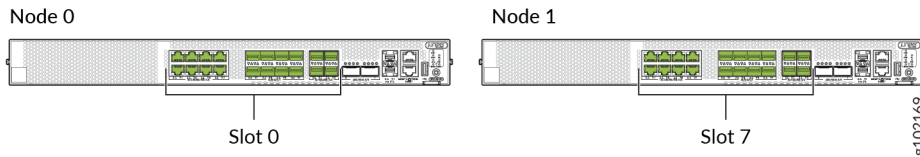
Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/ Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX2300	Node 0	1	0	fxp0	Dedicated control port, em0/em1	Revenue interfaces are used for dual fabric links, fab0.
	Node 1		7			Revenue interfaces are used for dual fabric links, fab1.
SRX4100	Node 0	1	0	fxp0	Dedicated control port, em0	Dedicated fabric port, any Ethernet port (for dual fabric-link), fab0
	Node 1		7			Dedicated fabric port, and any Ethernet port (for dual fabric-link), fab1
SRX4200	Node 0	1	0	fxp0	Dedicated control port,em0	Dedicated fabric port, and any Ethernet port (for dual fabric-link), fab0
	Node 1		7			Dedicated fabric port, and any Ethernet port (for dual fabric-link), fab1

**Table 10: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX2300, SRX4100, SRX4200, and SRX4300 Devices (Continued)**

Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/ Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX4300	Node 0	1	0	fxp0	Dedicated control port, em0/em1	Revenue interfaces are used for dual fabric links, fab0
	Node 1		7			Revenue interfaces for dual fabric links, fab1

Figure 12 on page 30 and Figure 13 on page 31 shows the slot numbering for both of the SRX Series Firewalls that become node 0 and node 1 of the chassis cluster after the cluster is formed.

**Figure 11: Slot Numbering in SRX2300**



**Figure 12: Slot Numbering in SRX4100 Chassis Cluster**

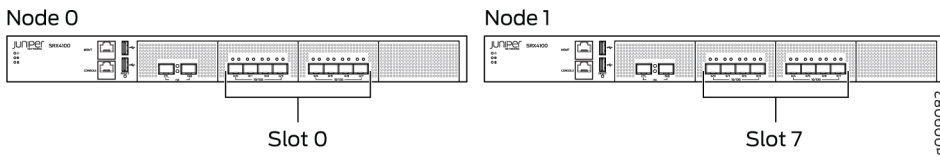


Figure 13: Slot Numbering in SRX4200 Chassis Cluster

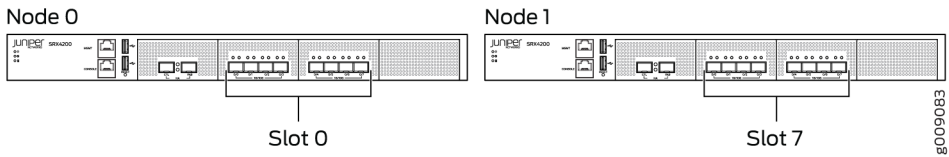
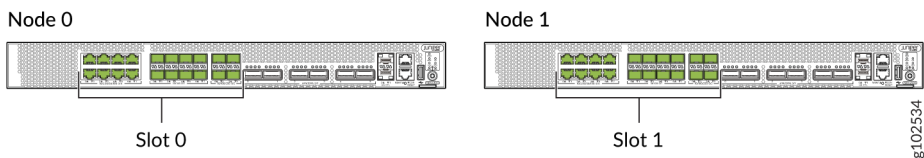


Figure 14: Slot Numbering in SRX4300 Chassis Cluster



The node 1 rennumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. For example, see [Table 11 on page 31](#) for interface renumbering on the SRX Series Firewalls (SRX4100, SRX4200, and SRX4300).

Table 11: Chassis Cluster Interface Renumbering for SRX1600, SRX2300, SRX4100, SRX4200, and SRX4300

Device	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX1600	7	xe-0/1/0	xe-7/1/0
SRX2300	7	xe-0/2/0	xe-7/2/0
SRX4100	7	xe-0/0/0	xe-7/0/0
SRX4200	7	xe-0/0/1	xe-7/0/1
SRX4300	7	xe-0/1/0	xe-7/1/0

On SRX4100 and SRX4200 devices, when the system comes up as chassis cluster, the xe-0/0/8 and xe-7/0/8 interfaces are automatically set as fabric interfaces links. You can set up another pair of fabric interfaces using any pair of 10-Gigabit interfaces to serve as the fabric between nodes. Note that, the automatically created fabric interfaces cannot be deleted. However, you can delete the second pair of fabric interfaces (manually configured interfaces).

## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX5800, SRX5600, and SRX5400 Devices

For chassis clustering, all SRX Series Firewalls have a built-in management interface named `fxp0`. For most SRX Series Firewalls, the `fxp0` interface is a dedicated port.

For the SRX5000 line, control interfaces are configured on SPCs.

Table 12 on page 32 shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series Firewalls that become node 0 and node 1 of the chassis cluster after the cluster is formed.

**Table 12: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX5000 Line Devices**

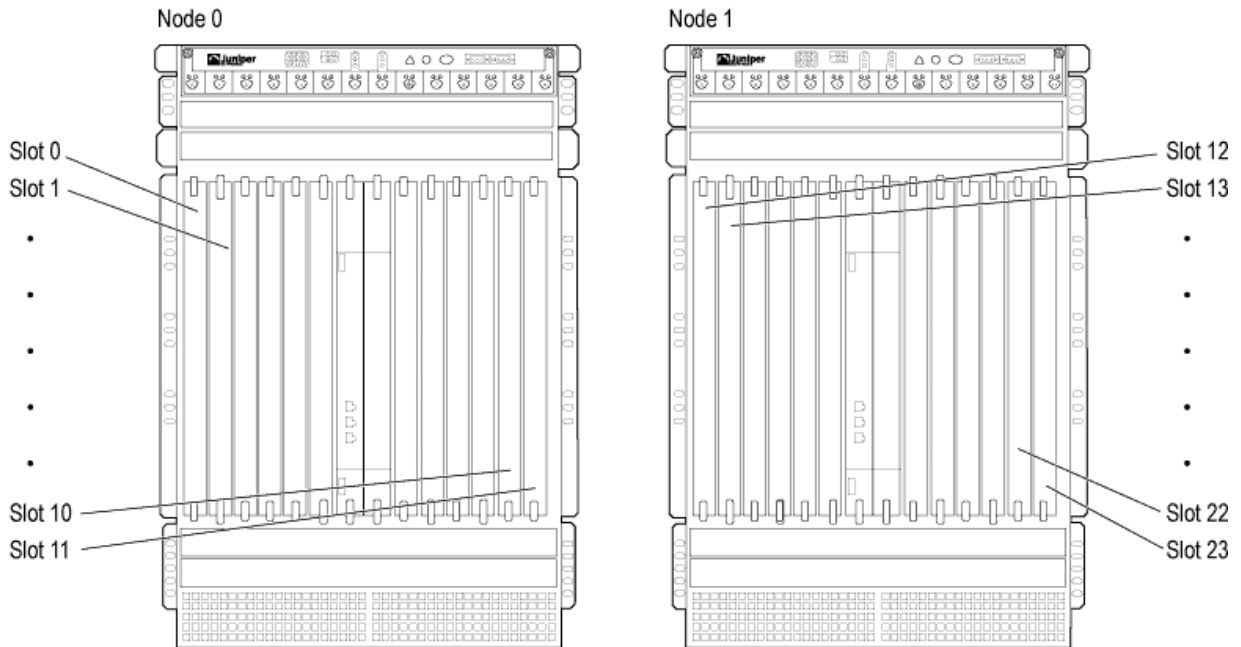
Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX5800	Node 0	12 (FPC slots)	0–11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				<code>fxp0</code>	<code>em0</code>	<code>fab0</code>
	Node 1		12–23	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				<code>fxp0</code>	<code>em0</code>	<code>fab1</code>
SRX5600	Node 0	6 (FPC slots)	0–5	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				<code>fxp0</code>	<code>em0</code>	<code>fab0</code>
	Node 1		6–11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				<code>fxp0</code>	<code>em0</code>	<code>fab1</code>

**Table 12: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX5000 Line Devices (Continued)**

Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/ Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX5400	Node 0	3 (FPC slots)	0–2	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1		3–5	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many slots. (See [Figure 15 on page 34.](#))

Figure 15: Slot Numbering in SRX5800 Chassis Cluster



## FPC Slot Numbering in SRX Series Firewall Cards

SRX5600 and SRX5800 devices have Flex I/O Cards (Flex IOCs) that have two slots to accept the following port modules:

- SRX-IOC-4XGE-XFP 4-Port XFP
- SRX-IOC-16GE-TX 16-Port RJ-45
- SRX-IOC-16GE-SFP 16-Port SFP

You can use these port modules to add from 4 to 16 Ethernet ports to your SRX Series Firewall. Port numbering for these modules is

`slot/port module/port`

where *slot* is the number of the slot in the device in which the Flex IOC is installed; *port module* is 0 for the upper slot in the Flex IOC or 1 for the lower slot when the card is vertical, as in an SRX5800 device; and *port* is the number of the port on the port module. When the card is horizontal, as in an SRX5400 or SRX5600 device, *port module* is 0 for the left-hand slot or 1 for the right-hand slot.



SRX5400 devices support only SRX5K-MPC cards. The SRX5K-MPC cards also have two slots to accept the following port modules:

- SRX-MIC-10XG-SFPP 10-port-SFP+ (xe)
- SRX-MIC-20GE-SFP 20-port SFP (ge)
- SRX-MIC-1X100G-CFP 1-port CFP (et)
- SRX-MIC-2X40G-QSFP 2-port QSFP (et)

See the hardware guide for your specific SRX Series model ([SRX Series Services Gateways](#)).

### RELATED DOCUMENTATION

| [Example: Configure Chassis Clustering on SRX Series Firewalls](#) | 122

## Preparing Your Equipment for Chassis Cluster Formation

To form a *chassis cluster*, a pair of the same kind of supported SRX Series Firewalls is combined to act as a single system that enforces the same overall security. SRX Series Firewalls must meet the following requirements to be included in a chassis clusters.

To form a *chassis cluster*, a pair of the same kind of supported SRX Series Firewalls is combined to act as a single system that enforces the same overall security.

- The network node redundancy is achieved by grouping a pair of the same kind of supported SRX Series Firewalls into a cluster.
- SRX Series Firewalls must be the same model.
- Junos OS requirements: Both the devices must be running the same Junos OS version
- Licensing requirements: Licenses are unique to each device and cannot be shared between the devices. Both devices (which are going to form chassis cluster) must have the identical features and license keys enabled or installed them. If both devices do not have an identical set of licenses, then after a failover, that particular licensed feature might not work or the configuration might not synchronize in chassis cluster formation.

- All services processing cards (SPCs), network processing cards (NPCs), and input/output cards (IOCs) on applicable SRX Series Firewalls must have the same slot placement and must be of same type.

Example:

- For SRX5400, SRX5600 and SRX5800 chassis clusters, the placement and the type of services processing cards (SPC, SPC2, SRX5K-SPC3), and input/output cards (IOC1, IOC2, IOC3, IOC4) must match in two devices. Only SCB4 is not supported on SRX5400. All other components are supported on SRX5400.
- For SRX3400 and SRX3600 chassis clusters, the placement and the type of SPCs, input/output cards (IOCs, NPIOCs), and network processing cards (NPCs) must match in two devices.

You can use the `show chassis hardware` command to identify the type of the card.

Following example shows the placement and the type of cards used in a chassis cluster setup:

```

user@host> show chassis hardware
node0:
-----
Hardware inventory:
Item          Version  Part number  Serial number  Description
Chassis                               JN1267B0FAGA  SRX5800
Midplane      REV 42   760-063937  ACRL3065      Enhanced SRX5800 Backplane
FPM Board     REV 05   760-061272  CAHE4860      Front Panel Display
PDM           Rev 01   740-063049  QCS2209509D   Power Distribution Module
PEM 0         Rev 04   740-034724  QCS171002016  PS 4.1kW; 200-240V AC in
PEM 1         Rev 11   740-027760  QCS1825N07S   PS 4.1kW; 200-240V AC in
Routing Engine 0 REV 01   750-095568  CALK8884      SRX5k RE-2000x6
Routing Engine 1 REV 01   750-095568  CADZ9076      SRX5k RE-2000x6
CB 0          REV 26   750-031391  CALV3002      SRX5k SCB4
CB 1          REV 26   750-031391  CALV3009      SRX5k SCB4
FPC 2         REV 28   750-073435  CALS4630      SPC3
  CPU                               BUILTIN      BUILTIN      SRX5k vCPP Broadwell
  PIC 0                               BUILTIN      BUILTIN      SPU Cp-Flow
  PIC 1                               BUILTIN      BUILTIN      SPU Flow
FPC 3         REV 17   750-044175  CABE7777      SRX5k SPC II
  CPU                               BUILTIN      BUILTIN      SRX5k DPC PPC
  PIC 0                               BUILTIN      BUILTIN      SPU Flow
  PIC 1                               BUILTIN      BUILTIN      SPU Flow
  PIC 2                               BUILTIN      BUILTIN      SPU Flow
  PIC 3                               BUILTIN      BUILTIN      SPU Flow
FPC 4         REV 08   750-061262  CAFD8147      SRX5k IOC II
  CPU         REV 02   711-061263  CAFV7488      SRX5k MPC PMB

```

MIC 0	REV 03	750-055732	CAFV9369	20x 1GE(LAN) SFP
PIC 0		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 0	REV 02	740-011613	PNB1GJR	SFP-SX
PIC 1		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
FPC 5	REV 10	750-062242	CAKX2328	SRX5k IOC3 2CGE+4XGE
PIC 0		BUILTIN	BUILTIN	2x 10GE SFP+
Xcvr 0	REV 01	740-021308	ANA07RE	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AQF0RBJ	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	1x 100GE CFP2
PIC 2		BUILTIN	BUILTIN	2x 10GE SFP+
Xcvr 0	REV 01	740-031980	AA1650304RF	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	AQ93BDK	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	1x 100GE CFP2
FPC 8	REV 46	750-056519	CALC4514	SRX5k IOC4 MRATE
CPU	REV 21	750-057177	CALC3494	SMPC PMB
PIC 0		BUILTIN	BUILTIN	MRATE-6xQSFP-XGE-XLGE-CGE
Xcvr 0	REV 01	740-059437	000T20128	QSFP28-LPBK
Xcvr 1	REV 01	740-067443	1ACP13450KH	QSFP+-40G-SR4
PIC 1		BUILTIN	BUILTIN	MRATE-6xQSFP-XGE-XLGE-CGE
Xcvr 0	REV 01	740-059437	0000T3443	QSFP28-LPBK
Fan Tray 0	REV 06	740-035409	ACAE9390	Enhanced Fan Tray
Fan Tray 1	REV 06	740-035409	ACAE9386	Enhanced Fan Tray

node1:

-----  
Hardware inventory:

Item	Version	Part number	Serial number	Description
Chassis			JN1267B01AGA	SRX5800
Midplane	REV 42	760-063937	ACRL3068	Enhanced SRX5800 Backplane
FPM Board	REV 05	760-061272	CAJX9988	Front Panel Display
PDM	Rev 01	740-063049	QCS2209507A	Power Distribution Module
PEM 0	Rev 11	740-027760	QCS1822N0EY	PS 4.1kW; 200-240V AC in
PEM 1	Rev 03	740-034724	QCS17020203F	PS 4.1kW; 200-240V AC in
Routing Engine 0	REV 01	750-095568	CALK8904	SRX5k RE-2000x6
Routing Engine 1	REV 01	750-095568	CADZ9076	SRX5k RE-2000x6
CB 0	REV 26	750-031391	CALV3010	SRX5k SCB4
CB 1	REV 26	750-031391	CALV3000	SRX5k SCB4
FPC 2	REV 28	750-073435	CAKZ9620	SPC3
CPU		BUILTIN	BUILTIN	SRX5k vCPP Broadwell
PIC 0		BUILTIN	BUILTIN	SPU Cp-Flow
PIC 1		BUILTIN	BUILTIN	SPU Flow
FPC 3	REV 18	750-054877	CACH4082	SRX5k SPC II
CPU		BUILTIN	BUILTIN	SRX5k DPC PPC

PIC 0		BUILTIN	BUILTIN	SPU Flow
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
FPC 4	REV 08	750-061262	CAFD8165	SRX5k IOC II
CPU	REV 02	711-061263	CAFV7507	SRX5k MPC PMB
MIC 0	REV 03	750-055732	CAFV6603	20x 1GE(LAN) SFP
PIC 0		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 0	REV 01	740-011613	AM0805S8M4N	SFP-SX
PIC 1		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
FPC 5	REV 03	750-062242	CAFZ2748	SRX5k IOC3 2CGE+4XGE
PIC 0		BUILTIN	BUILTIN	2x 10GE SFP+
Xcvr 0	REV 01	740-021308	11T511100788	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AS92WJ0	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	1x 100GE CFP2
PIC 2		BUILTIN	BUILTIN	2x 10GE SFP+
Xcvr 0	REV 01	740-031980	AA1650304EZ	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	ANS0EAR	SFP+-10G-SR
PIC 3		BUILTIN	BUILTIN	1x 100GE CFP2
FPC 8	REV 46	750-056519	CALC4526	SRX5k IOC4 MRATE
CPU	REV 21	750-057177	CALF5727	SMPC PMB
PIC 0		BUILTIN	BUILTIN	MRATE-6xQSFPP-XGE-XLGE-CGE
Xcvr 1	REV 01	740-067443	1ACP13450L9	QSFPP+-40G-SR4
PIC 1		BUILTIN	BUILTIN	MRATE-6xQSFPP-XGE-XLGE-CGE
Fan Tray 0	REV 06	740-035409	ACAE9298	Enhanced Fan Tray
Fan Tray 1	REV 06	740-035409	ACAE9314	Enhanced Fan Tray

- SRX1600, SRX2300 and SRX4300—Devices support dual control link and dual fab link.
- SRX4600—Has dedicated slots for each kind of card that cannot be interchanged.
- SRX300, SRX320, SRX340, SRX345, and SRX380—Although the devices must be of the same type, they can contain different Physical Interface Modules (PIMs).
- The control ports on the respective nodes are connected to form a control plane that synchronizes the configuration and kernel state to facilitate the high availability of interfaces and services.
- The data plane on the respective nodes is connected over the fabric ports to form a unified data plane. The fabric link allows for the management of cross-node flow processing and for the management of session redundancy.

## RELATED DOCUMENTATION

[Chassis Cluster Overview | 2](#)

[Understanding Chassis Cluster Fabric Interfaces | 60](#)

# Connecting SRX Series Firewalls to Create a Chassis Cluster

An SRX Series chassis cluster is created by physically connecting two identical cluster-supported SRX Series Firewalls together using a pair of the same type of Ethernet connections. The connection is made for both a control link and a fabric (data) link between the two devices.

Control links in a chassis cluster are made using specific ports.

The interface value changes with the cluster offset value. Based on the cluster index, the interface is named as type-fpc/pic/port. For example, ge-1/0/1, where 1 is cluster index and the FPC number. You must use the following ports to form the control link on the following SRX Series Firewalls:

- For SRX300 devices, connect the ge-0/0/1 on node 0 to the ge-1/0/1 on node 1.
- For SRX320 devices, connect the ge-0/0/1 on node 0 to the ge-3/0/1 on node 1.
- For SRX340, SRX345, and SRX380 devices, connect the ge-0/0/1 on node 0 to the ge-5/0/1 on node 1.
- For SRX1500 devices, connect the HA control port on node 0 to the HA control port on node 1.
- For SRX1600, SRX2300, and SRX4300 devices dual control link configuration, connect the HA control port 0 on node 0 to the HA control port 0 on node 1 and connect the HA control port 1 on node 0 to the control port 1 on node 1.

To establish a fabric link:

- For SRX300 and SRX320 devices, connect any interface except ge-0/0/0 and ge-0/0/1.
- For SRX340, SRX345, and SRX380 devices, connect any interface except fxp0 and ge-0/0/1.

[Figure 17 on page 40](#), [Figure 18 on page 40](#), [Figure 19 on page 40](#), and [Figure 21 on page 41](#) show pairs of SRX Series Firewalls with the fabric links and control links connected.

Figure 16: Connecting SRX300 Devices in a Chassis Cluster

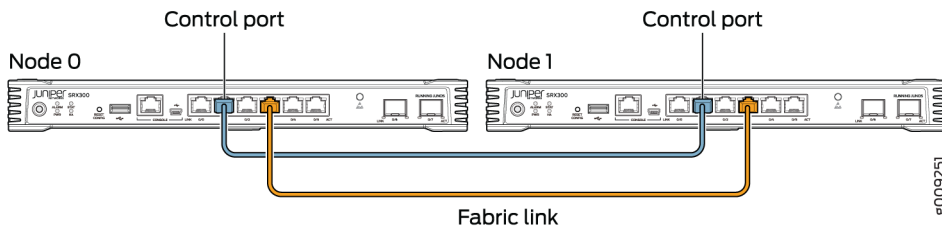


Figure 17: Connecting SRX320 Devices in a Chassis Cluster

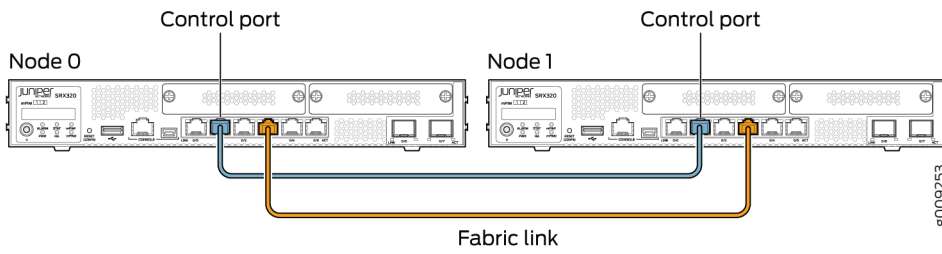


Figure 18: Connecting SRX340 Devices in a Chassis Cluster

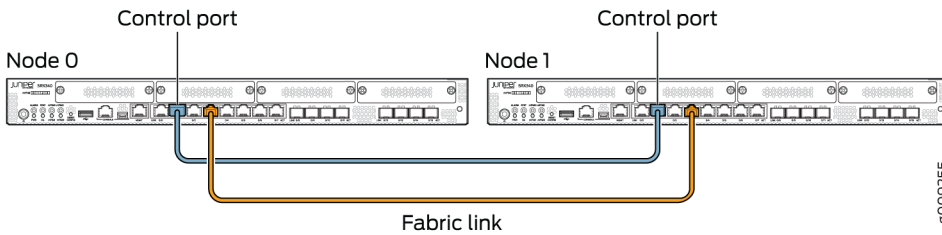
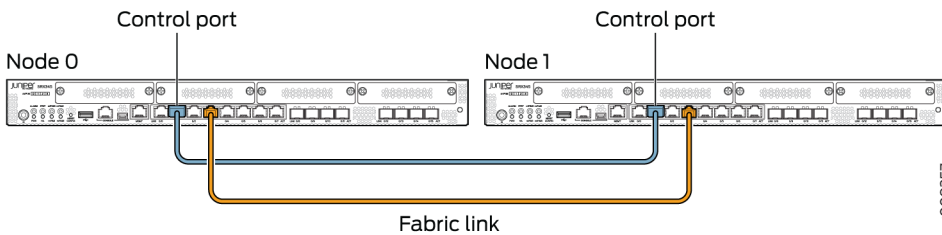
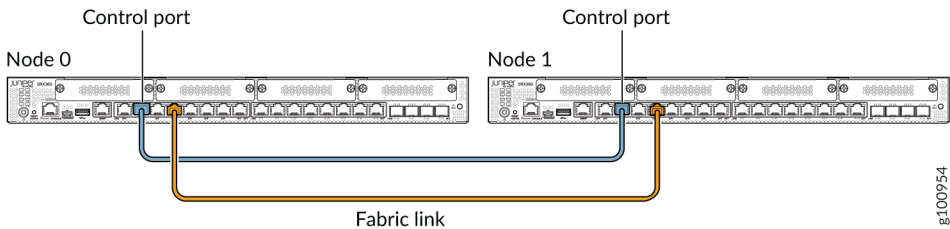


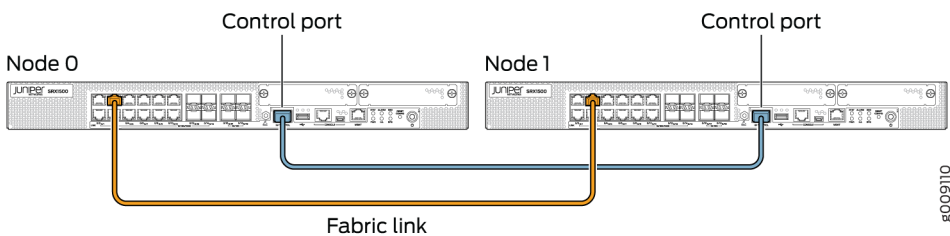
Figure 19: Connecting SRX345 Devices in a Chassis Cluster



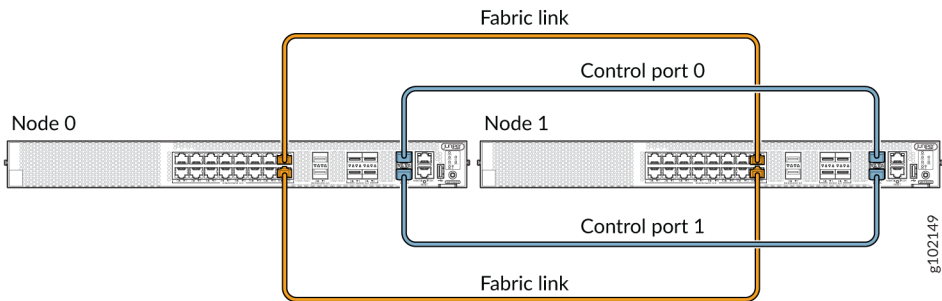
**Figure 20: Connecting SRX380 Devices in a Chassis Cluster**



**Figure 21: Connecting SRX1500 Devices in a Chassis Cluster**



**Figure 22: Connecting SRX1600 Devices in a Chassis Cluster**



For SRX1500, SRX1600, SRX2300, and SRX4300 devices, the connection that serves as the control link must be between the built-in control ports on each device.

You can connect two control links (SRX4600, SRX5600, SRX5800, and SRX3000 lines only) and two fabric links between the two devices in the cluster to reduce the chance of control link and fabric link failure. See ["Understanding Chassis Cluster Dual Control Links" on page 163](#) and ["Understanding Chassis Cluster Dual Fabric Links" on page 173](#).

[Figure 23 on page 42](#), [Figure 25 on page 42](#) and [Figure 26 on page 43](#) show pairs of SRX Series Firewalls with the fabric links and control links connected.

Figure 23: Connecting SRX4600 Devices in a Chassis Cluster

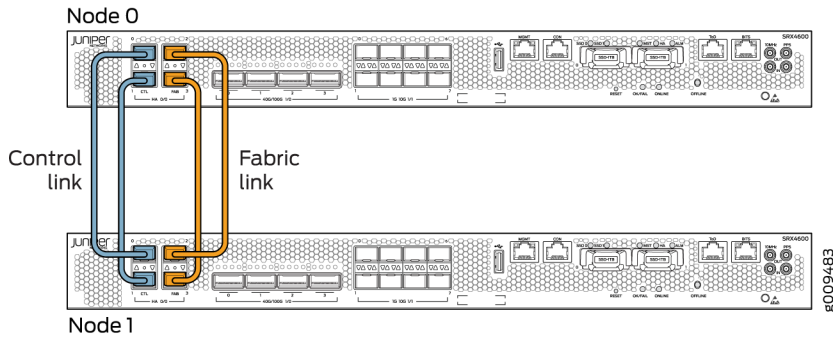


Figure 24: Connecting SRX2300 Devices in a Chassis Cluster

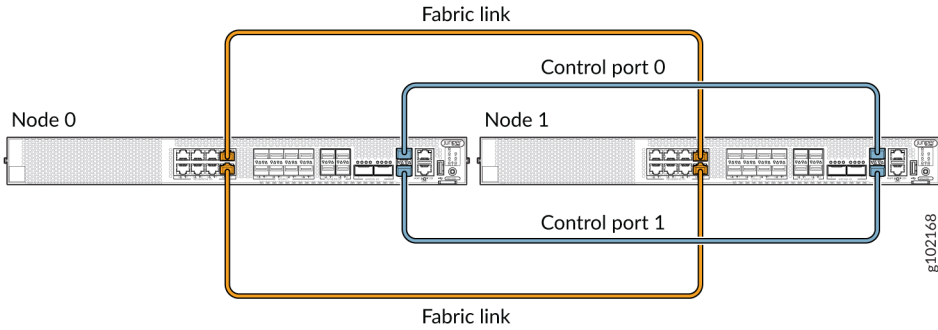


Figure 25: Connecting SRX4100 Devices in a Chassis Cluster

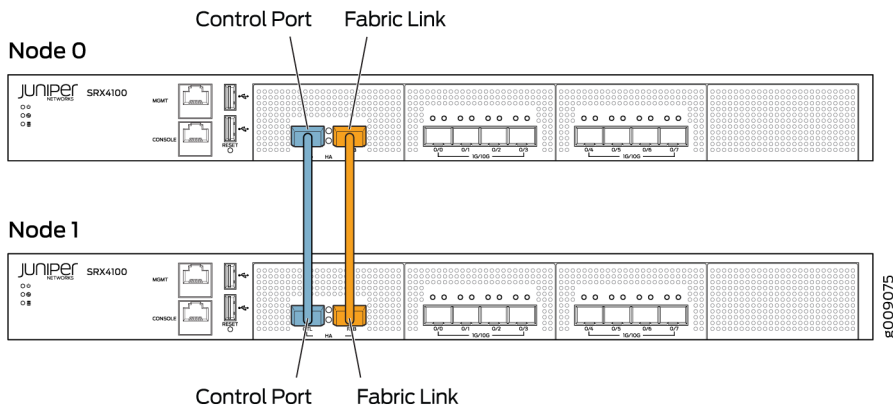




Figure 26: Connecting SRX4200 Devices in a Chassis Cluster

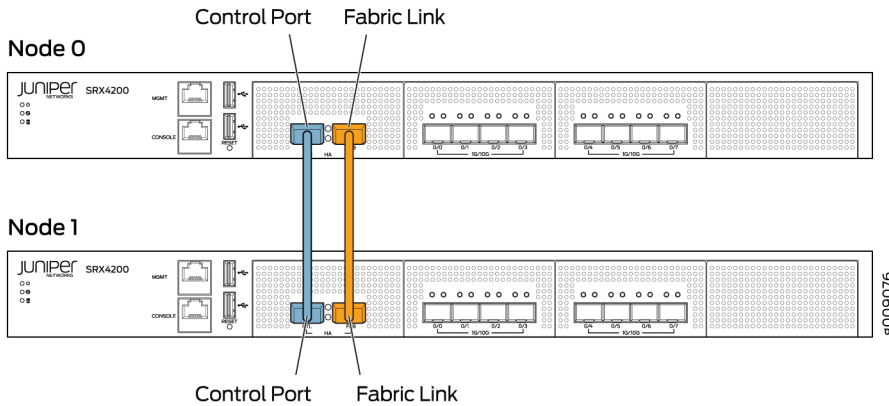


Figure 27: Connecting SRX4300 Devices in a Chassis Cluster

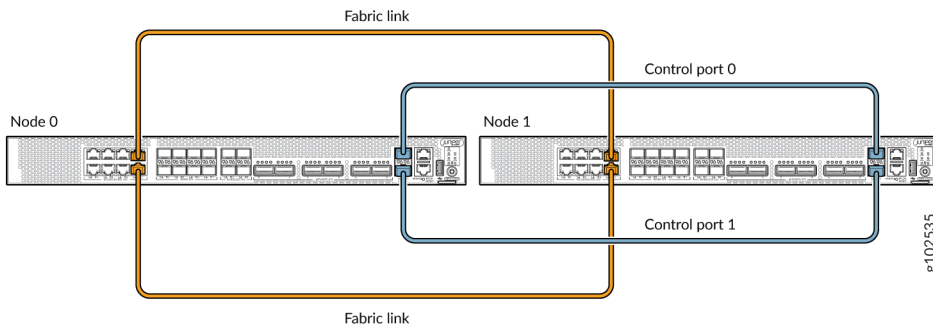


Figure 28 on page 44, Figure 29 on page 44, and Figure 30 on page 45 show pairs of SRX Series Firewalls with the fabric links and control links connected.

Service Processing Cards (SPC) have two dedicated ports ( HA0 and HA1) for connecting the control links in the chassis cluster.

Fabric ports are revenue ports available from any IOC card. Fabric links are connected to the same slot and port on both SRX5000 line of devices.

SRX5000 line devices do not have built-in ports, so the control link for these gateways must be the control ports on their SPCs with a slot numbering offset of 3 for SRX5400, offset of 6 for SRX5600 devices and 12 for SRX5800 devices.

Figure 28 on page 44 shows pair of SRX5800 devices having single SPC card each connected with a control link. The fabric link is connected using the IOC card. For dual control links, two SPC3 cards are required for connection.

Figure 28: Connecting SRX5800 Devices in a Chassis Cluster

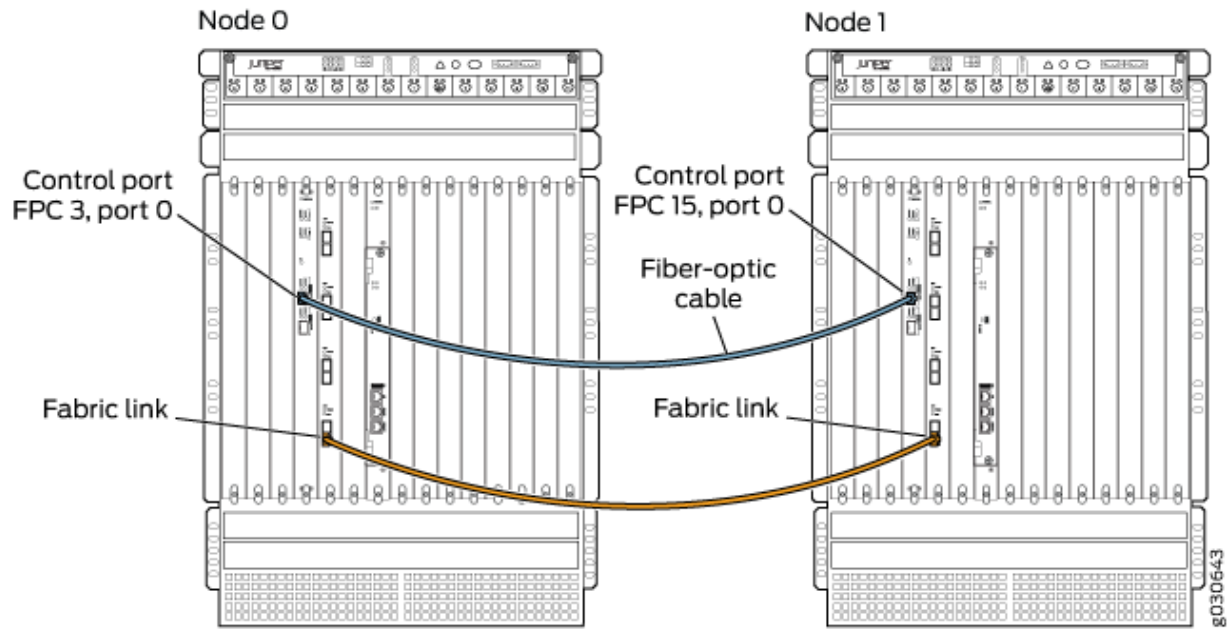
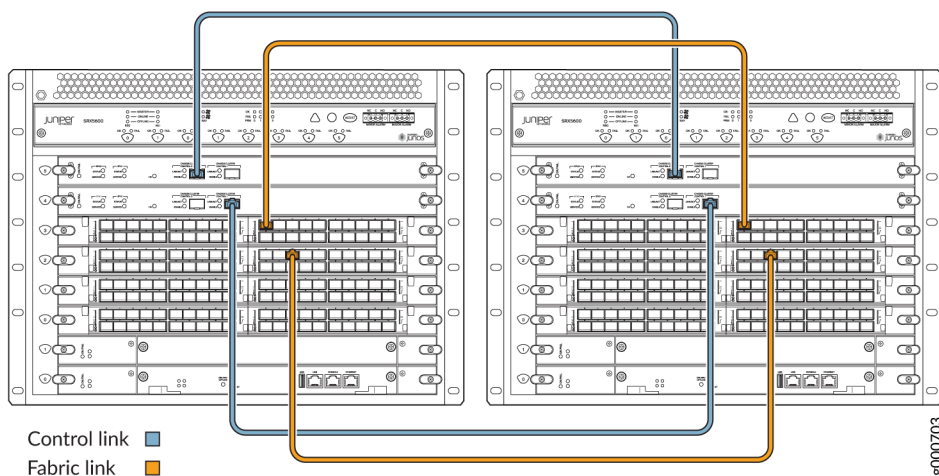


Figure 29 on page 44 shows dual control links connected using two SPC3 cards and dual fabric links using IOC cards.

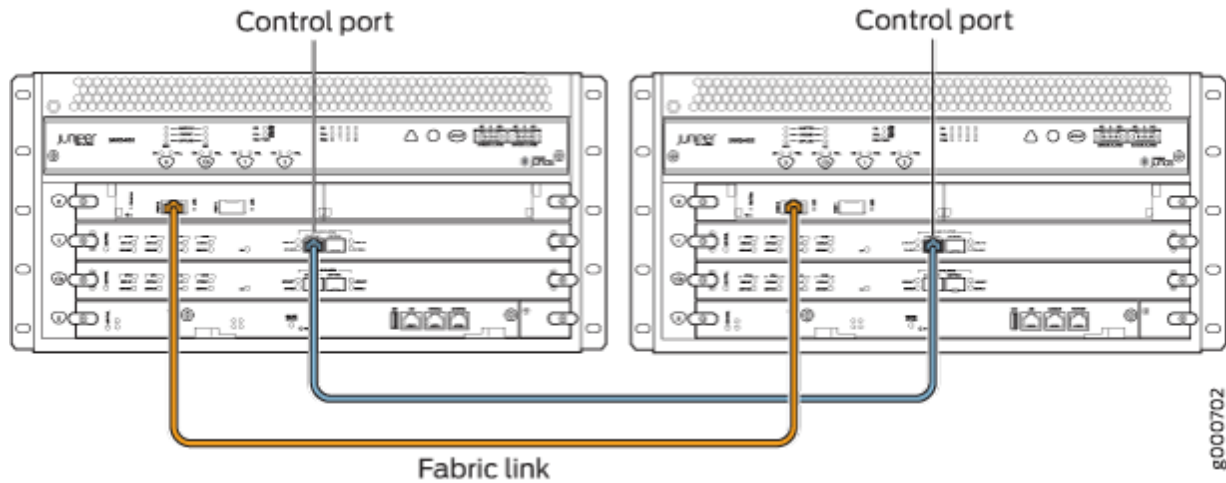
Figure 29: Connecting SRX5600 Devices in a Chassis Cluster



When you connect a single control link on SRX5000 line devices, the control link ports are a one-to-one mapping with the Routing Engine slot. If your Routing Engine is in slot 0, you must use control port 0 to link the Routing Engines.

When a SPC is the control plane as well as hosting the control port, this creates a single point of failure. If the SPC goes down on the primary node, the node is automatically rebooted to avoid split brain.

**Figure 30: Connecting SRX5400 Devices in a Chassis Cluster**



Dual control links are not supported on an SRX5400 device due to the limited number of slots.

#### RELATED DOCUMENTATION

[SRX Series Chassis Cluster Configuration Overview | 14](#)

[Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster | 45](#)

[Example: Configuring the Chassis Cluster Management Interface | 51](#)

[Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster](#)

## Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster

#### IN THIS SECTION

● [Requirements | 46](#)

● [Overview | 47](#)

- Configuration | 47
- Verification | 48

When a device joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

- A cluster is identified by a *cluster ID* (`cluster-id`) specified as a number from 1 through 255. Setting a cluster ID to 0 is equivalent to disabling a cluster. A cluster ID greater than 15 can only be set when the fabric and control link interfaces are connected back-to-back or connected on separate VLANs.

To use extended cluster IDs without back-to-back connectivity, control and fabric link traffic for each SRX cluster must be separated using unique VLAN IDs.

#### Example: Extended Cluster ID

The following message is displayed when you try to set a cluster ID greater than 15, and when fabric and control link interfaces are not connected back-to-back or are not connected on separate VLANs:

```
{primary:node1}
user@host> set chassis cluster cluster-id 254 node 1 reboot
For cluster-ids greater than 15 and when deploying more than one cluster in a single Layer 2
BROADCAST domain, it is mandatory that fabric and control links are either connected back-to-
back or are connected on separate private VLANS.
```

- A cluster node is identified by a *node ID* (`node`) specified as a number from 0 through 1.

This example shows how to set the chassis cluster node ID and chassis cluster ID, which you must configure after connecting two devices together. A chassis cluster ID identifies the cluster to which the devices belong, and a chassis cluster node ID identifies a unique node within the cluster. After wiring the two devices together, you use CLI *operational mode* commands to enable chassis clustering by assigning a cluster ID and node ID on each chassis in the cluster. The cluster ID is the same on both nodes.

## Requirements

Before you begin, ensure that you can connect to each device through the console port.

Ensure that the devices are running the same version of the Junos operating system (Junos OS) and the security devices are of same model.

The factory-default configuration of an SRX Series Firewall includes the configuration of the interfaces on the device. Therefore, before enabling chassis clustering on the device, you must remove any existing configuration associated with those interfaces that will be transformed into the control and fabric interfaces. See "[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming](#)" on page 19 for more information.

## Overview

The system uses the chassis cluster ID and chassis cluster node ID to apply the correct configuration for each node (for example, when you use the `apply-groups` command to configure the chassis cluster management interface). The chassis cluster ID and node ID statements are written to the EPROM, and the statements take effect when the system is rebooted.

In this example, you configure a chassis cluster ID of 1. You also configure a chassis cluster node ID of 0 for the first node, which allows redundancy groups to be primary on this node when priority settings for both nodes are the same, and a chassis cluster node ID of 1 for the other node.

Chassis cluster supports automatic synchronization of configurations. When a secondary node joins a primary node and a chassis cluster is formed, the primary node configuration is automatically copied and applied to the secondary node. See "[Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes](#)" on page 142.

## Configuration

### IN THIS SECTION

- [Procedure | 47](#)

### Procedure

#### Step-by-Step Procedure

To specify the chassis cluster node ID and cluster ID, you need to set two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices:

1. Connect to the first device through the console port.

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
Successfully enabled chassis cluster. Going to reboot now.
```

2. Connect to the second device through the console port.

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
Successfully enabled chassis cluster. Going to reboot now.
```

For SRX5400, SRX5600 and SRX5800 devices, you must configure the control ports before the cluster is formed.

To do this, you connect to the console port on the primary device, give it a node ID, and identify the cluster it will belong to, and then reboot the system. You then connect the console port to the other device, give it a node ID, and assign it the same cluster ID you gave to the first node, and then reboot the system. In both instances, you can cause the system to boot automatically by including the `reboot` parameter in the CLI command line. (For further explanation of primary and secondary nodes, see ["Understanding Chassis Cluster Redundancy Groups" on page 83.](#))

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Status | 48](#)

## Verifying Chassis Cluster Status

### Purpose

Verify the status of a chassis cluster.

## Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}[edit]
user@host> show chassis cluster status

Cluster ID: 1
Node          Priority      Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0        100          primary   no       no
  node1         1            secondary no       no

Redundancy group: 1 , Failover count: 1
  node0         0            primary   no       no
  node1         0            secondary no       no
```

## Meaning

The sample output shows that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the secondary node.

## RELATED DOCUMENTATION

[SRX Series Chassis Cluster Configuration Overview | 14](#)

[Example: Configuring the Chassis Cluster Management Interface | 51](#)

[Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster](#)

# Chassis Cluster Management Interfaces

## IN THIS SECTION

● [Understanding Management Interface on an Active Chassis Cluster | 50](#)

- [Example: Configuring the Chassis Cluster Management Interface | 51](#)

On SRX Series Firewalls in a chassis cluster, management interfaces allow out-of-band network access and network management to each node in the cluster. For more information, see the following topics:

## Understanding Management Interface on an Active Chassis Cluster

Most of SRX Series Firewalls contain an `fxp0` interface. The `fxp0` interfaces function like standard management interfaces on SRX Series Firewalls and allow network access to each node in the cluster.

Management interfaces are the primary interfaces for accessing the device remotely. Typically, a management interface is not connected to the in-band network, but is connected instead to the device's internal network. Through a management interface you can access the device over the network using utilities such as `ssh` and `telnet` and configure the device from anywhere, regardless of its physical location. SNMP can use the management interface to gather statistics from the device. A management interface enables authorized users and management systems connect to the device over the network.

Some SRX Series Firewalls have a dedicated management port on the front panel. For other types of platforms, you can configure a management interface on one of the network interfaces. This interface can be dedicated to management or shared with other traffic. Before users can access the management interface, you must configure it. Information required to set up the management interface includes its IP address and prefix. In many types of Junos OS devices (or recommended configurations), it is not possible to route traffic between the management interface and the other ports. Therefore, you must select an IP address in a separate (logical) network, with a separate prefix (netmask).

For most SRX Series chassis clusters, the `fxp0` interface is a dedicated port. SRX340 and SRX345 devices contain an `fxp0` interface. SRX300 and SRX320 devices do not have a dedicated port for `fxp0`. The `fxp0` interface is repurposed from a built-in interface. The `fxp0` interface is created when the system reboots the devices after you designate one node as the primary device and the other as the secondary device.

We recommend giving each node in a chassis cluster a unique IP address for the `fxp0` interface of each node. This practice allows independent node management.



## Example: Configuring the Chassis Cluster Management Interface

### IN THIS SECTION

- Requirements | 51
- Overview | 51
- Configuration | 52
- Verification | 58

This example shows how to provide network management access to a chassis cluster.

### Requirements

Before you begin, set the chassis cluster node ID and cluster ID. See ["Example: Setting the Chassis Cluster Node ID and Cluster ID" on page 45](#).

### Overview

You must assign a unique IP address to each node in the cluster to provide network management access. This configuration is not replicated across the two nodes.

If you try to access the nodes in a cluster over the network before you configure the fxp0 interface, you will lose access to the cluster.

In this example, you configure the following information for IPv4:

- Node 0 name—node0-router
- IP address assigned to node 0—10.1.1.1/24
- Node 1 name—node1-router
- IP address assigned to node 1—10.1.1.2/24

In this example, you configure the following information for IPv6:

- Node 0 name—node0-router
- IP address assigned to node 0—2001:db8:1::2/32
- Node 1 name—node1-router
- IP address assigned to node 1—2001:db8:1::3/32

## Configuration

### IN THIS SECTION

- [Configuring the Chassis Cluster Management Interface with IPv4 Addresses | 52](#)
- [Verifying the Chassis Cluster Management Interface Configuration \(IPv4 Addresses\) | 54](#)
- [Configuring the Chassis Cluster Management Interface with IPv6 Addresses | 55](#)

### Configuring the Chassis Cluster Management Interface with IPv4 Addresses

#### 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.

To configure a chassis cluster management interface for IPv4:

```
{primary:node0}[edit]
user@host#
set groups node0 system host-name node0-router
set groups node0 interfaces fxp0 unit 0 family inet address 10.1.1.1/24
set groups node1 system host-name node1-router
set groups node1 interfaces fxp0 unit 0 family inet address 10.1.1.2/24
set apply-groups "${node}"
```

#### Step-by-Step Procedure

To configure a chassis cluster management interface for IPv4:

1. Configure the name of node 0 and assign an IP address.

```
{primary:node0}[edit]
user@host# set groups node0 system host-name node0-router
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.1.1.1/24
```

2. Configure the name of node 1 and assign an IP address.

```
{primary:node0}[edit]
user@host# set groups node1 system host-name node1-router
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 10.1.1.2/24
```

3. Apply the groups configuration to the nodes.

```
{primary:node0}[edit]
user@host# set apply-groups "${node}"
```

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

```
{primary:node0}[edit]
user@host# commit
```

## Results

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

```
{primary:node0}[edit]
user@host# show groups
node0 {
  system {
    host-name node0-router;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet {
          address 10.1.1.1/24;
        }
      }
    }
  }
}
node1 {
```

```

system {
    host-name node1-router;
}
interfaces {
    fxp0 {
        unit 0 {
            family inet {
                address 10.1.1.2/24;
            }
        }
    }
}
}

```

```

{primary:node0}[edit]
user@host# show apply-groups
## Last changed: 2010-09-16 11:08:29 UTC
apply-groups "${node}";

```

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

## Verifying the Chassis Cluster Management Interface Configuration (IPv4 Addresses)

### Purpose

Verify the chassis cluster management interface configuration.

### Action

To verify the configuration is working properly, enter the `show interfaces terse`, `show configuration groups node node0 interfaces` and `show configuration groups node node1 interfaces` commands.

```

{primary:node0} [edit]
user@host> show interfaces terse | match fxp0

```

```
fxp0          up    up
fxp0.0       up    up    inet    10.1.1.1/24
```

```
{primary:node0} [edit]
user@host> show configuration groups node0 interfaces
```

```
fxp0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
```

```
{primary:node0} [edit]
user@host> show configuration groups node1 interfaces
```

```
fxp0 {
  unit 0 {
    family inet {
      address 10.1.1.2/24;
    }
  }
}
```

## Meaning

The output displays the management interface information with their status.

## Configuring the Chassis Cluster Management Interface with IPv6 Addresses

### 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.

To configure a chassis cluster management interface for IPv6:

```
{primary:node0}[edit]
user@host#
set groups node0 system host-name node0-router
set groups node0 interfaces fxp0 unit 0 family inet6 address 2001:db8:1::2/32
set groups node1 system host-name node1-router
set groups node1 interfaces fxp0 unit 0 family inet6 address 2001:db8:1::3/32
```

## Step-by-Step Procedure

To configure a chassis cluster management interface for IPv6:

1. Configure the name of node 0 and assign an IP address.

```
{primary:node0}[edit]
user@host# set groups node0 system host-name node0-router
user@host# set groups node0 interfaces fxp0 unit 0 family inet6 address 2001:db8:1::2/32
```

2. Configure the name of node 1 and assign an IP address.

```
{primary:node0}[edit]
user@host# set groups node1 system host-name node1-router
user@host# set groups node1 interfaces fxp0 unit 0 family inet6 address 2001:db8:1::3/32
```

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

```
{primary:node0}[edit]
user@host# commit
```

## Results

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

```
{primary:node0}[edit]
user@host# show groups
node0 {
  system {
    host-name node0-router;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet6 {
          address 2001:db8:1::2/32;
        }
      }
    }
  }
}
node1 {
  system {
    host-name node1-router;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet6 {
          address 2001:db8:1::3/32;
        }
      }
    }
  }
}
}
```

```
{primary:node0}[edit]
user@host# show apply-groups
```

```
## Last changed: 2010-09-16 11:08:29 UTC
apply-groups "${node}";
```

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

## Verification

### IN THIS SECTION

- [Verifying the Chassis Cluster Management Interface Configuration \(IPv6 Addresses\) | 58](#)

## Verifying the Chassis Cluster Management Interface Configuration (IPv6 Addresses)

### Purpose

Verify the chassis cluster management interface configuration.

### Action

To verify the configuration is working properly, enter the `show interfaces terse` and `show configuration groups node0 interfaces` commands.

```
{primary:node0} [edit]
user@host> show interfaces terse | match fxp0

fxp0                up    up
fxp0.0              up    up    inet    2001:db8:1::2/32
```

```
{primary:node0} [edit]
user@host> show configuration groups node0 interfaces

fxp0 {
  unit 0 {
    family inet {
      address 2001:db8:1::2/32;
    }
  }
}
```



```
}  
}
```

```
{primary:node0} [edit]  
user@host> show configuration groups node1 interfaces  
  
fxp0 {  
  unit 0 {  
    family inet {  
      address 2001:db8:1::3/32;  
    }  
  }  
}
```

## Meaning

The output displays the management interface information with their status.

## SEE ALSO

| [Example: Configure Chassis Clustering on SRX Series Firewalls | 122](#)

## RELATED DOCUMENTATION

| [Chassis Cluster Fabric Interfaces | 59](#)

| [Chassis Cluster Control Plane Interfaces | 72](#)

# Chassis Cluster Fabric Interfaces

## IN THIS SECTION

- [Understanding Chassis Cluster Fabric Interfaces | 60](#)
- [Example: Configuring the Chassis Cluster Fabric Interfaces | 65](#)

- [Verifying Chassis Cluster Data Plane Interfaces | 69](#)
- [Viewing Chassis Cluster Data Plane Statistics | 69](#)
- [Clearing Chassis Cluster Data Plane Statistics | 70](#)

SRX Series devices in a chassis cluster use the fabric (fab) interface for session synchronization and forward traffic between the two chassis. The fabric link is a physical connection between two Ethernet interfaces on the same LAN. Both interfaces must be the same media type. For more information, see the following topics:

## Understanding Chassis Cluster Fabric Interfaces

### IN THIS SECTION

- [Supported Fabric Interface Types for SRX Series Firewalls \(SRX300 Series, SRX1500, SRX1600, SRX2300, SRX4100/SRX4200, SRX4300, SRX4600, and SRX5000 line\) | 61](#)
- [Jumbo Frame Support | 62](#)
- [Understanding Fabric Interfaces on SRX5000 Line Devices for IOC2 and IOC3 | 62](#)
- [Understanding Session RTOs | 63](#)
- [Understanding Data Forwarding | 64](#)
- [Understanding Fabric Data Link Failure and Recovery | 64](#)

The fabric is a physical connection between two nodes of a cluster and is formed by connecting a pair of Ethernet interfaces back-to-back (one from each node).

Unlike for the control link, whose interfaces are determined by the system, you specify the physical interfaces to be used for the fabric data link in the configuration.

The fabric is the data link between the nodes and is used to forward traffic between the chassis. Traffic arriving on a node that needs to be processed on the other is forwarded over the fabric data link. Similarly, traffic processed on a node that needs to exit through an interface on the other node is forwarded over the fabric.

The data link is referred to as the fabric interface. It is used by the cluster's Packet Forwarding Engines to transmit transit traffic and to synchronize the data plane software's dynamic runtime state. The fabric provides for synchronization of session state objects created by operations such as authentication, Network Address Translation (NAT), Application Layer Gateways (ALGs), and IP Security (IPsec) sessions.

When the system creates the fabric interface, the software assigns it an internally derived IP address to be used for packet transmission.



**CAUTION:** After fabric interfaces have been configured on a chassis cluster, removing the fabric configuration on either node will cause the redundancy group 0 (RGO) secondary node to move to a disabled state. (Resetting a device to the factory default configuration removes the fabric configuration and thereby causes the RGO secondary node to move to a disabled state.) After the fabric configuration is committed, do not reset either device to the factory default configuration.

### Supported Fabric Interface Types for SRX Series Firewalls (SRX300 Series, SRX1500, SRX1600, SRX2300, SRX4100/SRX4200, SRX4300, SRX4600, and SRX5000 line)

For SRX Series chassis clusters, the fabric link can be any pair of Ethernet interfaces spanning the cluster; the fabric link can be any pair of Gigabit Ethernet interface. Examples:

- For SRX300, SRX320, SRX340, and SRX345 devices, the fabric link can be any pair of Gigabit Ethernet interfaces. For SRX380 devices, the fabric link can be any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit Ethernet interface.
- For SRX1500 and SRX1600, the fabric link can be any pair of Ethernet interfaces spanning the cluster; the fabric link can be any pair of Gigabit Ethernet interface or any pair of 10-Gigabit Ethernet interface. For SRX1600, the fabric link can also be any pair of 25-Gigabit Ethernet interface.
- Supported fabric interface types for SRX4100 and SRX4200 devices are 10-Gigabit Ethernet (xe) (10-Gigabit Ethernet Interface SFP+ slots).
- Supported fabric interface types for SRX4300 are:
  - 10-Gigabit Ethernet (mge)
  - 10-Gigabit Ethernet (xe) (10-Gigabit Ethernet Interface SFP+ slots)
  - 40-Gigabit Ethernet (et) (25-Gigabit Ethernet Interface SFP28 and 100-Gigabit Ethernet Interface QSFP28 slots)
- Supported fabric interface types for SRX4600 devices are 40-Gigabit Ethernet (et) (40-Gigabit Ethernet Interface QSFP slots) and 10-Gigabit Ethernet (xe).
- Supported fabric interface types supported for SRX5000 line devices are:

- Fast Ethernet
- Gigabit Ethernet
- 10-Gigabit Ethernet
- 40-Gigabit Ethernet
- 100-Gigabit Ethernet

Starting in Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, 100-Gigabit Ethernet interface is supported on SRX5000 line devices.

Starting in Junos OS Release 19.3R1, the SRX5K-IOC4-10G and SRX5K-IOC4-MRAT are supported along with SRX5K-SPC3 on the SRX5000 line devices. SRX5K-IOC4-10G MPIC supports MACsec.

For details about port and interface usage for management, control, and fabric links, see ["Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming"](#) on page 19.

## Jumbo Frame Support

The fabric data link does not support fragmentation. To accommodate this state, jumbo frame support is enabled by default on the link with an maximum transmission unit (MTU) size of 9014 bytes (9000 bytes of payload + 14 bytes for the Ethernet header) on SRX Series Firewalls. To ensure the traffic that transits the data link does not exceed this size, we recommend that no other interfaces exceed the fabric data link's MTU size.

## Understanding Fabric Interfaces on SRX5000 Line Devices for IOC2 and IOC3

Starting with Junos OS Release 15.1X49-D10, the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) are introduced.

The SRX5K-MPC (IOC2) is a Modular Port Concentrator (MPC) that is supported on the SRX5400, SRX5600, and SRX5800. This interface card accepts Modular Interface Cards (MICs), which add Ethernet ports to your services gateway to provide the physical connections to various network media types. The MPCs and MICs support fabric links for chassis clusters. The SRX5K-MPC provides 10-Gigabit Ethernet (with 10x10GE MIC), 40-Gigabit Ethernet, 100-Gigabit Ethernet, and 20x1GE Ethernet ports as fabric ports. On SRX5400 devices, only SRX5K-MPCs (IOC2) are supported.

The SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) are Modular Port Concentrators (MPCs) that are supported on the SRX5400, SRX5600, and SRX5800. These interface cards accept Modular Interface Cards (MICs), which add Ethernet ports to your services gateway to provide the physical connections to various network media types. The MPCs and MICs support fabric links for chassis clusters.

The two types of IOC3 Modular Port Concentrators (MPCs), which have different built-in MICs, are the 24x10GE + 6x40GE MPC and the 2x100GE + 4x10GE MPC.

Due to power and thermal constraints, all four PICs on the 24x10GE + 6x40GE cannot be powered on. A maximum of two PICs can be powered on at the same time.

Use the `set chassis fpc <slot> pic <pic> power off` command to choose the PICs you want to power on.

On SRX5400, SRX5600, and SRX5800 devices in a chassis cluster, when the PICs containing fabric links on the SRX5K-MPC3-40G10G (IOC3) are powered off to turn on alternate PICs, always ensure that:

- The new fabric links are configured on the new PICs that are turned on. At least one fabric link must be present and online to ensure minimal RTO loss.
- The chassis cluster is in active-passive mode to ensure minimal RTO loss, once alternate links are brought online.
- If no alternate fabric links are configured on the PICs that are turned on, RTO synchronous communication between the two nodes stops and the chassis cluster session state will not back up, because the fabric link is missing. You can view the CLI output for this scenario indicating a bad chassis cluster state by using the `show chassis cluster interfaces` command.

## Understanding Session RTOs

The data plane software, which operates in active/active mode, manages flow processing and session state redundancy and processes transit traffic. All packets belonging to a particular session are processed on the same node to ensure that the same security treatment is applied to them. The system identifies the node on which a session is active and forwards its packets to that node for processing. (After a packet is processed, the Packet Forwarding Engine transmits the packet to the node on which its egress interface exists if that node is not the local one.)

To provide for session (or flow) redundancy, the data plane software synchronizes its state by sending special payload packets called runtime objects (RTOs) from one node to the other across the fabric data link. By transmitting information about a session between the nodes, RTOs ensure the consistency and stability of sessions if a failover were to occur, and thus they enable the system to continue to process traffic belonging to existing sessions. To ensure that session information is always synchronized between the two nodes, the data plane software gives RTOs transmission priority over transit traffic.

The data plane software creates RTOs for UDP and TCP sessions and tracks state changes. It also synchronizes traffic for IPv4 pass-through protocols such as Generic Routing Encapsulation (GRE) and IPsec.

RTOs for synchronizing a session include:

- Session creation RTOs on the first packet
- Session deletion and age-out RTOs

- Change-related RTOs, including:
  - TCP state changes
  - Timeout synchronization request and response messages
  - RTOs for creating and deleting temporary openings in the firewall (pinholes) and child session pinholes

## Understanding Data Forwarding

For Junos OS, flow processing occurs on a single node on which the session for that flow was established and is active. This approach ensures that the same security measures are applied to all packets belonging to a session.

A *chassis cluster* can receive traffic on an interface on one node and send it out to an interface on the other node. (In active/active mode, the ingress interface for traffic might exist on one node and its egress interface on the other.)

This traversal is required in the following situations:

- When packets are processed on one node, but need to be forwarded out an egress interface on the other node
- When packets arrive on an interface on one node, but must be processed on the other node

If the ingress and egress interfaces for a packet are on one node, but the packet must be processed on the other node because its session was established there, it must traverse the data link twice. This can be the case for some complex media sessions, such as voice-over-IP (VoIP) sessions.

## Understanding Fabric Data Link Failure and Recovery

Intrusion Detection and Prevention (IDP) services do not support failover. For this reason, IDP services are not applied for sessions that were present prior to the failover. IDP services are applied for new sessions created on the new primary node.

The fabric data link is vital to the chassis cluster. If the link is unavailable, traffic forwarding and RTO synchronization are affected, which can result in loss of traffic and unpredictable system behavior.

To eliminate this possibility, Junos OS uses fabric monitoring to check whether the fabric link, or the two fabric links in the case of a dual fabric link configuration, are alive by periodically transmitting probes over the fabric links. If Junos OS detects fabric faults, RG1+ status of the secondary node changes to ineligible. It determines that a fabric fault has occurred if a fabric probe is not received but the fabric interface is active. To recover from this state, both the fabric links need to come back to online state and should start exchanging probes. As soon as this happens, all the FPCs on the previously ineligible node will be reset. They then come to online state and rejoin the cluster.

If you make any changes to the configuration while the secondary node is disabled, execute the `commit` command to synchronize the configuration after you reboot the node. If you did not make configuration changes, the configuration file remains synchronized with that of the primary node.

Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, the fabric monitoring feature is enabled by default on SRX5800, SRX5600, and SRX5400 devices.

Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, recovery of the fabric link and synchronization take place automatically.

When both the primary and secondary nodes are healthy (that is, there are no failures) and the fabric link goes down, RG1+ redundancy group(s) on the secondary node becomes ineligible. When one of the nodes is unhealthy (that is, there is a failure), RG1+ redundancy group(s) on this node (either the primary or secondary node) becomes ineligible. When both nodes are unhealthy and the fabric link goes down, RG1+ redundancy group(s) on the secondary node becomes ineligible. When the fabric link comes up, the node on which RG1+ became ineligible performs a cold synchronization on all Services Processing Units and transitions to active standby.

- If RG0 is primary on an unhealthy node, then RG0 will fail over from an unhealthy to a healthy node. For example, if node 0 is primary for RG0+ and node 0 becomes unhealthy, then RG1+ on node 0 will transition to ineligible after 66 seconds of a fabric link failure and RG0+ fails over to node 1, which is the healthy node.
- Only RG1+ transitions to an ineligible state. RG0 continues to be in either a primary or secondary state.

Use the `show chassis cluster interfaces` CLI command to verify the status of the fabric link.

## SEE ALSO

[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming](#) | 19

[Understanding Chassis Cluster Dual Fabric Links](#) | 173

## Example: Configuring the Chassis Cluster Fabric Interfaces

### IN THIS SECTION

- [Requirements](#) | 66

- [Overview | 66](#)
- [Configuration | 66](#)
- [Verification | 68](#)

This example shows how to configure the chassis cluster fabric. The fabric is the back-to-back data connection between the nodes in a cluster. Traffic on one node that needs to be processed on the other node or to exit through an interface on the other node passes over the fabric. Session state information also passes over the fabric.

## Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See ["Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster "](#) on page 45.

## Overview

In most SRX Series Firewalls in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes.

You cannot configure filters, policies, or services on the fabric interface. Fragmentation is not supported on the fabric link. The maximum MTU size for fabric interfaces is 9014 bytes and the maximum MTU size for other interfaces is 8900 bytes. Jumbo frame support on the member links is enabled by default.

This example illustrates how to configure the fabric link.

Only the same type of interfaces can be configured as fabric children, and you must configure an equal number of child links for `fab0` and `fab1`.

If you are connecting each of the fabric links through a switch, you must enable the jumbo frame feature on the corresponding switch ports. If both of the fabric links are connected through the same switch, the RTO-and-probes pair must be in one virtual LAN (VLAN) and the data pair must be in another VLAN. Here too, the jumbo frame feature must be enabled on the corresponding switch ports.

## Configuration

### IN THIS SECTION

- [Procedure | 67](#)



## 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-7/0/1
```

### Step-by-Step Procedure

To configure the chassis cluster fabric:

- Specify the fabric interfaces.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
{primary:node0}[edit]
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/1
```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
...
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/1;
    }
  }
}
```

```

    }
}
fab1 {
    fabric-options {
        member-interfaces {
            ge-7/0/1;
        }
    }
}
}

```

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

## Verification

### IN THIS SECTION

- [Verifying the Chassis Cluster Fabric | 68](#)

### Verifying the Chassis Cluster Fabric

#### Purpose

Verify the chassis cluster fabric.

#### Action

From operational mode, enter the `show interfaces terse | match fab` command.

```

{primary:node0}
user@host> show interfaces terse | match fab
ge-0/0/1.0          up   up   aenet  --> fab0.0
ge-7/0/1.0          up   up   aenet  --> fab1.0
fab0                 up   up
fab0.0              up   up   inet   30.17.0.200/24
fab1                 up   up
fab1.0              up   up   inet   30.18.0.200/24

```

## Verifying Chassis Cluster Data Plane Interfaces

### IN THIS SECTION

- [Purpose | 69](#)
- [Action | 69](#)

### Purpose

Display chassis cluster data plane interface status.

### Action

From the CLI, enter the `show chassis cluster data-plane interfaces` command:

```
{primary:node1}
user@host> show chassis cluster data-plane interfaces
fab0:
  Name           Status
  ge-2/1/9       up
  ge-2/2/5       up
fab1:
  Name           Status
  ge-8/1/9       up
  ge-8/2/5       up
```

## Viewing Chassis Cluster Data Plane Statistics

### IN THIS SECTION

- [Purpose | 70](#)
- [Action | 70](#)

## Purpose

Display chassis cluster data plane statistics.

## Action

From the CLI, enter the show chassis cluster data-plane statistics command:

```
{primary:node1}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
  Service name                RTOs sent   RTOs received
  Translation context          0           0
  Incoming NAT                 0           0
  Resource manager             0           0
  Session create               0           0
  Session close                0           0
  Session change               0           0
  Gate create                  0           0
  Session ageout refresh requests 0           0
  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
  RTSP ALG                     0           0
```

## Clearing Chassis Cluster Data Plane Statistics

To clear displayed chassis cluster data plane statistics, enter the clear chassis cluster data-plane statistics command from the CLI:

```
{primary:node1}
user@host> clear chassis cluster data-plane statistics
```

Cleared data-plane statistics

## SEE ALSO

[Configuring Chassis Clustering on SRX Series Devices | 120](#)

## 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
19.3R1	Starting in Junos OS Release 19.3R1, the SRX5K-IOC4-10G and SRX5K-IOC4-MRAT are supported along with SRX5K-SPC3 on the SRX5000 line devices. SRX5K-IOC4-10G MPIC supports MACsec.
15.1X49-D10	Starting with Junos OS Release 15.1X49-D10, the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) are introduced.
12.1X47	Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, the fabric monitoring feature is enabled by default on SRX5800, SRX5600, and SRX5400 devices.
12.1X47	Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, recovery of the fabric link and synchronization take place automatically.
12.1X46	Starting in Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, 100-Gigabit Ethernet interface is supported on SRX5000 line devices.

## RELATED DOCUMENTATION

[SRX Series Chassis Cluster Configuration Overview | 14](#)

[Example: Configure Chassis Clustering on SRX Series Firewalls | 122](#)

# Chassis Cluster Control Plane Interfaces

## IN THIS SECTION

- [Chassis Cluster Control Plane and Control Links | 72](#)
- [Example: Configure Chassis Cluster Control Ports for Control Link | 76](#)
- [Clear Chassis Cluster Control Plane Statistics | 81](#)
- [Change from Chassis Cluster to Standalone Mode | 81](#)

You can use control plane interfaces to synchronize the kernel state between Routing Engines on SRX Series Firewalls in a chassis cluster. Control plane interfaces provide the link between the two nodes in the cluster.

Control planes use this link to:

- Communicate node discovery.
- Maintains session state for a cluster.
- Access the configuration file.
- Detect liveliness signals across the nodes.

## Chassis Cluster Control Plane and Control Links

### IN THIS SECTION

- [Chassis Cluster Control Links | 73](#)

The control plane software, which operates in active or backup mode, is an integral part of Junos OS that is active on the primary node of a cluster. It achieves redundancy by communicating state, configuration, and other information to the inactive Routing Engine on the secondary node. If the primary Routing Engine fails, the secondary Routing Engine is ready to assume control.

The control plane software:

- Runs on the Routing Engine.
- Oversees the entire *chassis cluster* system, including interfaces on both nodes.
- Manages system and data plane resources, including the Packet Forwarding Engine (PFE) on each node.
- Synchronizes the configuration over the control link.
- Establishes and maintains sessions, including authentication, authorization, and accounting (AAA) functions.
- Manages application-specific signaling protocols.
- Establishes and maintains management sessions, such as Telnet connections.
- Handles asymmetric routing.
- Manages routing state, Address Resolution Protocol (ARP) processing, and Dynamic Host Configuration Protocol (DHCP) processing.

Information from the control plane software follows two paths:

- On the primary node (where the Routing Engine is active), control information flows from the Routing Engine to the local Packet Forwarding Engine.
- Control information flows across the control link to the secondary node's Routing Engine and Packet Forwarding Engine.

The control plane software running on the primary Routing Engine maintains state for the entire cluster. Only those processes running on the same node as the control plane software can update state information. The primary Routing Engine synchronizes state for the secondary node and also processes all host traffic.

## Chassis Cluster Control Links

The control interfaces provide the control link between the two nodes in the cluster and are used for routing updates and for control plane signal traffic, such as heartbeat and threshold information that triggers node failover. The control link also synchronizes the configuration between the nodes. When you submit configuration statements to the cluster, the control link synchronizes the configuration automatically.

The control link relies on a proprietary protocol to transmit session state, configuration, and liveness state across the nodes.

Starting in Junos OS Release 19.3R1, the SRX5K-RE3-128G device is supported along with the SRX5K-SPC3 device on the SRX5000 line devices. The control interfaces ixlv0 and igb0 are used to configure the SRX5K-RE3-128G device. Control links control the communication between the control plane, data plane, and heartbeat messages.

### Single Control Link in a Chassis Cluster

For a single control link in a chassis cluster, you must use the same control port for the control link connection and for configuration on both nodes.

For example, if you configure port 0 as a control port on node 0, you must configure port 0 as a control port on node 1. You must connect the ports with a cable.

### Dual Control Link in a Chassis Cluster

You must connect dual control links in a chassis cluster directly. Cross connections—that is, connecting port 0 on one node to port 1 on the other node and vice versa—do not work.

For dual control links, you must make these connections:

- Connect control port 0 on node 0 to control port 0 on node 1.
- Connect control port 1 on node 0 to control port 1 on node 1.

### Encryption on Chassis Cluster Control Link

Chassis cluster control links support an optional encrypted security feature that you can configure and activate.

**NOTE:** Note that Juniper Networks security documentation uses *chassis cluster* when referring to *high availability (HA)* control links. You will still see the abbreviation *ha* used in place of *chassis cluster* in commands.

The control link access prevents hackers from logging in to the system without authentication through the control link, with Telnet access disabled. Using the internal IPsec key for internal communication between devices, the configuration information that passes through the chassis cluster link from the primary node to the secondary node is encrypted. Without the IPsec key, an attacker cannot gain privilege access or observe traffic.

To enable this feature, run the `set security ipsec internal security-association manual encryption ike-ha-link-encryption enable` configuration command.

You must reboot both the nodes to activate this configuration.

Encryption on chassis cluster control link using IPsec is supported on SRX4600 line devices, SRX5000 line devices, and vSRX Virtual Firewall platforms.



When the chassis cluster is running with the IPsec key configured already, then you can make any changes to the key without rebooting the device. In this case, you will have to change the key only on one node.

When IPsec key encryption is configured, for any configuration changes under internal security association (SA) hierarchy, you must reboot both the nodes. To verify the configured Internet Key Exchange (IKE) chassis cluster link encryption algorithm, view the output of `show security internal-security-association`.

[Table 13 on page 75](#) lists the supported control ports on the SRX Series Firewalls.

**Table 13: Supported Control Ports on SRX Series Firewalls**

SRX Series Firewalls	Description
SRX5400, SRX5600, and SRX5800	By default, all control ports are disabled. Each Services Processing Card (SPC) in a device has two control ports, and each device can have multiple SPCs plugged in to it. To set up the control link in a <i>chassis cluster</i> , you connect and configure the control ports that you use on each device (fpc<n> and fpc<n>), and then initialize the device in cluster mode.
SRX4600	Dedicated chassis cluster control ports and fabric ports are available.  No control link configuration is needed for SRX4600 devices; however, you need to configure fabric link explicitly for chassis cluster deployments. If you want to configure 1-Gigabit Ethernet interfaces for the control ports, you must explicitly set the speed using the operational CLI command statement <code>set chassis cluster control-port speed 1g</code> . See <a href="#">speed (Chassis Cluster)</a> .

Table 13: Supported Control Ports on SRX Series Firewalls (*Continued*)

SRX Series Firewalls	Description
SRX4100 and SRX4200	<p>Dedicated chassis cluster control ports are available. Control link configuration is not required. For more information about all SRX4100 ports and SRX4200 ports, including dedicated control links ports and fabric link ports, see <a href="#">Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming</a>.</p> <p>When devices are not in cluster mode, dedicated chassis cluster ports cannot be used as revenue ports or traffic ports.</p>
SRX2300 and SRX4300	Devices use the dual dedicated control port with MACsec support.
SRX1600	Devices use the dual dedicated control port with MACsec support.
SRX1500	Devices use the dedicated control port.
SRX300, SRX320, SRX340, SRX345, and SRX380.	Control link uses the ge-0/0/1 interface.

For details about port usage and interface usage for management links, control links, and fabric links, see [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming](#).

## Example: Configure Chassis Cluster Control Ports for Control Link

### IN THIS SECTION

- Requirements | 77
- Overview | 77
- Configuration | 77

This example shows how to configure chassis cluster control ports on these devices: SRX5400, SRX5600, and SRX5800. You need to configure the control ports that you will use on each device to set up the control link.

## Requirements

Before you begin:

- Understand chassis cluster control links. See "[Understanding Chassis Cluster Control Plane and Control Links](#)" on page 72.
- Physically connect the control ports on the devices. See "[Connecting SRX Series Devices to Create a Chassis Cluster](#)" on page 39.

## Overview

Control link traffic passes through the switches in the Services Processing Cards (SPCs) and reaches the other node. On SRX Series Firewalls, chassis cluster ports are located at the SPCs in the chassis cluster. By default, all control ports on SRX5400 devices, SRX5600 devices, and SRX5800 devices are disabled. To set up the control links, you connect the control ports, configure the control ports, and set up the chassis cluster.

This example configures control ports with the following Flexible PIC Concentrators (FPCs) and ports as the control link:

- FPC 4, port 0
- FPC 10, port 0

## Configuration

### IN THIS SECTION

- Procedure | 78
- Verify the Chassis Cluster Status | 79
- Verify Chassis Cluster Control Plane Statistics | 80

## Procedure

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` in configuration mode.

```
{primary:node0}[edit]
set chassis cluster control-ports fpc 4 port 0
set chassis cluster control-ports fpc 10 port 0
{primary:node1}[edit]
set chassis cluster control-ports fpc 4 port 0
set chassis cluster control-ports fpc 10 port 0
```

### Step-by-Step Procedure

To configure control ports as the control link for the chassis cluster:

Specify the control ports.

```
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 4 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 10 port 0
{primary:node1}[edit]
user@host# set chassis cluster control-ports fpc 4 port 0
{primary:node1}[edit]
user@host# set chassis cluster control-ports fpc 10 port 0
```

### Results

In configuration mode, confirm your configuration by entering the `show chassis cluster` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show chassis cluster
...
control-ports {
  fpc 4 port 0;
  fpc 10 port 0;
}
...
```

After you configure the device, enter `commit` in configuration mode.

### Verify the Chassis Cluster Status

#### Purpose

Verify the chassis cluster status.

#### Action

In operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no

Redundancy group: 1 , Failover count: 1
  node0              0         primary  no       no
  node1              0         secondary no       no
```

## Meaning

Use the **show chassis cluster status** command to confirm that the devices in the chassis cluster are communicating with each other. The preceding output shows that chassis cluster is functioning properly, as one device is the primary node and the other is the secondary node.

## Verify Chassis Cluster Control Plane Statistics

### Purpose

Display chassis cluster control plane statistics.

### Action

At the CLI, enter the `show chassis cluster control-plane statistics` command:

```
{primary:node1}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 124
    Heartbeat packets received: 125
Fabric link statistics:
  Child link 0
    Probes sent: 124
    Probes received: 125
```

```
{primary:node1}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258698
    Heartbeat packets received: 258693
  Control link 1:
    Heartbeat packets sent: 258698
    Heartbeat packets received: 258693
Fabric link statistics:
  Child link 0
```

```
Probes sent: 258690
Probes received: 258690
Child link 1
Probes sent: 258505
Probes received: 258505
```

## SEE ALSO

[Configuring Chassis Clustering on SRX Series Devices | 120](#)

## SEE ALSO

[Connecting SRX Series Firewalls to Create a Chassis Cluster | 39](#)

[Chassis Cluster Dual Control Links | 163](#)

## Clear Chassis Cluster Control Plane Statistics

To clear displayed chassis cluster control plane statistics, enter the `clear chassis cluster control-plane statistics` command at the CLI:

```
{primary:node1}
user@host> clear chassis cluster control-plane statistics

Cleared control-plane statistics
```

## Change from Chassis Cluster to Standalone Mode

Change from cluster mode to standalone mode when using single primary control link.

1. Disable the chassis cluster and reboot the devices to enter the standalone mode.

```
user@host> set chassis cluster disable reboot
```

2. Enable the chassis cluster and reboot the devices to enter the chassis cluster mode.

## 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
19.3R1	Starting in Junos OS Release 19.3R1, the SRX5K-RE3-128G is supported along with SRX5K-SPC3 on the SRX5000 line devices. The control interfaces ixlv0 and igb0 are used to configure SRX5K-RE3-128G. Control links control the communication between the control, and data plane and the heartbeat messages.

### RELATED DOCUMENTATION

[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming | 19](#)

[Example: Configure Chassis Clustering on SRX Series Firewalls | 122](#)

## Chassis Cluster Redundancy Groups

### IN THIS SECTION

- [Understanding Chassis Cluster Redundancy Groups | 83](#)
- [Example: Configuring Chassis Cluster Redundancy Groups | 87](#)

A redundancy group (RG) includes and manages a collection of objects on both nodes of a cluster. An RG is primary on one node and backup on the other node at any given time. For more information, see the following topics:



## Understanding Chassis Cluster Redundancy Groups

### IN THIS SECTION

- [Understanding Chassis Cluster Redundancy Group 0: Routing Engines | 84](#)
- [Understanding Chassis Cluster Redundancy Groups 1 Through 128 | 84](#)

Chassis clustering provides high availability of interfaces and services through redundancy groups and primacy within groups.

A redundancy group is an abstract construct that includes and manages a collection of objects. A redundancy group contains objects on both nodes. A redundancy group is primary on one node and backup on the other at any time. When a redundancy group is said to be primary on a node, its objects on that node are active.

Redundancy groups are independent units of failover. Each redundancy group fails over from one node to the other independent of other redundancy groups. When a redundancy group fails over, all its objects fail over together.

Three things determine the primacy of a redundancy group: the priority configured for the node, the node ID (in case of tied priorities), and the order in which the node comes up. If a lower priority node comes up first, then it will assume the primacy for a redundancy group (and will stay as primary if preempt is not enabled). If preempt is added to a redundancy group configuration, the device with the higher priority in the group can initiate a failover to become primary. By default, preemption is disabled. For more information on preemption, see *preempt (Chassis Cluster)*.

A *chassis cluster* can include many redundancy groups, some of which might be primary on one node and some of which might be primary on the other. Alternatively, all redundancy groups can be primary on a single node. One redundancy group's primacy does not affect another redundancy group's primacy. You can create up to 128 redundancy groups.

The maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure.

You can configure redundancy groups to suit your deployment. You configure a redundancy group to be primary on one node and backup on the other node. You specify the node on which the group is primary by setting priorities for both nodes within a redundancy group configuration. The node with the higher priority takes precedence, and the redundancy group's objects on it are active.

If a redundancy group is configured so that both nodes have the same priority, the node with the lowest node ID number always takes precedence, and the redundancy group is primary on it. In a two-node cluster, node 0 always takes precedence in a priority tie.

## Understanding Chassis Cluster Redundancy Group 0: Routing Engines

When you initialize a device in *chassis cluster* mode, the system creates a redundancy group referred to as redundancy group 0. Redundancy group 0 manages the primacy and failover between the Routing Engines on each node of the cluster. As is the case for all redundancy groups, redundancy group 0 can be primary on only one node at a time. The node on which redundancy group 0 is primary determines which Routing Engine is active in the cluster. A node is considered the primary node of the cluster if its Routing Engine is the active one.

The redundancy group 0 configuration specifies the priority for each node. The following priority scheme determines redundancy group 0 primacy. Note that the three-second value is the interval if the default `heartbeat-threshold` and `heartbeat-interval` values are used.

- The node that comes up first (at least three seconds prior to the other node) is the primary node.
- If both nodes come up at the same time (or within three seconds of each other):
  - The node with the higher configured priority is the primary node.
  - If there is a tie (either because the same value was configured or because default settings were used), the node with the lower node ID (node 0) is the primary node.

The previous priority scheme applies to redundancy groups  $x$  (redundancy groups numbered 1 through 128) as well, provided `preempt` is not configured. (See ["Example: Configuring Chassis Cluster Redundancy Groups" on page 87.](#))

You cannot enable preemption for redundancy group 0. If you want to change the primary node for redundancy group 0, you must do a manual failover.

Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine failover, in which case all processes running on the primary node are killed and then spawned on the new primary Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

## Understanding Chassis Cluster Redundancy Groups 1 Through 128

You can configure one or more redundancy groups numbered 1 through 128, referred to as redundancy group  $x$ . The maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure (see ["Maximum Number of Redundant Ethernet Interfaces Allowed \(SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800, SRX300, SRX320, SRX340, SRX345,](#)

[SRX 380, and SRX1500\)" on page 91](#)). Each redundancy group  $x$  acts as an independent unit of failover and is primary on only one node at a time.

Each redundancy group  $x$  contains one or more redundant Ethernet interfaces. A redundant Ethernet interface is a pseudo interface that contains at minimum a pair of physical Gigabit Ethernet interfaces or a pair of Fast Ethernet interfaces. If a redundancy group is active on node 0, then the child links of all the associated redundant Ethernet interfaces on node 0 are active. If the redundancy group fails over to node 1, then the child links of all redundant Ethernet interfaces on node 1 become active.

The following priority scheme determines redundancy group  $x$  primacy, provided preempt is not configured. If preempt is configured, the node with the higher priority is the primary node. Note that the three-second value is the interval if the default heartbeat-threshold and heartbeat-interval values are used.

- The node that comes up first (at least three seconds prior to the other node) is the primary node.
- If both nodes come up at the same time (or within three seconds of each other):
  - The node with the higher configured priority is the primary node.
  - If there is a tie (either because the same value was configured or because default settings were used), the node with the lower node ID (node 0) is the primary node.

On SRX Series chassis clusters, you can configure multiple redundancy groups to load-share traffic across the cluster. For example, you can configure some redundancy groups  $x$  to be primary on one node and some redundancy groups  $x$  to be primary on the other node. You can also configure a redundancy group  $x$  in a one-to-one relationship with a single redundant Ethernet interface to control which interface traffic flows through.

The traffic for a redundancy group is processed on the node where the redundancy group is active. Because more than one redundancy group can be configured, it is possible that the traffic from some redundancy groups is processed on one node while the traffic for other redundancy groups is processed on the other node (depending on where the redundancy group is active). Multiple redundancy groups make it possible for traffic to arrive over an ingress interface of one redundancy group and over an egress interface that belongs to another redundancy group. In this situation, the ingress and egress interfaces might not be active on the same node. When this happens, the traffic is forwarded over the fabric link to the appropriate node.

When you configure a redundancy group  $x$ , you must specify a priority for each node to determine the node on which the redundancy group  $x$  is primary. The node with the higher priority is selected as primary. The primacy of a redundancy group  $x$  can fail over from one node to the other. When a redundancy group  $x$  fails over to the other node, its redundant Ethernet interfaces on that node are active and their interfaces are passing traffic.

[Table 14 on page 86](#) gives an example of redundancy group  $x$  in an SRX Series *chassis cluster* and indicates the node on which the group is primary. It shows the redundant Ethernet interfaces and their interfaces configured for redundancy group  $x$ .

Some devices have both Gigabit Ethernet ports and Fast Ethernet ports.

**Table 14: Example of Redundancy Groups in a Chassis Cluster**

Group	Primary	Priority	Objects	Interface (Node 0)	Interface (Node 1)
Redundancy group 0	Node 0	Node 0: 254	Routing Engine on node 0	—	—
		Node 1: 2	Routing Engine on node 1	—	—
Redundancy group 1	Node 0	Node 0: 254	Redundant Ethernet interface 0	ge-1/0/0	ge-5/0/0
		Node 1: 2	Redundant Ethernet interface 1	ge-1/3/0	ge-5/3/0
Redundancy group 2	Node 1	Node 0: 2	Redundant Ethernet interface 2	ge-2/0/0	ge-6/0/0
		Node 1: 254	Redundant Ethernet interface 3	ge-2/3/0	ge-6/3/0
Redundancy group 3	Node 0	Node 0: 254	Redundant Ethernet interface 4	ge-3/0/0	ge-7/0/0
		Node 1: 2	Redundant Ethernet interface 5	ge-3/3/0	ge-7/3/0

As the example for a chassis cluster in [Table 14 on page 86](#) shows:

- The Routing Engine on node 0 is active because redundancy group 0 is primary on node 0. (The Routing Engine on node 1 is passive, serving as backup.)
- Redundancy group 1 is primary on node 0. Interfaces ge-1/0/0 and ge-1/3/0 belonging to redundant Ethernet interface 0 and redundant Ethernet interface 1 are active and handling traffic.
- Redundancy group 2 is primary on node 1. Interfaces ge-6/0/0 and ge-6/3/0 belonging to redundant Ethernet interface 2 and redundant Ethernet interface 3 are active and handling traffic.

- Redundancy group 3 is primary on node 0. Interfaces ge-3/0/0 and ge-3/3/0 belonging to redundant Ethernet interface 4 and redundant Ethernet interface 5 are active and handling traffic.

## Example: Configuring Chassis Cluster Redundancy Groups

### IN THIS SECTION

- [Requirements | 87](#)
- [Overview | 87](#)
- [Configuration | 88](#)
- [Verification | 90](#)

This example shows how to configure a chassis cluster redundancy group.

### Requirements

Before you begin:

1. Set the chassis cluster node ID and cluster ID. See ["Example: Setting the Chassis Cluster Node ID and Cluster ID" on page 45](#).
2. Configure the chassis cluster management interface. See ["Example: Configuring the Chassis Cluster Management Interface" on page 51](#).
3. Configure the chassis cluster fabric. See ["Example: Configuring the Chassis Cluster Fabric Interfaces" on page 65](#).

### Overview

A chassis cluster redundancy group is an abstract entity that includes and manages a collection of objects. Each redundancy group acts as an independent unit of failover and is primary on only one node at a time.

In this example, you create two chassis cluster redundancy groups, 0 and 1:

- 0—Node 0 is assigned a priority of 100, and node 1 is assigned a priority of 1.
- 1—Node 0 is assigned a priority of 100, and node 1 is assigned a priority of 1.

The preempt option is enabled, and the number of gratuitous ARP requests that an interface can send to notify other network devices of its presence after the redundancy group it belongs to has failed over is 4.

## Configuration

### IN THIS SECTION

- [Procedure | 88](#)

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
[edit]
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 gratuitous-arp-count 4
```

#### Step-by-Step Procedure

To configure a chassis cluster redundancy group:

1. Specify a redundancy group's priority for primacy on each node of the cluster. The higher number takes precedence.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
```

```
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
```

2. Configure the node with the higher priority to preempt the device with the lower priority and become primary for the redundancy group.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 preempt
```

You cannot enable preemption for redundancy group 0. If you want to change the primary node for redundancy group 0, you must do a manual failover.

3. Specify the number of gratuitous ARP requests that an interface can send to notify other network devices of its presence after the redundancy group it belongs to has failed over.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 gratuitous-arp-count 4
```

## Results

From configuration mode, confirm your configuration by entering the `show chassis cluster status redundancy-group` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
{primary:node0}[edit]
user@host# show chassis cluster
chassis {
  cluster {
    redundancy-group 0 {
      node 0 priority 100;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 100;
      node 1 priority 1;
      preempt;
      gratuitous-arp-count 4;
    }
  }
}
```

```
}
}
```

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

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Redundancy Group Status | 90](#)

## Verifying Chassis Cluster Redundancy Group Status

### Purpose

Verify the status of a chassis cluster redundancy group.

### Action

From operational mode, enter the `show chassis cluster status redundancy-group` command.

```
{primary:node0}
user@host>show chassis cluster status redundancy-group 1

Cluster ID: 1
Node          Priority      Status    Preempt  Manual failover

Redundancy group: 1 , Failover count: 1
  node0          100         primary   no       no
  node1           1          secondary yes       no
```



# Chassis Cluster Redundant Ethernet Interfaces

## IN THIS SECTION

- [Understanding Chassis Cluster Redundant Ethernet Interfaces | 91](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces | 95](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on SRX4600 | 103](#)
- [Example: Configuring IP-over-IP Tunnel on SRX Series Firewalls | 110](#)

A redundant Ethernet (reth) interface is a pseudo-interface that includes minimum one physical interface from each node of a cluster. A reth interface of the active node is responsible for passing the traffic in a chassis cluster setup. For more information, see the following topics:

## Understanding Chassis Cluster Redundant Ethernet Interfaces

### IN THIS SECTION

- [IP-over-IP Tunneling | 94](#)

For SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, SRX1600, SRX2300, SRX4100, SRX4200, and SRX4300 devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a *chassis cluster* deployment is 1024.

For SRX5800, SRX5600, SRX5400, and SRX4600 devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a *chassis cluster* deployment is 4096.

Starting with Junos OS Release 12.1X45-D10 and later, sampling features such as flow monitoring, packet capture, and port mirroring are supported on reth interfaces.

A redundant Ethernet interface must contain, at minimum, a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces that are referred to as child interfaces of the redundant Ethernet interface (the redundant parent). If two or more child interfaces from each node are assigned to the redundant Ethernet interface, a redundant Ethernet interface link aggregation group can be formed. A single

redundant Ethernet interface might include a Fast Ethernet interface from node 0 and a Fast Ethernet interface from node 1 or a Gigabit Ethernet interface from node 0 and a Gigabit Ethernet interface from node 1.

On SRX5600, and SRX5800 devices, interfaces such as 10-Gigabit Ethernet (xe), 40-Gigabit Ethernet, and 100-Gigabit Ethernet can be redundant Ethernet (reth) interfaces.

SRX4100 and SRX4200 devices support 10-Gigabit Ethernet (xe) interfaces as redundant Ethernet (reth) interfaces.

A redundant Ethernet interface is referred to as a reth in configuration commands.

A redundant Ethernet interface's child interface is associated with the redundant Ethernet interface as part of the child interface configuration. The redundant Ethernet interface child interface inherits most of its configuration from its parent.

The maximum number of redundant Ethernet interfaces that you can configure varies, depending on the device type you are using, as shown in [Table 15 on page 92](#). The number of redundant Ethernet interfaces configured determines the number of redundancy groups that can be configured in the SRX Series Firewalls.

**Table 15: Maximum Number of Redundant Ethernet Interfaces Allowed (SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, SRX1600, SRX2300, SRX4100, SRX4200, SRX4300, SRX4600, SRX5400, SRX5600, SRX5800)**

Device	Maximum Number of reth Interfaces
SRX4600	128
SRX4100, SRX4200, and SRX4300	128
SRX5400, SRX5600, SRX5800	128
SRX300, SRX320, SRX340, SRX345,SRX380	128
SRX1500	128
SRX1600	128
SRX2300	128

You can enable promiscuous mode on redundant Ethernet interfaces. When promiscuous mode is enabled on a Layer 3 Ethernet interface, all packets received on the interface are sent to the central point or Services Processing Unit (SPU), regardless of the destination MAC address of the packet. If you enable promiscuous mode on a redundant Ethernet interface, promiscuous mode is then enabled on any child physical interfaces.

To enable promiscuous mode on a redundant Ethernet interface, use the `promiscuous-mode` statement at the `[edit interfaces]` hierarchy.

A redundant Ethernet interface inherits its failover properties from the redundancy group  $x$  that it belongs to. A redundant Ethernet interface remains active as long as its primary child interface is available or active. For example, if `reth0` is associated with redundancy group 1 and redundancy group 1 is active on node 0, then `reth0` is up as long as the node 0 child of `reth0` is up.

Point-to-Point Protocol over Ethernet (PPPoE) over redundant Ethernet (`reth`) interface is supported on SRX300, SRX320, SRX340, SRX345, SRX380, SRX1500, and SRX1600 devices in chassis cluster mode. This feature allows an existing PPPoE session to continue without starting a new PPPoE session in the event of a failover.

On SRX300, SRX320, SRX340, SRX345, and SRX380 devices, the number of child interfaces is restricted to 16 on the `reth` interface (eight per node).

When using SRX Series Firewalls in chassis cluster mode, it is not recommended to configure any local interfaces (or combination of local interfaces) along with redundant Ethernet interfaces.

For example:

The following configuration of chassis cluster with redundant Ethernet interfaces in which interfaces are configured as local interfaces:

```
ge-2/0/2 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
```

The following configuration of chassis cluster redundant Ethernet interfaces, in which interfaces are configured as part of redundant Ethernet interfaces, is supported:

```
interfaces {
  ge-2/0/2 {
    gigether-options {
```

```

        redundant-parent reth2;
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.1.1.1/24;
        }
    }
}
}

```

You can enable promiscuous mode on redundant Ethernet interfaces. When promiscuous mode is enabled on a Layer 3 Ethernet interface, all packets received on the interface are sent to the central point or Services Processing Unit (SPU), regardless of the destination MAC address of the packet. If you enable promiscuous mode on a redundant Ethernet interface, promiscuous mode is then enabled on any child physical interfaces.

To enable promiscuous mode on a redundant Ethernet interface, use the `promiscuous-mode` statement at the `[edit interfaces]` hierarchy.

## IP-over-IP Tunneling

IP-over-IP tunneling is supported over the reth interface in SRX chassis cluster configuration. Tunneling allows the encapsulation of one IP packet over another IP packet.

The tunnel configuration is created on both the primary and secondary devices. The traffic passing through the IP-IP tunnel is synced from primary device to secondary device. The tunnel configuration on secondary device is considered as backup and active in the event of failure of the primary device. The traffic resumes on the secondary device in the event of the failure of the primary.

On SRX Series Firewalls, Generic Routing Encapsulation (GRE) and IP-IP tunnels use internal interfaces, `gr-0/0/0` and `ip-0/0/0`, respectively. The Junos OS creates these interfaces at system bootup; they are not associated with physical interfaces.

## SEE ALSO

| [Understanding Link Aggregation Groups in a Chassis Cluster](#) | 270

## Example: Configuring Chassis Cluster Redundant Ethernet Interfaces

### IN THIS SECTION

- Requirements | 95
- Overview | 95
- Configuration | 96
- Verification | 101

This example shows how to configure chassis cluster redundant Ethernet interfaces. A redundant Ethernet interface is a pseudointerface that contains two or more physical interfaces, with at least one from each node of the cluster.

### Requirements

Before you begin:

- Understand how to set the chassis cluster node ID and cluster ID. See ["Example: Setting the Chassis Cluster Node ID and Cluster ID"](#) on page 45.
- Set the number of redundant Ethernet interfaces.
- Understand how to set the chassis cluster fabric. See ["Example: Configuring the Chassis Cluster Fabric Interfaces"](#) on page 65.
- Understand how to set the chassis cluster node redundancy groups. See ["Example: Configuring Chassis Cluster Redundancy Groups"](#) on page 87.

### Overview

After physical interfaces have been assigned to the redundant Ethernet interface, you set the configuration that pertains to them at the level of the redundant Ethernet interface, and each of the child interfaces inherits the configuration.

If multiple child interfaces are present, then the speed of all the child interfaces must be the same.

A redundant Ethernet interface is referred to as a reth in configuration commands.

You can enable promiscuous mode on redundant Ethernet interfaces. When promiscuous mode is enabled on a Layer 3 Ethernet interface, all packets received on the interface are sent to the central point or Services Processing Unit regardless of the destination MAC address of the packet. If you enable

promiscuous mode on a redundant Ethernet interface, promiscuous mode is then enabled on any child physical interfaces.

To enable promiscuous mode on a redundant Ethernet interface, use the `promiscuous-mode` statement at the `[edit interfaces]` hierarchy.

## Configuration

### IN THIS SECTION

- [Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 Addresses | 96](#)
- [Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv6 Addresses | 98](#)
- [Results | 99](#)

### Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 Addresses

#### 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.

```
{primary:node0}[edit]
set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces ge-7/0/0 gigether-options redundant-parent reth1
set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet mtu 1500
set interfaces reth1 unit 0 family inet address 10.1.1.3/24
set security zones security-zone Trust interfaces reth1.0
```

#### Step-by-Step Procedure

To configure redundant Ethernet interfaces for IPv4:

1. Bind redundant child physical interfaces to reth1.

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-7/0/0 gigether-options redundant-parent reth1
```

2. Bind redundant child physical interfaces to reth2.

```
{primary:node0}[edit]
user@host# set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
user@host# set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
```

3. Add reth1 to redundancy group 1.

```
{primary:node0}[edit]
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
```

4. Set the MTU size.

```
{primary:node0}[edit]
user@host# set interfaces reth1 unit 0 family inet mtu 1500
```

The maximum transmission unit (MTU) set on the reth interface can be different from the MTU on the child interface.

5. Assign an IP address to reth1.

```
{primary:node0}[edit]
user@host# set interfaces reth1 unit 0 family inet address 10.1.1.3/24
```

6. Associate reth1.0 to the trust security zone.

```
{primary:node0}[edit]
user@host# set security zones security-zone Trust interfaces reth1.0
```

## Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv6 Addresses

### 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.

```
{primary:node0}[edit]
set interfaces ge-0/0/0 gigether-options redundant-parent reth1
set interfaces ge-7/0/0 gigether-options redundant-parent reth1
set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth2 unit 0 family inet6 mtu 1500
set interfaces reth2 unit 0 family inet6 address 2010:2010:201::2/64
set security zones security-zone Trust interfaces reth2.0
```

### Step-by-Step Procedure

To configure redundant Ethernet interfaces for IPv6:

1. Bind redundant child physical interfaces to reth1.

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-7/0/0 gigether-options redundant-parent reth1
```

2. Bind redundant child physical interfaces to reth2.

```
{primary:node0}[edit]
user@host# set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
user@host# set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
```

3. Add reth2 to redundancy group 1.

```
{primary:node0}[edit]
user@host# set interfaces reth2 redundant-ether-options redundancy-group 1
```



4. Set the MTU size.

```
{primary:node0}[edit]
user@host# set interfaces reth2 unit 0 family inet6 mtu 1500
```

5. Assign an IP address to reth2.

```
{primary:node0}[edit]
user@host# set interfaces reth2 unit 0 family inet6 address 2010:2010:201::2/64
```

6. Associate reth2.0 to the trust security zone.

```
{primary:node0}[edit]
user@host# set security zones security-zone Trust interfaces reth2.0
```

## Step-by-Step Procedure

To set the number of redundant Ethernet interfaces for a chassis cluster:

1. Specify the number of redundant Ethernet interfaces:

```
{primary:node0}[edit]

user@host# set chassis cluster reth-count 2
```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
interfaces {
  ...
```

```
fe-1/0/0 {
    fastether-options {
        redundant-parent reth2;
    }
}
fe-8/0/0 {
    fastether-options {
        redundant-parent reth2;
    }
}
ge-0/0/0 {
    ggether-options {
        redundant-parent reth1;
    }
}
ge-7/0/0 {
    ggether-options {
        redundant-parent reth1;
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            mtu 1500;
            address 10.1.1.3/24;
        }
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet6 {
            mtu 1500;
            address 2010:2010:201::2/64;
        }
    }
}
```

```
...
}
```

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

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Redundant Ethernet Interfaces | 101](#)
- [Verifying Chassis Cluster Control Links | 101](#)

Confirm that the configuration is working properly.

### Verifying Chassis Cluster Redundant Ethernet Interfaces

#### Purpose

Verify the configuration of the chassis cluster redundant Ethernet interfaces.

#### Action

From operational mode, enter the `show interfaces terse | match reth1` command:

```
{primary:node0}
user@host> show interfaces terse | match reth1

ge-0/0/0.0          up   up   aenet  --> reth1.0
ge-7/0/0.0          up   up   aenet  --> reth1.0
reth1               up   up
reth1.0             up   up   inet   10.1.1.3/24
```

### Verifying Chassis Cluster Control Links

#### Purpose

Verify information about the control interface in a chassis cluster configuration.

## Action

From operational mode, enter the `show chassis cluster interfaces` command:

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      em0       Up                Disabled     Disabled
  1      em1       Up                Disabled     Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
                    (Physical/Monitored)
  fab0    xe-3/0/6         Up / Up         Enabled
  fab0
  fab1    xe-9/0/6         Up / Up         Enabled
  fab1

Redundant-ethernet Information:
  Name    Status  Redundancy-group
  reth0   Up      1
  reth1   Up      1
```

## SEE ALSO

[Example: Configuring Chassis Cluster Redundant Ethernet Interfaces](#) | 95

## Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on SRX4600

### IN THIS SECTION

- [Requirements | 103](#)
- [Overview | 103](#)
- [Configuration | 103](#)
- [Verification | 108](#)

This example shows how to configure child links or physical links on SRX4600 device in chassis cluster mode.

### Requirements

Before you begin:

- Understand how to set the chassis cluster node ID and cluster ID. See ["Example: Setting the Chassis Cluster Node ID and Cluster ID" on page 45](#).
- Understand how to set the chassis cluster node redundancy groups. See ["Example: Configuring Chassis Cluster Redundancy Groups" on page 87](#).

### Overview

You can configure up to eight number of child links for a reth bundle on SRX4600 devices per chassis.

### Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 104](#)
- [Configuring redundant Ethernet interfaces | 104](#)
- [Results | 106](#)

## 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.

```
{primary:node0}[edit]
set interfaces xe-1/0/0:0 ggether-options redundant-parent reth0
set interfaces xe-1/0/0:1 ggether-options redundant-parent reth0
set interfaces xe-1/0/0:2 ggether-options redundant-parent reth0
set interfaces xe-1/0/0:3 ggether-options redundant-parent reth0
set interfaces xe-1/0/1:0 ggether-options redundant-parent reth0
set interfaces xe-1/0/1:1 ggether-options redundant-parent reth0
set interfaces xe-1/0/1:2 ggether-options redundant-parent reth0
set interfaces xe-1/0/1:3 ggether-options redundant-parent reth0
set interfaces xe-1/1/0 ggether-options redundant-parent reth1
set interfaces xe-1/1/1 ggether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.0.2.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options lacp active
set interfaces reth1 unit 0 family inet address 198.51.100.1/24
set security zones security-zone Trust-zone interfaces reth0.0
set security zones security-zone Untrust-zone interfaces reth1.0
set chassis cluster reth-count 10
```

## Configuring redundant Ethernet interfaces

### Step-by-Step Procedure

To configure redundant Ethernet interfaces:

1. Bind eight redundant child physical interfaces to reth0.

```
{primary:node0}[edit]
user@host# set interfaces xe-1/0/0:0 ggether-options redundant-parent reth0
user@host# set interfaces xe-1/0/0:1 ggether-options redundant-parent reth0
user@host# set interfaces xe-1/0/0:2 ggether-options redundant-parent reth0
user@host# set interfaces xe-1/0/0:3 ggether-options redundant-parent reth0
user@host# set interfaces xe-1/0/1:0 ggether-options redundant-parent reth0
user@host# set interfaces xe-1/0/1:1 ggether-options redundant-parent reth0
```

```
user@host# set interfaces xe-1/0/1:2 gigether-options redundant-parent reth0
user@host# set interfaces xe-1/0/1:3 gigether-options redundant-parent reth0
```

2. Bind redundant child physical interfaces to reth1.

```
{primary:node0}[edit]
user@host# set interfaces xe-1/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-1/1/1 gigether-options redundant-parent reth1
```

3. Specify the number of redundant Ethernet interfaces:

```
{primary:node0}[edit]

user@host# set chassis cluster reth-count 10
```

4. Add reth0 to redundancy group 1.

```
{primary:node0}[edit]
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
```

5. Assign an IP address to reth0.

```
{primary:node0}[edit]
user@host# set interfaces reth0 unit 0 family inet address 192.0.2.1/24
```

6. Add reth1 to redundancy group1.

```
{primary:node0}[edit]
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 redundant-ether-options lacp active
```

7. Assign an IP address to reth1.

```
{primary:node0}[edit]
user@host# set interfaces reth1 unit 0 family inet address 198.51.100.1/24
```

## 8. Associate reth0.0 to the trust security zone.

```
{primary:node0}[edit]
user@host# set security zones security-zone Trust-zone interfaces reth0.0
```

## 9. Associate reth1.0 to untrust security zone.

```
{primary:node0}[edit]
user@host# set security zones security-zone Untrust-zone interfaces reth1.0
```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
xe-1/0/0:0 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-1/0/0:1 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-1/0/0:2 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-1/0/0:3 {
  gigether-options {
    redundant-parent reth0;
  }
}
```



```
}
xe-1/0/1:0 {
    gigether-options {
        redundant-parent reth0;
    }
}
xe-1/0/1:1 {
    gigether-options {
        redundant-parent reth0;
    }
}
xe-1/0/1:2 {
    gigether-options {
        redundant-parent reth0;
    }
}
xe-1/0/1:3 {
    gigether-options {
        redundant-parent reth0;
    }
}
xe-1/1/0 {
    gigether-options {
        redundant-parent reth1;
    }
}
xe-1/1/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
```

```

    redundancy-group 1;
    lacp {
        active;
    }
}
unit 0 {
    family inet {
        address 198.51.100.1/24;
    }
}
}

```

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

## Verification

### IN THIS SECTION

- [Verify Chassis Cluster Redundant Ethernet Interfaces | 108](#)
- [Verifying Chassis Cluster Control Links | 109](#)

Confirm that the configuration is working properly.

### Verify Chassis Cluster Redundant Ethernet Interfaces

#### Purpose

Verify the configuration of the chassis cluster redundant Ethernet interfaces on SRX4600 device.

#### Action

From operational mode, enter the `show interfaces terse | match reth0` command:

```

{primary:node0}
user@host> show interfaces terse | match reth0

xe-1/0/0:0.0      up    down aenet  --> reth0.0
xe-1/0/0:1.0      up    down aenet  --> reth0.0

```

```

xe-1/0/0:2.0      up    down aenet  --> reth0.0
xe-1/0/0:3.0      up    down aenet  --> reth0.0
xe-1/0/1:0.0      up    down aenet  --> reth0.0
xe-1/0/1:1.0      up    down aenet  --> reth0.0
xe-1/0/1:2.0      up    down aenet  --> reth0.0
xe-1/0/1:3.0      up    down aenet  --> reth0.0
reth0             up    down
reth0.0           up    down inet   192.0.2.1/24

```

## Meaning

You can view the maximum number of configured child link interfaces of a reth bundle from four to eight in one chassis.

## Verifying Chassis Cluster Control Links

### Purpose

Verify information about the control interface in a chassis cluster configuration.

### Action

From operational mode, enter the `show chassis cluster interfaces` command:

```

{primary:node0}
user@host> show chassis cluster interfaces

Control link status: Down

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  ----  -
  0      em0        Down              Disabled     Disabled
  1      em1        Down              Disabled     Disabled

Fabric link status: Down

Fabric interfaces:
  Name    Child-interface  Status
                    (Physical/Monitored)
  ----    -
  fab0    xe-0/0/2         Up / Down
  fab0

```

## Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Down	1
reth1	Up	1
reth2	Down	Not configured
reth3	Down	Not configured
reth4	Down	Not configured
reth5	Down	Not configured
reth6	Down	Not configured
reth7	Down	Not configured
reth8	Down	Not configured
reth9	Down	Not configured

## Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

## Example: Configuring IP-over-IP Tunnel on SRX Series Firewalls

### IN THIS SECTION

- [Requirements | 110](#)
- [Overview | 111](#)
- [Topology | 111](#)
- [Configuration | 111](#)
- [Verification | 116](#)

This example shows how to create IP-IP tunnel with a forwarding next-hop to pass IPv4 traffic through the tunnel and synchronize the configuration from primary device to secondary device.

### Requirements

Before you begin:

- Understand how to set the chassis cluster node ID and cluster ID. See ["Example: Setting the Chassis Cluster Node ID and Cluster ID" on page 45.](#)

This example uses the following hardware and software components:

- Junos OS Release 23.1R1 or later version.
- SRX345 Device

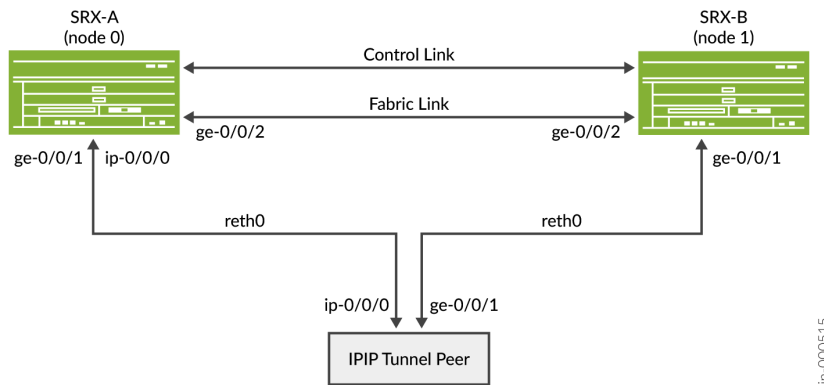
### Overview

Packets are routed to an internal interface where they are encapsulated with an IP packet and then forwarded to the encapsulating packet's destination address. The IP-IP interface is an internal interface only and is not associated with a physical interface. You must configure the interface for it to perform IP tunneling.

### Topology

[Figure 31 on page 111](#) illustrates IP-over-IP scenario with SRX Series Firewalls operating in chassis cluster mode.

**Figure 31: Configuring SRX Series Firewalls using IP-IP Tunnel**



### Configuration

#### IN THIS SECTION

- [Configuring IP-IP tunnel with Chassis Cluster Redundant Ethernet Interfaces for IPv4 Addresses | 112](#)
- [Results | 114](#)

## Configuring IP-IP tunnel with Chassis Cluster Redundant Ethernet Interfaces for IPv4 Addresses

### 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.

```
{primary:node0}[edit]
set chassis cluster reth-count 2
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1

set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1

set interfaces reth0 unit 0 family inet address 22.0.0.254/24

set interfaces reth1 unit 0 family inet address 1.0.0.254/24

set interfaces ip-0/0/0 unit 0 tunnel source 22.0.0.254
set interfaces ip-0/0/0 unit 0 tunnel destination 22.0.0.200
set interfaces ip-0/0/0 unit 0 family inet address 33.0.0.254/24

set interfaces ge-0/0/1 gigether-options redundant-parent reth0
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-7/0/1 gigether-options redundant-parent reth0
set interfaces ge-7/0/2 gigether-options redundant-parent reth1

set interfaces fab0 fabric-options member-interfaces ge-0/0/0
set interfaces fab1 fabric-options member-interfaces ge-7/0/0
```

```
{peer}
set interfaces ip-0/0/0 unit 0 tunnel source 22.0.0.200
set interfaces ip-0/0/0 unit 0 tunnel destination 22.0.0.254
set interfaces ip-0/0/0 unit 0 family inet address 33.0.0.200/24
set interfaces ge-0/0/1 unit 0 family inet address 22.0.0.200/24
```

```
set interfaces ge-0/0/2 unit 0 family inet address 2.0.0.200/24
set routing-options static route 1.0.0.0/24 next-hop ip-0/0/0.0
```

## Step-by-Step Procedure

To configure redundant Ethernet interfaces for IPv4:

1. Set up redundancy group 0 for the Routing Engine failover properties, and set up redundancy group 1 (all interfaces are in one redundancy group in this example) to define the failover properties for the redundant Ethernet interfaces.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
```

2. Set up the redundant Ethernet (reth) interfaces.

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
user@host# set interfaces ge-0/0/1 gigether-options redundant-parent reth0
user@host# set interfaces ge-0/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-7/0/1 gigether-options redundant-parent reth0
user@host# set interfaces ge-7/0/2 gigether-options redundant-parent reth1
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 22.0.0.254/24
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 1.0.0.254/24
```

3. Configure tunnel over redundant ethernet interface on both the nodes.

```
{primary:node0}[edit]
user@host# set interfaces ip-0/0/0 unit 0 tunnel source 22.0.0.254
user@host# set interfaces ip-0/0/0 unit 0 tunnel destination 22.0.0.200
user@host# set interfaces ip-0/0/0 unit 0 family inet address 33.0.0.254/24
```

#### 4. Configure tunnel session on the peer.

```
{peer}
user@host# set interfaces ip-0/0/0 unit 0 tunnel source 22.0.0.200
user@host# set interfaces ip-0/0/0 unit 0 tunnel destination 22.0.0.254
user@host# set interfaces ip-0/0/0 unit 0 family inet address 33.0.0.200/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 22.0.0.200/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 2.0.0.200/24
user@host# set routing-options static route 1.0.0.0/24 next-hop ip-0/0/0.0
```

#### 5. Configure routing-options on both the nodes:

```
{primary:node0}
set routing-options static route 2.0.0.0/24 next-hop ip-0/0/0.0
```

#### 6. Configure fabric interfaces on both the nodes.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/0
```

### Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
ip-0/0/0 {
  unit 0 {
    tunnel {
      source 22.0.0.254;
      destination 22.0.0.200;
    }
    family inet {
```



```
        address 33.0.0.254/24;
    }
}
ge-0/0/1 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-0/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-7/0/1 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-7/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
fab0 {
    fabric-options {
        member-interfaces {
            ge-0/0/0;
        }
    }
}
fab1 {
    fabric-options {
        member-interfaces {
            ge-7/0/0;
        }
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
}
unit 0 {
```

```
        family inet {
            address 22.0.0.254/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 1.0.0.254/24;
        }
    }
}
```

## Verification

### IN THIS SECTION

- [Purpose | 116](#)
- [Action | 116](#)
- [Meaning | 120](#)

### Purpose

Display the information about chassis cluster interfaces and status.

### Action

From operational mode, enter the show chassis cluster interfaces, show chassis cluster status, and show security flow session command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up
```

## Control interfaces:

Index	Interface	Monitored-Status	Internal-SA	Security
0	fxp1	Up	Disabled	Disabled

Fabric link status: Down

## Fabric interfaces:

Name	Child-interface	Status (Physical/Monitored)	Security
fab0	ge-0/0/0	Up / Up	Disabled
fab0			
fab1	ge-0/0/7	Up / Up	Disabled
fab1			

## Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1

## Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

{primary:node0}

user@host&gt; show chassis cluster status

## Monitor Failure codes:

CS	Cold Sync monitoring	FL	Fabric Connection monitoring
GR	GRES monitoring	HW	Hardware monitoring
IF	Interface monitoring	IP	IP monitoring
LB	Loopback monitoring	MB	Mbuf monitoring
NH	Nexthop monitoring	NP	NPC monitoring
SP	SPU monitoring	SM	Schedule monitoring
CF	Config Sync monitoring	RE	Relinquish monitoring
IS	IRQ storm		

Cluster ID: 1

Node	Priority	Status	Preempt	Manual	Monitor-failures
------	----------	--------	---------	--------	------------------

Redundancy group: 0 , Failover count: 1

node0	200	primary	no	no	None
node1	100	secondary	no	no	None

Redundancy group: 1 , Failover count: 1

node0	200	primary	no	no	None
node1	100	secondary	no	no	None

```
{primary:node0}
```

```
user@host> show security flow session
```

```
node0:
```

```
-----
Session ID: 6323, Policy name: N/A, HA State: Active, Timeout: N/A, Session State: Valid
  In: 2012::2:2/1 --> 2012::2:1/1;ipip, Conn Tag: 0x0, If: reth1.0, Pkts: 0, Bytes: 0,
```

```
Session ID: 6324, Policy name: N/A, HA State: Active, Timeout: N/A, Session State: Valid
  In: 2012::2:2/1 --> 2012::2:1/1;ipv6, Conn Tag: 0x0, If: reth1.0, Pkts: 0, Bytes: 0,
```

```
Session ID: 6361, Policy name: self-traffic-policy/1, HA State: Active, Timeout: 56, Session
State: Valid
```

```
  In: fe80::2:2/1 --> ff02::5/1;ospf, Conn Tag: 0x0, If: ip-0/0/0.1, Pkts: 153842, Bytes:
12371296,
```

```
  Out: ff02::5/1 --> fe80::2:2/1;ospf, Conn Tag: 0x0, If: .local..0, Pkts: 0, Bytes: 0,
```

```
Session ID: 6362, Policy name: self-traffic-policy/1, HA State: Active, Timeout: 52, Session
State: Valid
```

```
  In: 100.0.2.2/1 --> 224.0.0.5/1;ospf, Conn Tag: 0x0, If: ip-0/0/0.1, Pkts: 152030, Bytes:
12178352,
```

```
  Out: 224.0.0.5/1 --> 100.0.2.2/1;ospf, Conn Tag: 0x0, If: .local..0, Pkts: 0, Bytes: 0,
```

```
Session ID: 6363, Policy name: self-traffic-policy/1, HA State: Active, Timeout: 60, Session
State: Valid
```

```
  In: 100.0.2.2/49152 --> 100.0.2.1/3784;udp, Conn Tag: 0x0, If: ip-0/0/0.1, Pkts: 1509142,
Bytes: 78475384,
```

```
  Out: 100.0.2.1/3784 --> 100.0.2.2/49152;udp, Conn Tag: 0x0, If: .local..0, Pkts: 0, Bytes: 0,
```

```
Session ID: 6364, Policy name: self-traffic-policy/1, HA State: Active, Timeout: 60, Session
State: Valid
```

```
  In: fe80::2:2/49152 --> fe80::2:1/3784;udp, Conn Tag: 0x0, If: ip-0/0/0.1, Pkts: 1509355,
Bytes: 108673560,
```

```
  Out: fe80::2:1/3784 --> fe80::2:2/49152;udp, Conn Tag: 0x0, If: .local..0, Pkts: 0, Bytes: 0,
```

```
Total sessions: 6
```

node1:

-----

Session ID: 1304, Policy name: N/A, HA State: Backup, Timeout: N/A, Session State: Valid

In: 2012::2:2/1 --> 2012::2:1/1;ipip, Conn Tag: 0x0, If: reth1.0, Pkts: 0, Bytes: 0,

Session ID: 1305, Policy name: N/A, HA State: Backup, Timeout: N/A, Session State: Valid

In: 2012::2:2/1 --> 2012::2:1/1;ipv6, Conn Tag: 0x0, If: reth1.0, Pkts: 0, Bytes: 0,

Session ID: 1306, Policy name: self-traffic-policy/1, HA State: Backup, Timeout: 1482, Session State: Valid

In: 100.0.2.2/49152 --> 100.0.2.1/3784;udp, Conn Tag: 0x0, If: ip-0/0/0.1, Pkts: 0, Bytes: 0,

Out: 100.0.2.1/3784 --> 100.0.2.2/49152;udp, Conn Tag: 0x0, If: .local..0, Pkts: 0, Bytes: 0,

Session ID: 1307, Policy name: self-traffic-policy/1, HA State: Backup, Timeout: 1538, Session State: Valid

In: fe80::2:2/49152 --> fe80::2:1/3784;udp, Conn Tag: 0x0, If: ip-0/0/0.1, Pkts: 0, Bytes: 0,

Out: fe80::2:1/3784 --> fe80::2:2/49152;udp, Conn Tag: 0x0, If: .local..0, Pkts: 0, Bytes: 0,

Total sessions: 4

{primary:node0}

user@host> show security flow statistics

node0:

-----

Current sessions: 6

Packets received: 12528819

Packets transmitted: 12523469

Packets forwarded/queued: 44

Packets copied: 0

Packets dropped: 5306

Services-offload packets processed: 0

Fragment packets: 0

Pre fragments generated: 0

Post fragments generated: 0

node1:

-----

Current sessions: 4

Packets received: 1608551

Packets transmitted: 1588679

```

Packets forwarded/queued: 0
Packets copied: 0
Packets dropped: 19874
Services-offload packets processed: 0
Fragment packets: 0
Pre fragments generated: 0
Post fragments generated: 0

```

### Meaning

The chassis cluster configuration displays the reth interface as the bind interface to exchange routes through IP-over-IP tunnel.

### 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
12.1X45-D10	Starting with Junos OS Release 12.1X45-D10 and later, sampling features such as flow monitoring, packet capture, and port mirroring are supported on reth interfaces.

### RELATED DOCUMENTATION

[Example: Configure Chassis Clustering on SRX Series Firewalls | 122](#)

[Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Firewalls in a Chassis Cluster | 144](#)

## Configuring Chassis Clustering on SRX Series Devices

### IN THIS SECTION

- [Example: Configure Chassis Clustering on SRX Series Firewalls | 122](#)
- [Viewing a Chassis Cluster Configuration | 138](#)

- [Viewing Chassis Cluster Statistics | 139](#)
- [Clearing Chassis Cluster Statistics | 141](#)
- [Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes | 142](#)
- [Verifying Chassis Cluster Configuration Synchronization Status | 143](#)

SRX Series Services gateways can be configured to operate in cluster mode, where a pair of devices can be connected together and configured to operate like a single device to provide high availability. When configured as a chassis cluster, the two nodes back up each other, with one node acting as the primary device and the other as the secondary device, ensuring stateful failover of processes and services in the event of system or hardware failure. If the primary device fails, the secondary device takes over processing of traffic.

For SRX300, SRX320, SRX340, SRX345, and SRX380 devices, connect ge-0/0/1 on node 0 to ge-0/0/1 on node 1. The factory-default configuration does not include HA configuration. To enable HA, if the physical interfaces used by HA have some configurations, these configurations need to be removed. [Table 16 on page 121](#) lists the physical interfaces used by HA on SRX300, SRX320, SRX340, SRX345, and SRX380.

**Table 16: Mapping Between HA Interface and Physical Interface on SRX300, SRX320, SRX340, SRX345, and SRX380**

Device	fxp0 Interface (HA MGT)	fxp1 Interface (HA Control)	Fab Interface
SRX300	ge-0/0/0	ge-0/0/1	User defined
SRX320	ge-0/0/0	ge-0/0/1	User defined
SRX340	dedicated	ge-0/0/1	User defined
SRX345	dedicated	ge-0/0/1	User defined
SRX380	dedicated	ge-0/0/1	User defined

For more information, see the following topics:

## Example: Configure Chassis Clustering on SRX Series Firewalls

### IN THIS SECTION

- [Requirements | 122](#)
- [Overview | 123](#)
- [Configuration | 124](#)
- [Verification | 133](#)

This example shows how to set up chassis clustering on an SRX Series Firewall (using SRX1500 or SRX1600 as example).

### Requirements

Before you begin:

- Physically connect the two devices and ensure that they are the same models. For example, on the SRX1500 or SRX1600 Firewall, connect the dedicated control ports on node 0 and node 1.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:
  - On node 0:

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster-id is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster-id is 0 through 255 and setting it to 0 is equivalent to disabling cluster mode.

- After clustering occurs for the devices, continuing with the SRX1500 or SRX1600 Firewall example, the ge-0/0/0 interface on node 1 changes to ge-7/0/0.

After clustering occurs,



- For SRX300 devices, the ge-0/0/1 interface on node 1 changes to ge-1/0/1.
- For SRX320 devices, the ge-0/0/1 interface on node 1 changes to ge-3/0/1.
- For SRX340 and SRX345 devices, the ge-0/0/1 interface on node 1 changes to ge-5/0/1.

After the reboot, the following interfaces are assigned and repurposed to form a cluster:

- For SRX300 and SRX320 devices, ge-0/0/0 becomes fxp0 and is used for individual management of the chassis cluster.
- SRX340 and SRX345 devices contain a dedicated port fxp0.
- For all SRX300, SRX320, SRX340, SRX345, and SRX380 devices, ge-0/0/1 becomes fxp1 and is used as the control link within the chassis cluster.
- The other interfaces are also renamed on the secondary device.

See ["Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming" on page 19](#) for complete mapping of the SRX Series Firewalls.

From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device.

## Overview

This example shows how to set up chassis clustering on an SRX Series Firewall using the SRX1500 or SRX1600 device as example.

The node 1 renumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. See [Table 17 on page 123](#) for interface renumbering on the SRX Series Firewall.

**Table 17: SRX Series Firewalls Interface Renumbering**

SRX Series Services Gateway	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX300	1	ge-0/0/0	ge-1/0/0
SRX320	3	ge-0/0/0	ge-3/0/0
SRX340	5	ge-0/0/0	ge-5/0/0
SRX345			
SRX380			

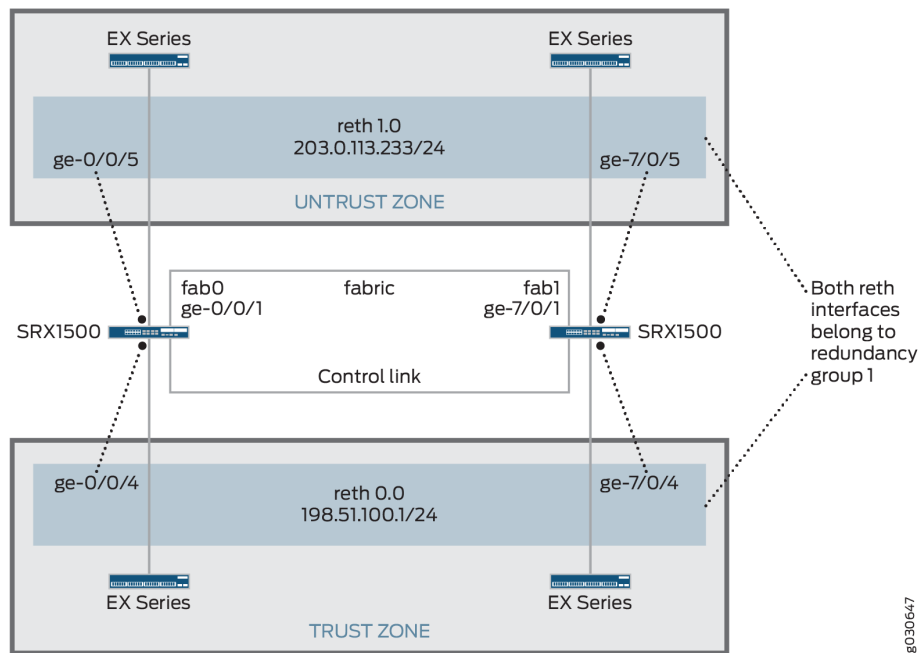
**Table 17: SRX Series Firewalls Interface Renumbering (Continued)**

SRX Series Services Gateway	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX1500	7	<b>ge-0/0/0</b>	<b>ge-7/0/0</b>
SRX1600	7	<b>ge-0/0/0</b>	<b>ge-7/0/0</b>

After clustering is enabled, the system creates fpx0, fpx1, and em0 interfaces. Depending on the device, the fpx0, fpx1, and em0 interfaces that are mapped to a physical interface are not user defined. However, the fab interface is user defined.

Figure 32 on page 124 shows the topology used in this example.

**Figure 32: SRX Series Firewalls (SRX1500) In Chassis Cluster**



## Configuration

### IN THIS SECTION

- Procedure | 125

8030647

## Procedure

### CLI Quick Configuration

To quickly configure a chassis cluster on an SRX1500 Firewall, copy the following commands and paste them into the CLI:

On {primary:node0}

```
[edit]
set groups node0 system host-name srx1500-1
set groups node0 interfaces fxp0 unit 0 family inet address 192.16.35.46/24
set groups node1 system host-name srx1500-2
set groups node1 interfaces fxp0 unit 0 family inet address 192.16.35.47/24
set groups node0 system backup-router <backup next-hop from fxp0> destination <management network/mask>
set groups node1 system backup-router <backup next-hop from fxp0> destination <management network/mask>
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-7/0/1
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/5 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/4 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-7/0/5 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-7/0/4 weight 255
set chassis cluster reth-count 2
set interfaces ge-0/0/5 gigether-options redundant-parent reth1
set interfaces ge-7/0/5 gigether-options redundant-parent reth1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 203.0.113.233/24
set interfaces ge-0/0/4 gigether-options redundant-parent reth0
set interfaces ge-7/0/4 gigether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 198.51.100.1/24
set security zones security-zone Untrust interfaces reth1.0
set security zones security-zone Trust interfaces reth0.0
```

If you are configuring SRX300, SRX320, SRX340, SRX345, and SRX380 devices, see [Table 18 on page 126](#) for command and interface settings for your device and substitute these commands into your CLI.

**Table 18: SRX Series Firewalls Interface Settings**

Command	SRX300	SRX320	SRX340 SRX345 SRX380
set interfaces fab0 fabric-options member-interfaces	ge-0/0/2	ge-0/0/2	ge-0/0/2
set interfaces fab1 fabric-options member-interfaces	ge-1/0/2	ge-3/0/2	ge-5/0/2
set chassis cluster redundancy-group 1 interface-monitor	ge-0/0/3 weight 255	ge-0/0/3 weight 255	ge-0/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor	ge-0/0/4 weight 255	ge-0/0/4 weight 255	ge-0/0/4 weight 255
set chassis cluster redundancy-group 1 interface-monitor	ge-1/0/3 weight 255	ge-3/0/3 weight 255	ge-5/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor	ge-1/0/4 weight 255	ge-3/0/4 weight 255	ge-5/0/4 weight 255
set interfaces	ge-0/0/3 gigether-options redundant-parent reth0	ge-0/0/3 gigether-options redundant-parent reth0	ge-0/0/3 gigether-options redundant-parent reth0

**Table 18: SRX Series Firewalls Interface Settings (Continued)**

Command	SRX300	SRX320	SRX340 SRX345 SRX380
set interfaces	ge-0/0/4 gige-ther- options redundant-parent reth1	ge-0/0/4 gige-ther- options redundant-parent reth1	ge-0/0/4 gige-ther-options redundant-parent reth1
set interfaces	ge-1/0/3 gige-ther- options redundant-parent reth0	ge-3/0/3 gige-ther- options redundant-parent reth0	ge-5/0/3 gige-ther-options redundant-parent reth0
set interfaces	ge-1/0/4 gige-ther- options redundant-parent reth1	ge-3/0/4 gige-ther- options redundant-parent reth1	ge-5/0/4 gige-ther-options redundant-parent reth1

### 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*.

To configure a chassis cluster on an SRX Series Firewall:

Perform Steps 1 through 5 on the primary device (node 0). They are automatically copied over to the secondary device (node 1) when you execute a `commit` command. The configurations are synchronized because the control link and fab link interfaces are activated. To verify the configurations, use the `show interface terse` command and review the output.

1. Set up hostnames and management IP addresses for each device using configuration groups. These configurations are specific to each device and are unique to its specific node.

```

user@host# set groups node0 system host-name srx1500-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 192.16.35.46/24
user@host# set groups node1 system host-name srx1500-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 192.16.35.47/24

```

Set the default route and backup router for each node.

```
user@host# set groups node0 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
user@host# set groups node1 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
```

Set the apply-group command so that the individual configurations for each node set by the previous commands are applied only to that node.

```
user@host# set apply-groups "${node}"
```

2. Define the interfaces used for the fab connection (data plane links for RTO sync) by using physical ports ge-0/0/1 from each node. These interfaces must be connected back-to-back, or through a Layer 2 infrastructure.

```
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/1
```

3. Set up redundancy group 0 for the Routing Engine failover properties, and set up redundancy group 1 (all interfaces are in one redundancy group in this example) to define the failover properties for the redundant Ethernet interfaces.

```
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
```

4. Set up interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.

We do not recommend Interface monitoring for redundancy group 0 because it causes the control plane to switch from one node to another node in case interface flap occurs.

```
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/5 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/4 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/5 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/4 weight 255
```

Interface failover only occurs after the weight reaches 0.

5. Set up the redundant Ethernet (reth) interfaces and assign the redundant interface to a zone.

```

user@host# set chassis cluster reth-count 2
user@host# set interfaces ge-0/0/5 gigether-options redundant-parent reth1
user@host# set interfaces ge-7/0/5 gigether-options redundant-parent reth1
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 203.0.113.233/24
user@host# set interfaces ge-0/0/4 gigether-options redundant-parent reth0
user@host# set interfaces ge-7/0/4 gigether-options redundant-parent reth0
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 198.51.100.1/24
user@host# set security zones security-zone Untrust interfaces reth1.0
user@host# set security zones security-zone Trust interfaces reth0.0

```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX1500-1;
      backup-router 10.100.22.1 destination 66.129.243.0/24;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.16.35.46/24;
          }
        }
      }
    }
  }
}

```

```

    }
  }
  node1 {
    system {
      host-name SRX1500-2;
      backup-router 10.100.21.1 destination 66.129.243.0/24;    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.16.35.47/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
chassis {
  cluster {
    reth-count 2;
    redundancy-group 0 {
      node 0 priority 100;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 100;
      node 1 priority 1;
      interface-monitor {
        ge-0/0/5 weight 255;
        ge-0/0/4 weight 255;
        ge-7/0/5 weight 255;
        ge-7/0/4 weight 255;
      }
    }
  }
}
interfaces {
  ge-0/0/5 {
    gigheter-options {
      redundant-parent reth1;
    }
  }
}

```



```
ge-0/0/4 {
  gigether-options {
    redundant-parent reth0;
  }
}
ge-7/0/5 {
  gigether-options {
    redundant-parent reth1;
  }
}
ge-7/0/4 {
  gigether-options {
    redundant-parent reth0;
  }
}
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/1;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-7/0/1;
    }
  }
}
reth0 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 198.51.100.1/24;
    }
  }
}
reth1 {
  redundant-ether-options {
    redundancy-group 1;
  }
}
```

```
    unit 0 {
        family inet {
            address 203.0.113.233/24;
        }
    }
}
...
security {
    zones {
        security-zone Untrust {
            interfaces {
                reth1.0;
            }
        }
        security-zone Trust {
            interfaces {
                reth0.0;
            }
        }
    }
    policies {
        from-zone Trust to-zone Untrust {
            policy 1 {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
}
```

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

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Status | 133](#)
- [Verifying Chassis Cluster Interfaces | 134](#)
- [Verifying Chassis Cluster Statistics | 134](#)
- [Verifying Chassis Cluster Control Plane Statistics | 135](#)
- [Verifying Chassis Cluster Data Plane Statistics | 136](#)
- [Verifying Chassis Cluster Redundancy Group Status | 137](#)
- [Troubleshooting with Logs | 137](#)

Confirm that the configuration is working properly.

### Verifying Chassis Cluster Status

#### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

#### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host# show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
node0                100       primary  no       no
node1                1         secondary no       no

Redundancy group: 1 , Failover count: 1
node0                0         primary  no       no
node1                0         secondary no       no
```

## Verifying Chassis Cluster Interfaces

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: em0

Redundant-ethernet Information:
  Name      Status   Redundancy-group
  reth0     Up       1
  reth1     Up       1

Interface Monitoring:
  Interface  Weight  Status  Redundancy-group
  ge-7/0/5   255    Up      1
  ge-7/0/4   255    Up      1
  ge-0/0/5   255    Up      1
  ge-0/0/4   255    Up      1
```

## Verifying Chassis Cluster Statistics

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster statistics` command.

```
{primary:node0}
user@host> show chassis cluster statistics
```

## Control link statistics:

## Control link 0:

Heartbeat packets sent: 2276

Heartbeat packets received: 2280

Heartbeat packets errors: 0

## Fabric link statistics:

## Child link 0

Probes sent: 2272

Probes received: 597

## Services Synchronized:

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

## Verifying Chassis Cluster Control Plane Statistics

### Purpose

Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

## Action

From operational mode, enter the `show chassis cluster control-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics
```

```
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 2294
    Heartbeat packets received: 2298
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 2290
    Probes received: 615
```

## Verifying Chassis Cluster Data Plane Statistics

### Purpose

Verify information about the number of RTOs sent and received for services.

### Action

From operational mode, enter the `show chassis cluster data-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics
```

Services Synchronized:		
Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0

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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

## Verifying Chassis Cluster Redundancy Group Status

### Purpose

Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

### Action

From operational mode, enter the chassis cluster status redundancy-group command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority  Status  Preempt  Manual failover

Redundancy group: 1, Failover count: 1
node0           100     primary no       no
node1           50      secondary no       no
```

## Troubleshooting with Logs

### Purpose

Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

## Action

From operational mode, enter these `show log` commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

## SEE ALSO

[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming | 19](#)

## Viewing a Chassis Cluster Configuration

### IN THIS SECTION

- [Purpose | 138](#)
- [Action | 138](#)

## Purpose

Display chassis cluster verification options.

## Action

From the CLI, enter the `show chassis cluster ?` command:

```
{primary:node1}
user@host> show chassis cluster ?
Possible completions:
  interfaces          Display chassis-cluster interfaces
```



statistics	Display chassis-cluster traffic statistics
status	Display chassis-cluster status

## Viewing Chassis Cluster Statistics

### IN THIS SECTION

- Purpose | 139
- Action | 139

### Purpose

Display information about chassis cluster services and interfaces.

### Action

From the CLI, enter the `show chassis cluster statistics` command:

```
{primary:node1}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 798
    Heartbeat packets received: 784
Fabric link statistics:
  Child link 0
    Probes sent: 793
    Probes received: 0
Services Synchronized:
  Service name          RT0s sent  RT0s received
  Translation context    0          0
  Incoming NAT           0          0
  Resource manager       0          0
  Session create         0          0
```

Session close	0	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
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
RTSP ALG	0	0

```
{primary:node1}
```

```
user@host> show chassis cluster statistics
```

```
Control link statistics:
```

```
Control link 0:
```

```
Heartbeat packets sent: 258689
```

```
Heartbeat packets received: 258684
```

```
Control link 1:
```

```
Heartbeat packets sent: 258689
```

```
Heartbeat packets received: 258684
```

```
Fabric link statistics:
```

```
Child link 0
```

```
Probes sent: 258681
```

```
Probes received: 258681
```

```
Child link 1
```

```
Probes sent: 258501
```

```
Probes received: 258501
```

```
Services Synchronized:
```

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	0	0
Session create	1	0
Session close	1	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0

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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

```
{primary:node1}
user@host> show chassis cluster statistics
```

Control link statistics:

Control link 0:

Heartbeat packets sent: 82371

Heartbeat packets received: 82321

Control link 1:

Heartbeat packets sent: 0

Heartbeat packets received: 0

## Clearing Chassis Cluster Statistics

To clear displayed information about chassis cluster services and interfaces, enter the `clear chassis cluster statistics` command from the CLI:

```
{primary:node1}
user@host> clear chassis cluster statistics
```

Cleared control-plane statistics

Cleared data-plane statistics

## Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes

When you set up an SRX Series chassis cluster, the SRX Series Firewalls must be identical, including their configuration. The chassis cluster synchronization feature automatically synchronizes the configuration from the primary node to the secondary node when the secondary joins the primary as a cluster. By eliminating the manual work needed to ensure the same configurations on each node in the cluster, this feature reduces expenses.

If you want to disable automatic chassis cluster synchronization between the primary and secondary nodes, you can do so by entering the `set chassis cluster configuration-synchronize no-secondary-bootup-auto` command in configuration mode.

At any time, to reenable automatic chassis cluster synchronization, use the `delete chassis cluster configuration-synchronize no-secondary-bootup-auto` command in configuration mode.

To see whether the automatic chassis cluster synchronization is enabled or not, and to see the status of the synchronization, enter the `show chassis cluster information configuration-synchronization operational` command.

Either the entire configuration from the primary node is applied successfully to the secondary node, or the secondary node retains its original configuration. There is no partial synchronization.

If you create a cluster with cluster IDs greater than 16, and then decide to roll back to a previous release image that does not support extended cluster IDs, the system comes up as standalone.

If you have a cluster set up and running with an earlier release of Junos OS, you can upgrade to Junos OS Release 12.1X45-D10 and re-create a cluster with cluster IDs greater than 16. However, if for any reason you decide to revert to the previous version of Junos OS that did not support extended cluster IDs, the system comes up with standalone devices after you reboot. However, if the cluster ID set is less than 16 and you roll back to a previous release, the system will come back with the previous setup.

### SEE ALSO

[NTP Time Synchronization on SRX Series Devices](#) | 333

## Verifying Chassis Cluster Configuration Synchronization Status

### IN THIS SECTION

- Purpose | 143
- Action | 143

### Purpose

Display the configuration synchronization status of a chassis "[Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes](#)" on page 142cluster.

### Action

From the CLI, enter the show chassis cluster information configuration-synchronization command:

```
{primary:node0}
user@host> show chassis cluster information configuration-synchronization

node0:
-----

Configuration Synchronization:
  Status:
    Activation status: Enabled
    Last sync operation: Auto-Sync
    Last sync result: Not needed
    Last sync mgd messages:

  Events:
    Mar  5 01:48:53.662 : Auto-Sync: Not needed.

node1:
-----

Configuration Synchronization:
  Status:
```

```
Activation status: Enabled
Last sync operation: Auto-Sync
Last sync result: Succeeded
Last sync mgd messages:
  mgd: rcp: /config/juniper.conf: No such file or directory
  mgd: commit complete
```

**Events:**

```
Mar 5 01:48:55.339 : Auto-Sync: In progress. Attempt: 1
Mar 5 01:49:40.664 : Auto-Sync: Succeeded. Attempt: 1
```

**SEE ALSO**

| [\*show chassis cluster information configuration-synchronization\*](#)

**RELATED DOCUMENTATION**

| [Preparing Your Equipment for Chassis Cluster Formation | 35](#)

| [Connecting SRX Series Firewalls to Create a Chassis Cluster | 39](#)

| [SRX Series Chassis Cluster Configuration Overview | 14](#)

## Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Firewalls in a Chassis Cluster

**IN THIS SECTION**

- [Requirements | 145](#)
- [Overview | 145](#)
- [Configuration | 147](#)
- [Verification | 155](#)

This example shows how to enable eight-queue CoS on redundant Ethernet interfaces on SRX Series Firewalls in a chassis cluster. This example is applicable to SRX1600, SRX2300, SRX4100, SRX4200, SRX4300, SRX4600, SRX5400, SRX5600, and SRX5800. The eight-queue CoS is also supported on redundant Ethernet interfaces for branch SRX Series Firewalls in a chassis cluster. The SRX Series for the branch support eight queues, but only four queues are enabled by default.

## Requirements

This example uses the following hardware and software components:

- Two SRX5600 Service Gateways in a chassis cluster
- Junos OS Release 11.4R4 or later for SRX Series Firewalls

Before you begin:

- Understand chassis cluster configuration. See ["Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Devices" on page 342.](#)
- Understand chassis cluster redundant interface configuration. See ["Example: Configuring Chassis Cluster Redundant Ethernet Interfaces" on page 95.](#)

## Overview

### IN THIS SECTION

- [Topology | 146](#)

The SRX Series Firewalls support eight queues, but only four queues are enabled by default. Use the `set chassis fpc  $x$  pic  $y$  max-queues-per-interface 8` command to enable eight queues explicitly at the chassis level. The values of  $x$  and  $y$  depends on the location of the IOC and the PIC number where the interface is located on the device on which CoS needs to be implemented. To find the IOC location use the `show chassis fpc pic-status` or `show chassis hardware` commands.

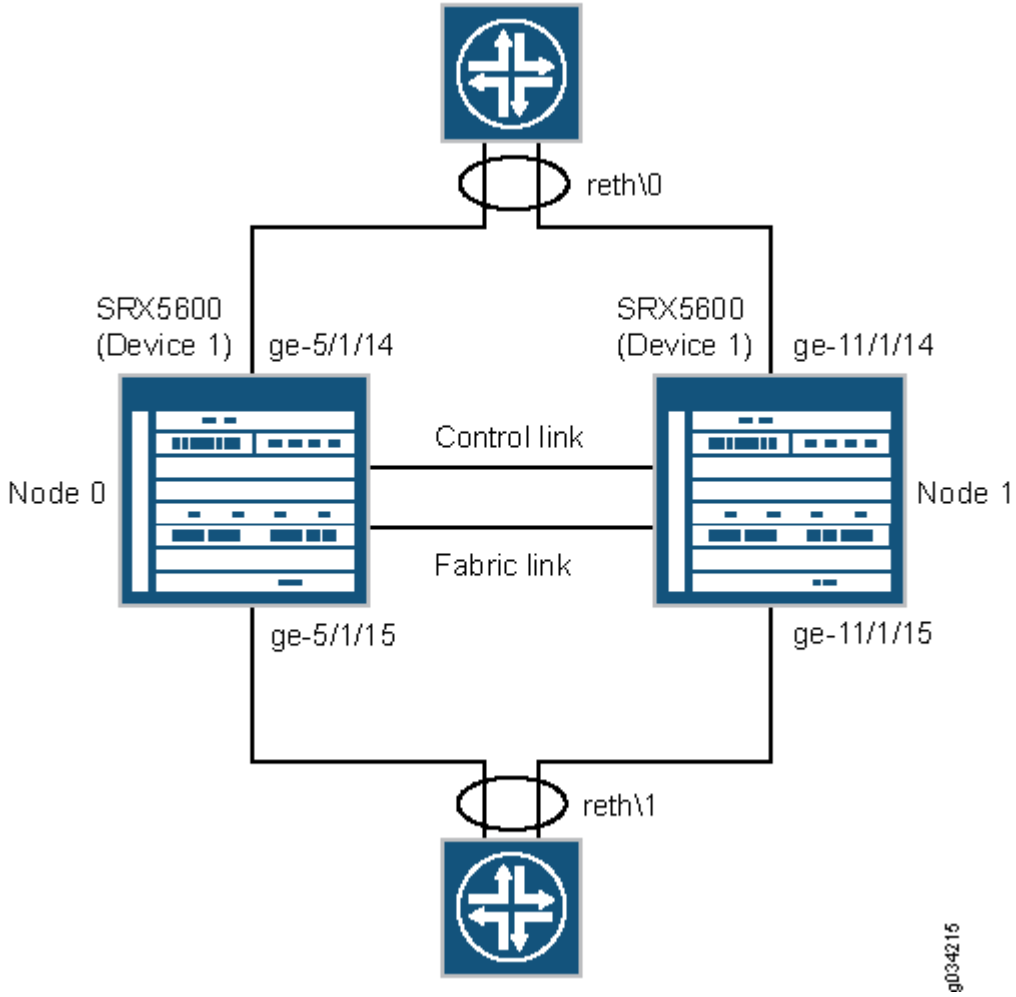
You must restart the chassis control for the configuration to take effect.

On SRX Series Firewalls, eight QoS queues are supported per ae interface.

Figure 33 on page 146 shows how to configure eight-queue CoS on redundant Ethernet interfaces on SRX Series Firewalls in a chassis cluster.

### Topology

Figure 33: Eight-Queue CoS on Redundant Ethernet Interfaces





## Configuration

### IN THIS SECTION

- [Procedure | 147](#)

## 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set chassis fpc 5 pic 1 max-queues-per-interface 8
set chassis fpc 5 pic 1 max-queues-per-interface 8
set chassis cluster reth-count 2
set chassis cluster control-ports fpc 4 port 0
set chassis cluster control-ports fpc 10 port 0
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
set interfaces ge-5/1/14 gigether-options redundant-parent reth0
set interfaces ge-5/1/15 gigether-options redundant-parent reth1
set interfaces ge-11/1/14 gigether-options redundant-parent reth0
set interfaces ge-11/1/15 gigether-options redundant-parent reth1
set interfaces reth0 vlan-tagging
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 vlan-id 1350
set interfaces reth0 unit 0 family inet address 192.0.2.1/24
set interfaces reth1 hierarchical-scheduler
set interfaces reth1 vlan-tagging
set interfaces reth1 redundant-ether-options redundancy-group 2
set interfaces reth1 unit 0 vlan-id 1351
set interfaces reth1 unit 0 family inet address 192.0.2.2/24
set interfaces reth1 unit 1 vlan-id 1352
set interfaces reth1 unit 1 family inet address 192.0.2.3/24
```

```
set interfaces reth1 unit 2 vlan-id 1353
set interfaces reth1 unit 2 family inet address 192.0.2.4/24
set interfaces reth1 unit 3 vlan-id 1354
set interfaces reth1 unit 3 family inet address 192.0.2.5/24
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q0 loss-priority
low code-points 000
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q2 loss-priority
low code-points 010
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q3 loss-priority
low code-points 011
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q1 loss-priority
low code-points 001
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q4 loss-priority
low code-points 100
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q5 loss-priority
low code-points 101
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q6 loss-priority
low code-points 110
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q7 loss-priority
low code-points 111
set class-of-service forwarding-classes queue 0 q0
set class-of-service forwarding-classes queue 1 q1
set class-of-service forwarding-classes queue 2 q2
set class-of-service forwarding-classes queue 3 q3
set class-of-service forwarding-classes queue 4 q4
set class-of-service forwarding-classes queue 5 q5
set class-of-service forwarding-classes queue 6 q6
set class-of-service forwarding-classes queue 7 q7
set class-of-service traffic-control-profiles 1 scheduler-map sched_map
set class-of-service traffic-control-profiles 1 shaping-rate 200m
set class-of-service interfaces reth0 unit 0 classifiers inet-precedence inet_prec_4
set class-of-service interfaces reth1 unit 0 output-traffic-control-profile 1
set class-of-service scheduler-maps sched_map forwarding-class q0 scheduler S0
set class-of-service scheduler-maps sched_map forwarding-class q1 scheduler S1
set class-of-service scheduler-maps sched_map forwarding-class q2 scheduler S2
set class-of-service scheduler-maps sched_map forwarding-class q3 scheduler S3
set class-of-service scheduler-maps sched_map forwarding-class q4 scheduler S4
set class-of-service scheduler-maps sched_map forwarding-class q5 scheduler S5
set class-of-service scheduler-maps sched_map forwarding-class q6 scheduler S6
set class-of-service scheduler-maps sched_map forwarding-class q7 scheduler S7
set class-of-service schedulers S0 transmit-rate percent 20
set class-of-service schedulers S1 transmit-rate percent 5
set class-of-service schedulers S2 transmit-rate percent 5
```

```

set class-of-service schedulers S3 transmit-rate percent 10
set class-of-service schedulers S4 transmit-rate percent 10
set class-of-service schedulers S5 transmit-rate percent 10
set class-of-service schedulers S6 transmit-rate percent 10
set class-of-service schedulers S7 transmit-rate percent 30

```

## 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.

To enable eight-queue CoS on redundant Ethernet interfaces:

1. Configure a maximum of eight queues on the interfaces on Node 0 and Node 1.

```

[edit chassis]
user@host# set fpc 5 pic 1 max-queues-per-interface 8

```

In addition to configuring eight queues at the [edit chassis] hierarchy level, the configuration at the [edit class-of-service] hierarchy level must support eight queues per interface.

2. Specify the number of redundant Ethernet interfaces.

```

[edit chassis cluster]
user@host# set reth-count 2

```

3. Configure the control ports.

```

[edit chassis cluster]
user@host# set control-ports fpc 4 port 0
user@host# set control-ports fpc 10 port 0

```

4. Configure redundancy groups.

```

[edit chassis cluster]
user@host# set redundancy-group 0 node 0 priority 254
user@host# set redundancy-group 0 node 1 priority 1
user@host# set redundancy-group 1 node 0 priority 200
user@host# set redundancy-group 1 node 1 priority 100

```

5. Configure the redundant Ethernet interfaces.

```
[edit interfaces]
user@host# set ge-5/1/14 gigether-options redundant-parent reth0
user@host# set ge-11/1/14 gigether-options redundant-parent reth0
user@host# set ge-5/1/15 gigether-options redundant-parent reth1
user@host# set ge-11/1/15 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 vlan-tagging
user@host# set reth0 unit 0 vlan-id 1350
user@host# set reth0 unit 0 family inet address 192.0.2.1/24
user@host# set reth1 hierarchical-scheduler
user@host# set reth1 vlan-tagging
user@host# set reth1 redundant-ether-options redundancy-group 2
user@host# set reth1 unit 0 vlan-id 1351
user@host# set reth1 unit 0 family inet address 192.0.2.2/24
user@host# set reth1 unit 1 vlan-id 1352
user@host# set reth1 unit 1 family inet address 192.0.2.3/24
user@host# set reth1 unit 2 vlan-id 1353
user@host# set reth1 unit 2 family inet address 192.0.2.4/24
user@host# set reth1 unit 3 vlan-id 1354
user@host# set reth1 unit 3 family inet address 192.0.2.5/24
```

6. Define a classifier and apply it to a logical interface.

```
[edit class-of-service]
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q0 loss-priority
low code-points 000
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q2 loss-priority
low code-points 010
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q3 loss-priority
low code-points 011
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q1 loss-priority
low code-points 001
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q4 loss-priority
low code-points 100
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q5 loss-priority
low code-points 101
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q6 loss-priority
low code-points 110
```

```
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q7 loss-priority
low code-points 111
```

7. Map forwarding classes to CoS queues.

```
[edit class-of-service]
user@host# set forwarding-classes queue 0 q0
user@host# set forwarding-classes queue 1 q1
user@host# set forwarding-classes queue 2 q2
user@host# set forwarding-classes queue 3 q3
user@host# set forwarding-classes queue 4 q4
user@host# set forwarding-classes queue 5 q5
user@host# set forwarding-classes queue 6 q6
user@host# set forwarding-classes queue 7 q7
```

8. Configure traffic control profiles.

```
[edit class-of-service]
user@host# set traffic-control-profiles 1 scheduler-map sched_map
user@host# set traffic-control-profiles 1 shaping-rate 200m
```

9. Define packet flow through the CoS elements.

```
[edit class-of-service]
user@host# set interfaces reth0 unit 0 classifiers inet-precedence inet_prec_4
```

10. Apply a traffic scheduling profile to the interface.

```
[edit class-of-service]
user@host# set interfaces reth1 unit 0 output-traffic-control-profile 1
```

11. Configure the CoS schedulers.

```
[edit class-of-service]
user@host# set scheduler-maps sched_map forwarding-class q0 scheduler S0
user@host# set scheduler-maps sched_map forwarding-class q1 scheduler S1
user@host# set scheduler-maps sched_map forwarding-class q2 scheduler S2
user@host# set scheduler-maps sched_map forwarding-class q3 scheduler S3
```

```

user@host# set scheduler-maps sched_map forwarding-class q4 scheduler S4
user@host# set scheduler-maps sched_map forwarding-class q5 scheduler S5
user@host# set scheduler-maps sched_map forwarding-class q6 scheduler S6
user@host# set scheduler-maps sched_map forwarding-class q7 scheduler S7
user@host# set schedulers S0 transmit-rate percent 20
user@host# set schedulers S1 transmit-rate percent 5
user@host# set schedulers S2 transmit-rate percent 5
user@host# set schedulers S3 transmit-rate percent 10
user@host# set schedulers S4 transmit-rate percent 10
user@host# set schedulers S5 transmit-rate percent 10
user@host# set schedulers S6 transmit-rate percent 10
user@host# set schedulers S7 transmit-rate percent 30

```

## Results

From configuration mode, confirm your configuration by entering the `show class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

[edit]
user@host# show class-of-service
classifiers {
    inet-precedence inet_prec_4 {
        forwarding-class q0 {
            loss-priority low code-points 000;
        }
        forwarding-class q2 {
            loss-priority low code-points 010;
        }
        forwarding-class q3 {
            loss-priority low code-points 011;
        }
        forwarding-class q1 {
            loss-priority low code-points 001;
        }
        forwarding-class q4 {
            loss-priority low code-points 100;
        }
        forwarding-class q5 {

```

```
        loss-priority low code-points 101;
    }
    forwarding-class q6 {
        loss-priority low code-points 110;
    }
    forwarding-class q7 {
        loss-priority low code-points 111;
    }
}
forwarding-classes {
    queue 0 q0;
    queue 1 q1;
    queue 2 q2;
    queue 3 q3;
    queue 4 q4;
    queue 5 q5;
    queue 6 q6;
    queue 7 q7;
}
traffic-control-profiles {
    1 {
        scheduler-map sched_map;
        shaping-rate 200m;
    }
}
interfaces {
    reth0 {
        unit 0 {
            classifiers {
                inet-precedence inet_prec_4;
            }
        }
    }
    reth1 {
        unit 0 {
            output-traffic-control-profile 1;
        }
    }
}
scheduler-maps {
    sched_map {
        forwarding-class q0 scheduler S0;
```

```

        forwarding-class q1 scheduler S1;
        forwarding-class q2 scheduler S2;
        forwarding-class q3 scheduler S3;
        forwarding-class q4 scheduler S4;
        forwarding-class q5 scheduler S5;
        forwarding-class q6 scheduler S6;
        forwarding-class q7 scheduler S7;
    }
}
schedulers {
    S0 {
        transmit-rate percent 20;
    }
    S1 {
        transmit-rate percent 5;
    }
    S2 {
        transmit-rate percent 5;
    }
    S3 {
        transmit-rate percent 10;
    }
    S4 {
        transmit-rate percent 10;
    }
    S5 {
        transmit-rate percent 10;
    }
    S6 {
        transmit-rate percent 10;
    }
    S7 {
        transmit-rate percent 30;
    }
}
}

```

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

To restart chassis control, enter `restart chassis-control` command from operational mode.

When you execute the `restart chassis-control` command all the FRU cards on the box are reset, thus impacting traffic. Changing the number of queues must be executed during a scheduled downtime. It takes 5-10 minutes for the cards to come online after the `restart chassis-control` command is executed.



## Verification

### IN THIS SECTION

- [Verifying the Eight-Queue COS Configuration | 155](#)

## Verifying the Eight-Queue COS Configuration

### Purpose

Verify that eight-queue CoS is enabled properly.

### Action

From the operational mode, enter the following commands:

- `show interfaces ge-5/1/14 extensive`
- `show interfaces queue ge-5/1/14`
- `show class-of-service forwarding-class`
- `show class-of-service interface ge-5/1/14`

### RELATED DOCUMENTATION

[SRX Series Chassis Cluster Configuration Overview | 14](#)

[Chassis Cluster Redundant Ethernet Interfaces | 91](#)

# Conditional Route Advertisement over Redundant Ethernet Interfaces on SRX Series Firewalls in a Chassis Cluster

## IN THIS SECTION

- [Understanding Conditional Route Advertising in a Chassis Cluster | 156](#)
- [Example: Configuring Conditional Route Advertising in a Chassis Cluster | 157](#)

Conditional route advertising allows you to add criteria on route advertisements before they are installed in the route table or advertised to peers and neighbors. The goal of conditional route advertisement in a chassis cluster is to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface. For more information, see the following topics:

## Understanding Conditional Route Advertising in a Chassis Cluster

Route advertisement over redundant Ethernet interfaces in a *chassis cluster* is complicated by the fact that the active node in the cluster can change dynamically. Conditional route advertisement enables you to advertise routes in such a way that incoming traffic from the core network is attracted to the Border Gateway Protocol (BGP) interface that exists on the same node as the currently active redundant Ethernet interface. In this way, traffic is processed by the active node and does not traverse the fabric interface between nodes. You do this by manipulating the BGP attribute at the time routes are advertised by BGP.

The goal of conditional route advertisement in a chassis cluster is to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface. To understand how this works, keep in mind that in a chassis cluster, each node has its own set of interfaces.

## SEE ALSO

[Example: Configuring Conditional Route Advertising in a Chassis Cluster | 157](#)

[Viewing a Chassis Cluster Configuration | 138](#)

[Viewing Chassis Cluster Statistics | 139](#)

## Example: Configuring Conditional Route Advertising in a Chassis Cluster

### IN THIS SECTION

- [Requirements | 157](#)
- [Overview | 157](#)
- [Configuration | 159](#)

This example shows how to configure conditional route advertising in a chassis cluster to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface.

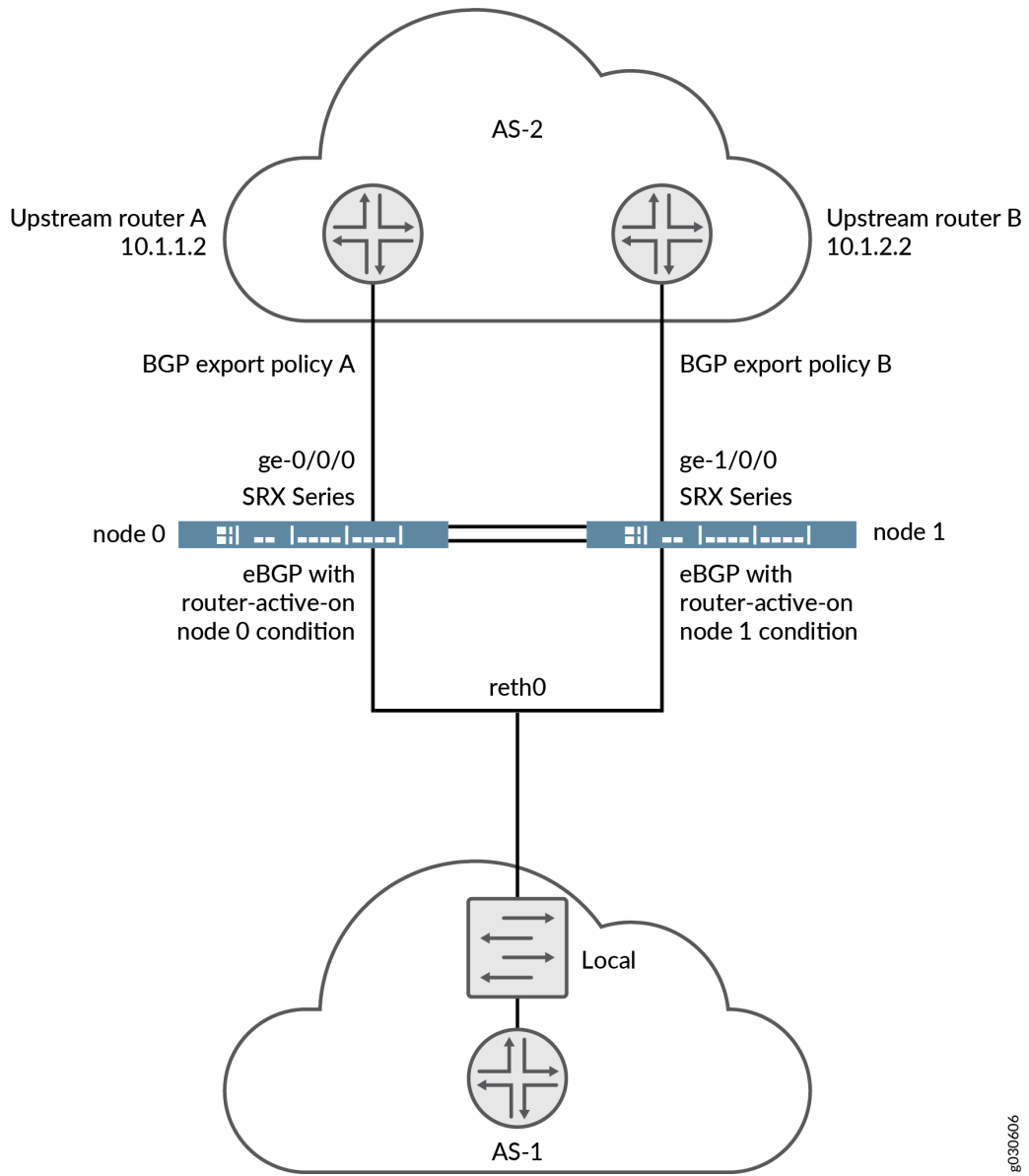
### Requirements

Before you begin, understand conditional route advertising in a chassis cluster. See "[Understanding Conditional Route Advertising in a Chassis Cluster](#)" on page 156.

### Overview

As illustrated in [Figure 34 on page 158](#), routing prefixes learned from the redundant Ethernet interface through the IGP are advertised toward the network core using BGP. Two BGP sessions are maintained, one from interface ge-0/0/0 and one from ge-1/0/0 for BGP multihoming. All routing prefixes are advertised on both sessions. Thus, for a route advertised by BGP, learned over a redundant Ethernet interface, if the active redundant Ethernet interface is on the same node as the BGP session, you advertise the route with a "good" BGP attribute.

Figure 34: Conditional Route Advertising on SRX Series Firewalls in a Chassis Cluster



To achieve this behavior, you apply a policy to BGP before exporting routes. An additional term in the policy match condition determines the current active redundant Ethernet interface child interface of the next hop before making the routing decision. When the active status of a child redundant Ethernet interface changes, BGP reevaluates the export policy for all routes affected.

The condition statement in this configuration works as follows. The command states that any routes evaluated against this condition will pass only if:

- The routes have a redundant Ethernet interface as their next-hop interface.

- The current child interface of the redundant Ethernet interface is active at node 0 (as specified by the `route-active-on node0` keyword).

```
{primary:node0}[edit]
user@host# set policy-options condition reth-nh-active-on-0 route-active-on node0
```

Note that a route might have multiple equal-cost next hops, and those next hops might be redundant Ethernet interfaces, regular interfaces, or a combination of both. The route still satisfies the requirement that it has a redundant Ethernet interface as its next hop.

If you use the BGP export policy set for node 0 in the previous example command, only OSPF routes that satisfy the following requirements will be advertised through the session:

- The OSPF routes have a redundant Ethernet interface as their next hop.
- The current child interface of the redundant Ethernet interface is currently active at node 0.

You must also create and apply a separate policy statement for the other BGP session by using this same process.

In addition to the BGP MED attribute, you can define additional BGP attributes, such as `origin-code`, `as-path`, and `community`.

## Configuration

### IN THIS SECTION

- [Procedure | 159](#)

## 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, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 from protocol ospf
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 from condition reth-nh-
```

```

active-on-0
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 then metric 10
set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 then accept
set policy-options condition reth-nh-active-on-0 route-active-on node0

```

## Step-by-Step Procedure

To configure conditional route advertising:

- Create the export policies with the created condition using the condition statement.

```

{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 from
protocol ospf
{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 from
condition reth-nh-active-on-0
{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 then metric
10
{primary:node0}[edit]
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0 then accept
{primary:node0}[edit]
user@host# set policy-options condition reth-nh-active-on-0 route-active-on node0

```

## Results

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

```

{primary:node0}[edit]
user@host# show policy-options
policy-statement reth-nh-active-on-0 {
  term ospf-on-0 {
    from {
      protocol ospf;
      condition reth-nh-active-on-0;
    }
  }
}

```

```
        then {
            metric 10;
            accept;
        }
    }
}
condition reth-nh-active-on-0 route-active-on node0;
```

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

## SEE ALSO

[Understanding Conditional Route Advertising in a Chassis Cluster | 156](#)

[Viewing a Chassis Cluster Configuration | 138](#)

[Viewing Chassis Cluster Statistics | 139](#)

# 3

CHAPTER

## Configuring Redundancy and Failover in a Chassis Cluster

---

[Chassis Cluster Dual Control Links | 163](#)

[Chassis Cluster Dual Fabric Links | 172](#)

[Monitoring of Global-Level Objects in a Chassis Cluster | 181](#)

[Monitoring Chassis Cluster Interfaces | 185](#)

[Monitoring IP Addresses on a Chassis Cluster | 227](#)

[Configuring Cluster Failover Parameters | 248](#)

[Understanding Chassis Cluster Resiliency | 253](#)

[Chassis Cluster Redundancy Group Failover | 254](#)

---



# Chassis Cluster Dual Control Links

## IN THIS SECTION

- [Chassis Cluster Dual Control Links Overview | 163](#)
- [Dual Control Link Connections for SRX Series Firewalls in a Chassis Cluster | 165](#)
- [Upgrade the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices | 166](#)
- [Example: Configure Chassis Cluster Control Ports for Dual Control Links | 169](#)

Dual control links provide a redundant link for controlling network traffic.

## Chassis Cluster Dual Control Links Overview

### IN THIS SECTION

- [Benefit of Dual Control Links | 164](#)
- [Dual Control Link Functionality Requirements | 164](#)

A control link connects two SRX Series Firewalls and sends chassis cluster control data, including heartbeats and configuration synchronization, between them. The link is a single point of failure: If the control link goes down, the secondary SRX Series is disabled from the cluster.

Dual control links prevent downtime due to a single point of failure. Two control link interfaces connect each device in a cluster. Dual control links provide a redundant link for controlling traffic. Unlike dual fabric links, only one control link is used at any one time.

The SRX1600, SRX2300, SRX4300, SRX4600, SRX5600, and SRX5800 Services Gateways support dual control links.

We do not support dual control link functionality on these Services Gateways: SRX4100, SRX4200, or SRX5400.

Starting with Junos OS Release 20.4R1, you can enable or disable the control links on SRX1500 Services Gateways using operational mode CLI commands and configuration mode CLI commands, described in a subsequent paragraph. This CLI feature enables you to control the status of cluster nodes during a cluster upgrade.

Previously, if you wanted to disable the control link and fabric link, you had to unplug the cables manually.

The CLI commands work as follows:

- **In configuration mode**

- To disable the control link, run the `set chassis cluster control-interface <node0/node1> disable` on node 0 or node 1.

If you disable the links using the configuration command, the links remain disabled even after system reboot.

- To enable the control link, run the `delete chassis cluster control-interface <node0/node1> disable` on both nodes.

- **In operational mode**

- To disable the control link from the local node, run the `request chassis cluster control-interface <node0/node1> disable` command.

If you disable the control link using the operational mode CLI command, the link will be enabled after system reboot.

- To enable the control link on a local node, run the `request chassis cluster control-interface <node0/node1> enable` command.

## Benefit of Dual Control Links

Dual control links prevent the possibility of a single point of failure by providing a redundant link for control traffic.

## Dual Control Link Functionality Requirements

For the SRX5600 and SRX5800 Services Gateways, dual control link functionality requires that a second Routing Engine and a second Switch Control Board (SCB) be installed on each device in the cluster. The purpose of the second Routing Engine is to initialize the switch on the primary SCB. The second SCB houses the second Routing Engine.

For the SRX5000 Services Gateways only, the second Routing Engine must be running Junos OS Release 10.0 or later.

This second Routing Engine does not provide backup functionality. It does not need to be upgraded, even when you upgrade the software on the primary Routing Engine on the same node. Note the following conditions:

- You can run CLI commands and enter configuration mode only on the primary Routing Engine.
- You set the chassis ID and cluster ID only on the primary Routing Engine.
- If you want to be able to check that the second Routing Engine boots up, or if you want to upgrade a software image, you need a console connection to the second Routing Engine.

As long as the first Routing Engine is installed (even if it reboots or fails), the second Routing Engine cannot take over the chassis primary role; that is, it cannot control any of the hardware on the chassis.

A redundancy group 0 failover implies a Routing Engine failover. In the case of a Routing Engine failover, all processes running on the primary node are killed and then spawned on the new primary Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

For SRX3000 Services Gateways, dual control link functionality requires that an SRX Clustering Module (SCM) be installed on each device in the cluster. Although the SCM fits in the Routing Engine slot, it is not a Routing Engine. The SRX3000 devices do not support a second Routing Engine. The purpose of the SCM is only to initialize the second control link.

## SEE ALSO

[Chassis Cluster Control Plane Interfaces | 72](#)

## Dual Control Link Connections for SRX Series Firewalls in a Chassis Cluster

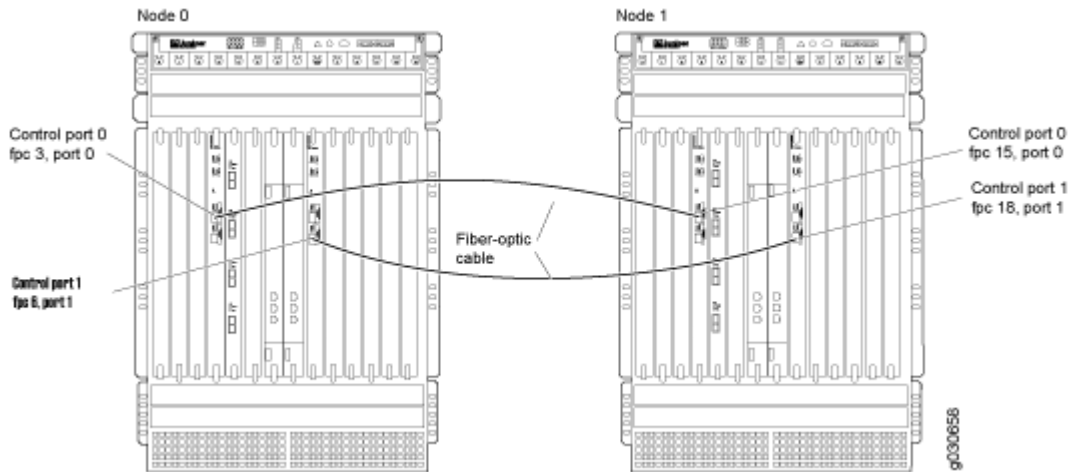
You can connect two control links between SRX5600 devices and SRX5800 devices, effectively reducing the chance of control link failure.

**NOTE:** Junos OS does not support dual control links on SRX5400 devices, due to the limited number of slots.

For SRX5600 devices and SRX5800 devices, connect two pairs of the same type of Ethernet ports. For each device, you can use ports on the same Services Processing Card (SPC), but we recommend that you connect the control ports to two different SPCs to provide high availability. [Figure 35 on page 166](#)

shows a pair of SRX5800 devices with dual control links connected. In this example, control port 0 and control port 1 are connected on different SPCs.

**Figure 35: Dual Control Link Connections (SRX5800 Devices)**



For SRX5600 devices and SRX5800 devices, you must connect control port 0 on one node to control port 0 on the other node. You must also connect control port 1 on one node to control port 1 on the other node. If you connect control port 0 to control port 1, the nodes cannot receive heartbeat packets across the control links.

## SEE ALSO

[Connecting SRX Series Firewalls to Create a Chassis Cluster | 39](#)

[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming | 19](#)

*request chassis fpc-control-port*

*request chassis primary-ha-control-port-transition*

## Upgrade the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices

You must use a second Routing Engine for each SRX5600 device and SRX5800 device in a cluster if you are using dual control links. The second Routing Engine does not provide backup functionality; its purpose is only to initialize the switch on the Switch Control Board (SCB). The second Routing Engine

must be running Junos OS Release 12.1X47-D35, 12.3X48-D30, 15.1X49-D40, or later. For more information, see knowledge base article [KB30371](#).

On SRX5600 devices and SRX5800 devices, you can use the `show chassis hardware` command to see the serial number and the hardware version details of the second Routing Engine. To use this functionality, ensure that the second Routing Engine is running either Junos OS Release 15.1X49-D70 or Junos OS Release 17.3R1.

Junos OS does not support dual control link functionality on the SRX5400 Services Gateways, due to limited slots.

Instead, use the primary Routing Engine to create a bootable USB storage device, which you can then use to install a software image on the second Routing Engine.

To upgrade the software image on the second Routing Engine:

1. Use FTP to copy the installation media into the `/var/tmp` directory of the primary Routing Engine.
2. Insert a USB storage device into the USB port on the primary Routing Engine.
3. In the UNIX shell, navigate to the `/var/tmp` directory:

```
start shell
cd /var/tmp
```

4. Log in as root or superuser:

```
su [enter]
password: [enter SU password]
```

5. Issue the following command:

```
dd if=installMedia of=/dev/externalDrive bs=1m
```

where

- *externalDrive*—Refers to the removable media name. For example, the removable media name on an SRX5000 line device is `da0` for both Routing Engines.
- *installMedia*—Refers to the installation media downloaded into the `/var/tmp` directory. For example, `junos-install-media-usb-srx5000-x86-64-21.4R1.7.img.gz`.

Copy the install-media image to the primary Routing Engine in step 1 onto the USB storage device:

```
dd if=junos-install-media-usb-srx5000-x86-64-21.4R1.7.img.gz of=/dev/da0 bs=1m
```

6. Log out as root or superuser:

```
exit
```

7. After the software image is written to the USB storage device, remove the device and insert it into the USB port on the second Routing Engine.
8. Move the console connection from the primary Routing Engine to the second Routing Engine, if you do not already have a connection.
9. Reboot the second Routing Engine. Issue the following command (for Junos OS Release 15.1X49-D65 and earlier):

```
# reboot
```

Starting with Junos OS Release 15.1X49-D70, issue the following command:

```
login : root
```

```
root % reboot
```

- When the following system output appears, press y:

```
WARNING: The installation will erase the contents of your disks.
Do you wish to continue (y/n)?
```

- When the following system output appears, remove the USB storage device and press Enter:

```
Eject the installation media and hit [Enter] to reboot?
```

## Example: Configure Chassis Cluster Control Ports for Dual Control Links

### IN THIS SECTION

- [Requirements | 169](#)
- [Overview | 169](#)
- [Configuration | 170](#)
- [Verification | 171](#)

This example shows how to configure chassis cluster control ports for use as dual control links on SRX5600 devices and SRX5800 devices. You need to configure the control ports that you will use on each device to set up the control links.

Junos OS does not support dual control links on SRX5400 devices, due to the limited number of slots.

### Requirements

Before you begin:

- Understand chassis cluster control links. See "[Understanding Chassis Cluster Control Plane and Control Links](#)" on page 72.
- Physically connect the control ports on the devices. See "[Connecting SRX Series Devices to Create a Chassis Cluster](#)" on page 39.

### Overview

By default, all control ports on SRX5600 devices and SRX5800 devices are disabled. After connecting the control ports, configuring the control ports, and establishing the chassis cluster, the control links are set up.

This example configures control ports with the following FPCs and ports as the dual control links:

- FPC 4, port 0
- FPC 10, port 0
- FPC 6, port 1
- FPC 12, port 1

## Configuration

### IN THIS SECTION

- [Procedure | 170](#)

### Procedure

#### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
set chassis cluster control-ports fpc 4 port 0
set chassis cluster control-ports fpc 10 port 0
set chassis cluster control-ports fpc 6 port 1
set chassis cluster control-ports fpc 12 port 1
```

#### Step-by-Step Procedure

To configure control ports for use as dual control links for the chassis cluster:

Specify the control ports.

```
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 4 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 10 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 6 port 1
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 12 port 1
```



## Results

In configuration mode, confirm your configuration by entering the `show chassis cluster` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show chassis cluster
...
control-ports {
  fpc 4 port 0;
  fpc 6 port 1;
  fpc 10 port 0;
  fpc 12 port 1;
}
...
```

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

## Verification

### IN THIS SECTION

- [Verification of the Chassis Cluster Status | 171](#)

### Verification of the Chassis Cluster Status

#### Purpose

Verify the chassis cluster status.

## Action

In operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no

Redundancy group: 1 , Failover count: 1
  node0              0         primary  no       no
  node1              0         secondary no       no
```

## Meaning

Use the `show chassis cluster status` command to confirm that the devices in the chassis cluster are communicating with each other. The output shows that the chassis cluster is functioning properly, as one device is the primary node and the other is the secondary node.

## RELATED DOCUMENTATION

[Connecting SRX Series Firewalls to Create a Chassis Cluster | 39](#)

[SRX Series Chassis Cluster Configuration Overview | 14](#)

# Chassis Cluster Dual Fabric Links

## IN THIS SECTION

- [Understanding Chassis Cluster Dual Fabric Links | 173](#)
- [Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports | 174](#)

- [Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports | 177](#)

Dual fabric links remove single point of failure in a chassis cluster setup. If one fabric link fails and one remains functional, all sessions are maintained between the two nodes and the chassis cluster status is preserved. For more information, see the following topics:

## Understanding Chassis Cluster Dual Fabric Links

You can connect two fabric links between each device in a cluster, which provides a redundant fabric link between the members of a cluster. Having two fabric links helps to avoid a possible single point of failure.

When you use dual fabric links, the RTOs and probes are sent on one link and the fabric-forwarded and flow-forwarded packets are sent on the other link. If one fabric link fails, the other fabric link handles the RTOs and probes, as well as the data forwarding. The system selects the physical interface with the lowest slot, PIC, or port number on each node for the RTOs and probes.

For all SRX Series Firewalls, you can connect two fabric links between two devices, effectively reducing the chance of a fabric link failure.

In most SRX Series Firewalls in a *chassis cluster*, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes.

For dual fabric links, both of the child interface types should be the same type. For example, both should be Gigabit Ethernet interfaces or 10-Gigabit interfaces.

SRX300, SRX320, SRX340, and SRX345 devices support Gigabit Ethernet interfaces only.

SRX380 devices support any of Gigabit Ethernet and 10-Gigabit Ethernet interfaces.

### SEE ALSO

| [Understanding Chassis Cluster Fabric Interfaces | 60](#)

## Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports

### IN THIS SECTION

- [Requirements | 174](#)
- [Overview | 174](#)
- [Configuration | 175](#)
- [Verification | 176](#)

This example shows how to configure the chassis cluster fabric with dual fabric links with matching slots and ports. The fabric is the back-to-back data connection between the nodes in a cluster. Traffic on one node that needs to be processed on the other node or to exit through an interface on the other node passes over the fabric. Session state information also passes over the fabric.

### Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See "[Example: Setting the Chassis Cluster Node ID and Cluster ID](#)" on page 45.

### Overview

In most SRX Series Firewalls in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes.

You cannot configure filters, policies, or services on the fabric interface. Fragmentation is not supported on the fabric link. The MTU size is 8984 bytes. We recommend that no interface in the cluster exceed this MTU size. Jumbo frame support on the member links is enabled by default.

This example illustrates how to configure the fabric link with dual fabric links with matching slots and ports on each node.

A typical configuration is where the dual fabric links are formed with matching slots/ports on each node. That is, ge-3/0/0 on node 0 and ge-10/0/0 on node 1 match, as do ge-0/0/0 on node 0 and ge-7/0/0 on node 1 (the FPC slot offset is 7).

Only the same type of interfaces can be configured as fabric children, and you must configure an equal number of child links for fab0 and fab1.

If you are connecting each of the fabric links through a switch, you must enable the jumbo frame feature on the corresponding switch ports. If both of the fabric links are connected through the same switch, the RTO-and-probes pair must be in one virtual LAN (VLAN) and the data pair must be in another VLAN. Here, too, the jumbo frame feature must be enabled on the corresponding switch ports.

## Configuration

### IN THIS SECTION

- [Procedure | 175](#)

### 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, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
set interfaces fab0 fabric-options member-interfaces ge-0/0/0
set interfaces fab0 fabric-options member-interfaces ge-3/0/0
set interfaces fab1 fabric-options member-interfaces ge-7/0/0
set interfaces fab1 fabric-options member-interfaces ge-10/0/0
```

#### Step-by-Step Procedure

To configure the chassis cluster fabric with dual fabric links with matching slots and ports on each node:

- Specify the fabric interfaces.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/0
user@host# set interfaces fab0 fabric-options member-interfaces ge-3/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-10/0/0
```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
...
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/0;
      ge-3/0/0;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-7/0/0;
      ge-10/0/0;
    }
  }
}
```

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

## Verification

### IN THIS SECTION

- [Verifying the Chassis Cluster Fabric | 177](#)

## Verifying the Chassis Cluster Fabric

### Purpose

Verify the chassis cluster fabric.

### Action

From operational mode, enter the `show interfaces terse | match fab` command.

```
{primary:node0}
user@host> show interfaces terse | match fab
ge-0/0/0.0          up   up   aenet  --> fab0.0
ge-3/0/0.0          up   up   aenet  --> fab0.0
ge-7/0/0.0          up   up   aenet  --> fab1.0
ge-10/0/0.0         up   up   aenet  --> fab1.0
fab0                 up   up
fab0.0               up   up   inet   10.17.0.200/24
fab1                 up   up
fab1.0               up   up   inet   10.18.0.200/24
```

### SEE ALSO

[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming | 19](#)

## Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports

### IN THIS SECTION

- [Requirements | 178](#)
- [Overview | 178](#)
- [Configuration | 179](#)
- [Verification | 180](#)

This example shows how to configure the chassis cluster fabric with dual fabric links with different slots and ports. The fabric is the back-to-back data connection between the nodes in a cluster. Traffic on one node that needs to be processed on the other node or to exit through an interface on the other node passes over the fabric. Session state information also passes over the fabric.

## Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See ["Example: Setting the Chassis Cluster Node ID and Cluster ID" on page 45](#).

## Overview

In most SRX Series Firewalls in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes.

You cannot configure filters, policies, or services on the fabric interface. Fragmentation is not supported on the fabric link.

The maximum transmission unit (MTU) size supported is 8984 We recommend that no interface in the cluster exceed this MTU size. Jumbo frame support on the member links is enabled by default.

This example illustrates how to configure the fabric link with dual fabric links with different slots and ports on each node.

Make sure you physically connect the RTO-and-probes link to the RTO-and-probes link on the other node. Likewise, make sure you physically connect the data link to the data link on the other node.

That is, physically connect the following two pairs:

- The node 0 RTO-and-probes link ge-2/1/9 to the node 1 RTO-and-probes link ge-11/0/0
- The node 0 data link ge-2/2/5 to the node 1 data link ge-11/3/0

Only the same type of interfaces can be configured as fabric children, and you must configure an equal number of child links for fab0 and fab1.

If you are connecting each of the fabric links through a switch, you must enable the jumbo frame feature on the corresponding switch ports. If both of the fabric links are connected through the same switch, the RTO-and-probes pair must be in one virtual LAN (VLAN) and the data pair must be in another VLAN. Here too, the jumbo frame feature must be enabled on the corresponding switch ports.



## Configuration

### IN THIS SECTION

- Procedure | 179

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
set interfaces fab0 fabric-options member-interfaces ge-2/1/9
set interfaces fab0 fabric-options member-interfaces ge-2/2/5
set interfaces fab1 fabric-options member-interfaces ge-11/0/0
set interfaces fab1 fabric-options member-interfaces ge-11/3/0
```

#### Step-by-Step Procedure

To configure the chassis cluster fabric with dual fabric links with different slots and ports on each node:

- Specify the fabric interfaces.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-2/1/9
user@host# set interfaces fab0 fabric-options member-interfaces ge-2/2/5
user@host# set interfaces fab1 fabric-options member-interfaces ge-11/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-11/3/0
```

### Results

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

For brevity, this show command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show interfaces
...
fab0 {
  fabric-options {
    member-interfaces {
      ge-2/1/9;
      ge-2/2/5;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-11/0/0;
      ge-11/3/0;
    }
  }
}
```

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

## Verification

### IN THIS SECTION

- [Verifying the Chassis Cluster Fabric | 180](#)

## Verifying the Chassis Cluster Fabric

### Purpose

Verify the chassis cluster fabric.

## Action

From operational mode, enter the `show interfaces terse | match fab` command.

```
{primary:node0}
user@host> show interfaces terse | match fab
ge-2/1/9.0          up   up   aenet  --> fab0.0
ge-2/2/5.0          up   up   aenet  --> fab0.0
ge-11/0/0.0         up   up   aenet  --> fab1.0
ge-11/3/0.0         up   up   aenet  --> fab1.0
fab0                 up   up
fab0.0               up   up   inet   30.17.0.200/24
fab1                 up   up
fab1.0               up   up   inet   30.18.0.200/24
```

## RELATED DOCUMENTATION

[Connecting SRX Series Firewalls to Create a Chassis Cluster | 39](#)

[SRX Series Chassis Cluster Configuration Overview | 14](#)

# Monitoring of Global-Level Objects in a Chassis Cluster

## IN THIS SECTION

- [Understanding SPU Monitoring | 182](#)
- [Understanding flowd Monitoring | 183](#)
- [Understanding Cold-Sync Monitoring | 183](#)

There are various types of objects to monitor as you work with devices configured as chassis clusters, including global-level objects and objects that are specific to redundancy groups. This section describes the monitoring of global-level objects.

The SRX5000 lines have one or more Services Processing Units (SPUs) that run on a Services Processing Card (SPC). All flow-based services run on the SPU. Other SRX Series Firewalls have a flow-based forwarding process, *flowd*, which forwards packets through the device.

## Understanding SPU Monitoring

SPU monitoring tracks the health of the SPUs and of the central point (CP). The chassis manager on each SPC monitors the SPUs and the central point, and also maintains the heartbeat with the Routing Engine chassisd. In this hierarchical monitoring system, chassisd is the center for hardware failure detection. SPU monitoring is enabled by default.

SPU monitoring is supported on SRX4600 and SRX5000 line devices.

Persistent SPU and central point failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups  $x$  to 0.

- A central point failure triggers failover to the secondary node. The failed node's PFE, which includes all SPCs and all I/O cards (IOCs), is automatically restarted. If the secondary central point has failed as well, the cluster is unable to come up because there is no primary device. Only the data plane (redundancy group  $x$ ) is failed over.
- A single, failed SPU causes failover of redundancy group  $x$  to the secondary node. All IOCs and SPCs on the failed node are restarted and redundancy group  $x$  is failed over to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group  $x$ . The interval for dead SPU detection is 30 seconds.

On SRX5400, SRX5600, and SRX5800 SPCs, the Routing Engine monitors the health of the chassis manager. The chassis manager sends a heartbeat message to the Routing Engine chassisd every second. When the Routing Engine chassisd detects a heartbeat loss, it initiates a power cycle for the entire SPC. If multiple recoveries fail within a certain timeframe, the Routing Engine powers off the SPC to prevent it from affecting the entire system.

This event triggers an alarm, indicating that a new field-replaceable unit (FRU) is needed.

The following list describes the limitations for inserting an SPC on SRX5400, SRX5600, and SRX5800 devices in chassis cluster mode:

- The chassis cluster must be in active/passive mode before and during the SPC insert procedure.
- A different number of SPCs cannot be inserted in two different nodes.
- A new SPC must be inserted in a slot that is higher than the central point slot.

The existing combo central point cannot be changed to a full central point after the new SPC is inserted.

- During an SPC insert procedure, the IKE and IPsec configurations cannot be modified.

An SPC is not hot-insertable. Before inserting an SPC, the device must be taken offline. After inserting an SPC, the device must be rebooted.

- Users cannot specify the SPU and the IKE instance to anchor a tunnel.
- After a new SPC is inserted, the existing tunnels cannot use the processing power of the new SPC and redistribute it to the new SPC.

## Understanding flowd Monitoring

Flowd monitoring tracks the health of the flowd process. Flowd monitoring is enabled by default.

Persistent flowd failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups  $x$  to 0.

A failed flowd process causes failover of redundancy group  $x$  to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group  $x$ .

During SPC and flowd monitoring failures on a local node, the data plane redundancy group RG1+ fails over to the other node that is in a good state. However, the control plane RGO does not fail over and remains primary on the same node as it was before the failure.

## Understanding Cold-Sync Monitoring

The process of synchronizing the data plane runtime objects (RTOs) on the startup of the SPUs or flowd is called *cold sync*. When all the RTOs are synchronized, the cold-sync process is complete, and the SPU or flowd on the node is ready to take over for the primary node, if needed. The process of monitoring the cold-sync state of all the SPUs or flowd on a node is called *cold-sync monitoring*. Keep in mind that when preempt is enabled, cold-sync monitoring prevents the node from taking over the primary role until the cold-sync process is completed for the SPUs or flowd on the node. Cold-sync monitoring is enabled by default.

When the node is rebooted, or when the SPUs or flowd come back up from failure, the priority for all the redundancy groups *1+* is 0. When an SPU or flowd comes up, it tries to start the cold-sync process with its mirror SPU or flowd on the other node.

If this is the only node in the cluster, the priorities for all the redundancy groups *1+* stay at 0 until a new node joins the cluster. Although the priority is at 0, the device can still receive and send traffic over its interfaces. A priority of 0 implies that it cannot fail over in case of a failure. When a new node joins the cluster, all the SPUs or flowd, as they come up, will start the cold-sync process with the mirror SPUs or flowd of the existing node.

When the SPU or flowd of a node that is already up detects the cold-sync request from the SPU or flowd of the peer node, it posts a message to the system indicating that the cold-sync process is complete. The SPUs or flowd of the newly joined node posts a similar message. However, they post this message only after all the RTOs are learned and cold-sync is complete. On receipt of completion messages from all the SPUs or flowd, the priority for redundancy groups *1+* moves to the configured priority on each node if there are no other failures of monitored components, such as interfaces. This action ensures that the existing primary node for redundancy *1+* groups always moves to the configured priority first. The node joining the cluster later moves to its configured priorities only after all its SPUs or flowd have completed their cold-sync process. This action in turn guarantees that the newly added node is ready with all the RTOs before it takes over the primary role.

## Understanding Cold-Sync Monitoring with SPU Replacement or Expansion

If your SRX5600 or SRX5800 Firewall is part of a *chassis cluster*, when you replace a Services Processing Card (SPC) with a SPC2 or an SPC3 on the device, you must fail over all redundancy groups to one node.

For SRX5400 devices, SPC2 and SPC3 are supported.

The following events take place during this scenario:

- When the SPC2 is installed on a node (for example, on node 1, the secondary node), node 1 is shut down so the SPC2 can be installed.
- Once node 1 is powered up and rejoins the cluster, the number of SPUs on node 1 will be higher than the number of SPUs on node 0, the primary node. Now, one node (node 0) still has an old SPC while the other node has the new SPC2; SPC2s have four SPUs per card, and the older SPCs have two SPUs per card.

The cold-sync process is based on node 0 total SPU number. Once those SPUs in node 1 corresponding to node 0 SPUs have completed the cold-sync, the node 1 will declare cold-sync completed. Since the additional SPUs in node 1 do not have the corresponding node 0 SPUs, there is nothing to be synchronized and failover from node 0 to node 1 does not cause any issue.

SPU monitoring functionality monitors all SPUs and reports if there are any SPU failure.

For example assume that both nodes originally have 2 existing SPCs and you have replaced both SPCs with SPC2 on node 1. Now we have 4 SPUs in node 0 and 8 SPUs in node 1. The SPU monitoring function monitors the 4 SPUs on node 0 and 8 SPUs on node 1. If any of those 8 SPUs failed in node 1, the SPU monitoring will still report to the Juniper Services Redundancy Protocol (jsrpd) process that there is an SPU failure. The jsrpd process controls chassis clustering.

- Once node 1 is ready to failover, you can initiate all redundancy group failover manually to node 1. Node 0 will be shut down to replace its SPC with the SPC2. After the replacement, node 0 and node 1 will have exactly the same hardware setup.

Once node 0 is powered up and rejoins the cluster, the system will operate as a normal chassis cluster.

Starting from Junos OS Release 15.1X49-D120, when the cold-sync process is still in progress on SRX Series Firewall in chassis cluster, and if the control link is down, a delay (of 30 seconds) is expected before the node takes transition from the secondary state to the primary state.

## RELATED DOCUMENTATION

[Understanding Chassis Cluster Redundancy Group Interface Monitoring | 186](#)

[Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring | 187](#)

[Understanding Chassis Cluster Redundancy Group IP Address Monitoring | 231](#)

[Example: Configure Chassis Cluster Redundancy Group IP Address Monitoring | 233](#)

# Monitoring Chassis Cluster Interfaces

## IN THIS SECTION

- [Understanding Chassis Cluster Redundancy Group Interface Monitoring | 186](#)
- [Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring | 187](#)

Interface monitoring monitors the state of an interface by checking if the interface is in an up or down state. When one or more monitored interfaces fail, the redundancy group fails over to the other node in the cluster. For more information, see the following topics:

## Understanding Chassis Cluster Redundancy Group Interface Monitoring

### IN THIS SECTION

- [Benefits of Monitoring Chassis Cluster Redundancy Group Interfaces | 187](#)

For a redundancy group to automatically failover to another node, its interfaces must be monitored. When you configure a redundancy group, you can specify a set of interfaces that the redundancy group is to monitor for status (or “health”) to determine whether the interface is up or down. A monitored interface can be a child interface of any of its redundant Ethernet interfaces. When you configure an interface for a redundancy group to monitor, you give it a weight.

Every redundancy group has a threshold tolerance value initially set to 255. When an interface monitored by a redundancy group becomes unavailable, its weight is subtracted from the redundancy group's threshold. When a redundancy group's threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

To check the interface weight, use the following commands:

- `show chassis cluster information`
- `show chassis cluster interfaces`

We do not recommend configuring data plane modules such as interface monitoring and IP monitoring on redundancy group 0 (RG0) for SRX Series Firewalls in a chassis cluster.

Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine (RE) failover, in which case all processes running on the primary node are killed and then spawned on the new primary Routing Engine (RE). This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

A redundancy group failover occurs because the cumulative weight of the redundancy group's monitored interfaces has brought its threshold value to 0. When the monitored interfaces of a redundancy group on both nodes reach their thresholds at the same time, the redundancy group is primary on the node with the lower node ID, in this case node 0.

- If you want to dampen the failovers occurring because of interface monitoring failures, use the `hold-down-interval` statement.



- If a failover occurs on redundancy group 0 (RGO), the interface monitoring on the RGO secondary is disabled for 30 seconds. This prevents failover of other redundancy groups along with RGO failover.

## Benefits of Monitoring Chassis Cluster Redundancy Group Interfaces

- Helps to determine the status of a specific interface in a chassis cluster setup by a specific redundancy group.
- Enables automatic failover of an interface to another node if the interface is down.

### SEE ALSO

| [Understanding Chassis Cluster Redundancy Groups](#) | 83

## Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring

### IN THIS SECTION

- [Requirements](#) | 187
- [Overview](#) | 188
- [Configuration](#) | 189
- [Verification](#) | 194

This example shows how to specify that an interface be monitored by a specific redundancy group for automatic failover to another node. You assign a weight to the interface to be monitored also shows how to verify the process of the remaining threshold of a monitoring interface by configuring two interfaces from each node and mapping them to redundancy groups.

### Requirements

Before you begin, create a redundancy group. See "[Example: Configuring Chassis Cluster Redundancy Groups](#)" on page 87.

## Overview

### IN THIS SECTION

- [Topology | 189](#)

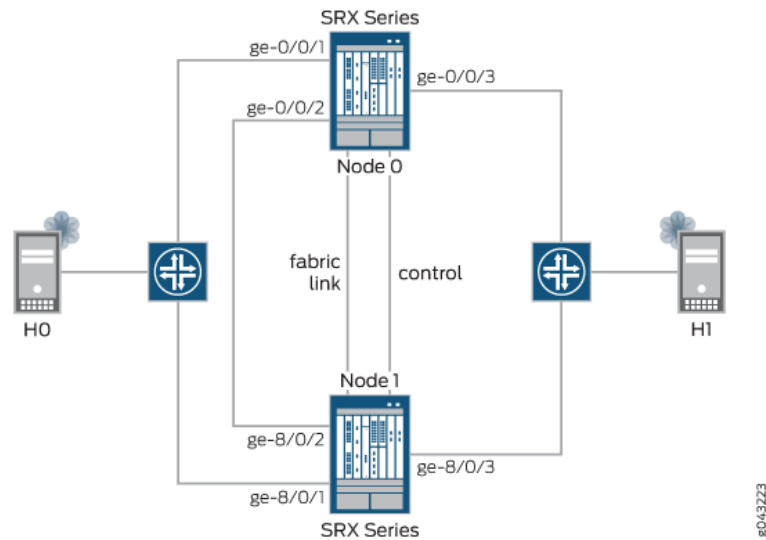
To retrieve the remaining redundancy group threshold after a monitoring interface is down, you can configure your system to monitor the health of the interfaces belonging to a redundancy group. When you assign a weight to an interface to be monitored, the system monitors the interface for availability. If a physical interface fails, the weight is deducted from the corresponding redundancy group's threshold. Every redundancy group has a threshold of 255. If the threshold hits 0, a failover is triggered, even if the redundancy group is in manual failover mode and the `preempt` option is not enabled.

In this example, you check the process of the remaining threshold of a monitoring interface by configuring two interfaces from each node and mapping them to redundancy group 1 (RG1), each with different weights. You use 130 and 140 for node 0 interfaces and 150 and 120 for node 1 interfaces. You configure one interface from each node and map the interfaces to redundancy group 2 (RG2), each with default weight of 255.

[Figure 36 on page 189](#) illustrates the network topology used in this example.

## Topology

Figure 36: SRX Series Chassis Cluster Interface Monitoring Topology Example



## Configuration

### IN THIS SECTION

- CLI Quick Configuration | 189
- Procedure | 190

### 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.

```
set chassis cluster reth-count 3
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
```

```

set chassis cluster redundancy-group 1 interface-monitor ge-0/0/1 weight 130
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 140
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/1 weight 150
set chassis cluster redundancy-group 1 interface-monitor ge-8/0/2 weight 120
set chassis cluster redundancy-group 2 node 0 priority 200
set chassis cluster redundancy-group 2 node 1 priority 100
set chassis cluster redundancy-group 2 interface-monitor ge-0/0/3 weight 255
set chassis cluster redundancy-group 2 interface-monitor ge-8/0/3 weight 255
set interfaces ge-0/0/1 gigether-options redundant-parent reth0
set interfaces ge-0/0/2 gigether-options redundant-parent reth1
set interfaces ge-0/0/3 gigether-options redundant-parent reth2
set interfaces ge-8/0/1 gigether-options redundant-parent reth0
set interfaces ge-8/0/2 gigether-options redundant-parent reth1
set interfaces ge-8/0/3 gigether-options redundant-parent reth2
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.2.2/24
set interfaces reth2 redundant-ether-options redundancy-group 2
set interfaces reth2 unit 0 family inet address 10.3.3.3/24

```

## Procedure

### 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 Junos OS CLI User Guide](#).

To configure chassis cluster interface monitoring:

1. Specify the number of redundant Ethernet interfaces.

```

[edit chassis cluster]
user@host# set reth-count 3

```

2. Set up redundancy group 0 for the Routing Engine failover properties, and set up RG1 and RG2 (all interfaces are in one redundancy group in this example) to define the failover properties for the redundant Ethernet interfaces.

```
[edit chassis cluster]
user@host# set redundancy-group 0 node 0 priority 254
user@host# set redundancy-group 0 node 1 priority 1
user@host# set redundancy-group 1 node 0 priority 200
user@host# set redundancy-group 1 node 1 priority 100
user@host# set redundancy-group 2 node 0 priority 200
user@host# set redundancy-group 2 node 1 priority 100
```

3. Set up interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.

We do not recommend interface monitoring for RGO, because it causes the control plane to switch from one node to another node in case interface flap occurs.

```
[edit chassis cluster]
user@host# set redundancy-group 1 interface-monitor ge-0/0/1 weight 130
user@host# set redundancy-group 1 interface-monitor ge-0/0/2 weight 140
user@host# set redundancy-group 1 interface-monitor ge-8/0/1 weight 150
user@host# set redundancy-group 1 interface-monitor ge-0/0/2 weight 120
user@host# set redundancy-group 2 interface-monitor ge-0/0/3 weight 255
user@host# set redundancy-group 2 interface-monitor ge-8/0/3 weight 255
```

Interface failover only occurs after the weight reaches zero.

4. Set up the redundant Ethernet (reth) interfaces and assign them to a zone.

```
[edit interfaces]
user@host# set ge-0/0/1 gigether-options redundant-parent reth0
user@host# set ge-0/0/2 gigether-options redundant-parent reth1
user@host# set ge-0/0/3 gigether-options redundant-parent reth2
user@host# set ge-8/0/1 gigether-options redundant-parent reth0
user@host# set ge-8/0/2 gigether-options redundant-parent reth1
user@host# set ge-8/0/3 gigether-options redundant-parent reth2
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 unit 0 family inet address 10.1.1.1/24
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth1 unit 0 family inet address 10.2.2.2/24
```

```
user@host# set reth2 redundant-ether-options redundancy-group 2
user@host# set reth2 unit 0 family inet address 10.3.3.3/24
```

## Results

From configuration mode, confirm your configuration by entering the `show chassis` and `show interfaces` 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
cluster {
  reth-count 3;
  redundancy-group 0 {
    node 0 priority 254;
    node 1 priority 1;
  }
  redundancy-group 1 {
    node 0 priority 200;
    node 1 priority 100;
    interface-monitor {
      ge-0/0/1 weight 130;
      ge-0/0/2 weight 140;
      ge-8/0/1 weight 150;
      ge-8/0/2 weight 120;
    }
  }
  redundancy-group 2 {
    node 0 priority 200;
    node 1 priority 100;
    interface-monitor {
      ge-0/0/3 weight 255;
      ge-8/0/3 weight 255;
    }
  }
}
[edit]
user@host# show interfaces
ge-0/0/1 {
  ggether-options {
    redundant-parent reth0;
```

```
    }  
  }  
  ge-0/0/2 {  
    gigether-options {  
      redundant-parent reth1;  
    }  
  }  
  ge-0/0/3 {  
    gigether-options {  
      redundant-parent reth2;  
    }  
  }  
  ge-8/0/1 {  
    gigether-options {  
      redundant-parent reth0;  
    }  
  }  
  ge-8/0/2 {  
    gigether-options {  
      redundant-parent reth1;  
    }  
  }  
  ge-8/0/3 {  
    gigether-options {  
      redundant-parent reth2;  
    }  
  }  
  reth0 {  
    redundant-ether-options {  
      redundancy-group 1;  
    }  
    unit 0 {  
      family inet {  
        address 10.1.1.1/24;  
      }  
    }  
  }  
  reth1 {  
    redundant-ether-options {  
      redundancy-group 1;  
    }  
    unit 0 {  
      family inet {
```

```

        address 10.2.2.2/24;
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 2;
    }
    unit 0 {
        family inet {
            address 10.3.3.3/24;
        }
    }
}

```

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

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Status | 195](#)
- [Verifying Chassis Cluster Interfaces | 196](#)
- [Verifying Chassis Cluster Information | 198](#)
- [Verifying Interface ge-0/0/1 Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 | 200](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 | 200](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 | 202](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 | 203](#)
- [Verifying Interface ge-0/0/2 Is Disabled | 205](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/2 | 206](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/2 | 207](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/2 | 208](#)
- [Verifying Interface Status After Disabling ge-0/0/3 | 210](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/3 | 211](#)



- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/3 | 212](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/3 | 213](#)
- [Verifying That Interface ge-0/0/2 Is Enabled | 215](#)
- [Verifying Chassis Cluster Status After Enabling Interface ge-0/0/2 | 216](#)
- [Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/2 | 217](#)
- [Verifying Chassis Cluster Information After Enabling Interface ge-0/0/2 | 218](#)
- [Verifying Chassis Cluster RG2 Preempt | 220](#)
- [Verifying Chassis Cluster Status After Preempting RG2 | 221](#)
- [Verifying That Interface ge-0/0/3 Is Enabled | 222](#)
- [Verifying Chassis Cluster Status After Enabling Interface ge-0/0/3 | 223](#)
- [Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/3 | 224](#)
- [Verifying Chassis Cluster Information After Enabling Interface ge-0/0/3 | 225](#)

The following sections walk you through the process of verifying and (in some cases) troubleshooting the interface status. The process shows you how to check the status of each interface in the redundancy group, check them again after they have been disabled, and looks for details about each interface, until you have circled through all interfaces in the redundancy group.

In this example, you verify the process of the remaining threshold of a monitoring interface by configuring two interfaces from each node and mapping them to RG1, each with different weights. You use 130 and 140 for node 0 interfaces and 150 and 120 for node 1 interfaces. You configure one interface from each node and map the interfaces to RG2, each with the default weight of 255.

## Verifying Chassis Cluster Status

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
```

```

CS Cold Sync monitoring      FL Fabric Connection monitoring
GR GRES monitoring          HW Hardware monitoring
IF Interface monitoring      IP IP monitoring
LB Loopback monitoring      MB Mbuf monitoring
NH Nexthop monitoring       NP NPC monitoring
SP SPU monitoring           SM Schedule monitoring
CF Config Sync monitoring

```

```
Cluster ID: 2
```

```
Node  Priority Status          Preempt Manual  Monitor-failures
```

```
Redundancy group: 0 , Failover count: 1
```

```
node0 254    primary    no    no    None
node1  1     secondary  no    no    None
```

```
Redundancy group: 1 , Failover count: 1
```

```
node0 200    primary    no    no    None
node1 100    secondary  no    no    None
```

```
Redundancy group: 2 , Failover count: 1
```

```
node0 200    primary    no    no    None
node1 100    secondary  no    no    None
```

## Meaning

Use the `show chassis cluster status` command to confirm that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the secondary node.

## Verifying Chassis Cluster Interfaces

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

## Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
  ----  -
  0      em0        Up                 Disabled
  1      em1        Down                Disabled

Fabric link status: Up

Fabric interfaces:
  Name      Child-interface  Status
              (Physical/Monitored)
  ----      -
  fab0      ge-0/0/0         Up / Up
  fab0
  fab1      ge-8/0/0         Up / Up
  fab1

Redundant-ethernet Information:
  Name      Status  Redundancy-group
  ----      -
  reth0     Up      1
  reth1     Up      1
  reth2     Up      2

Redundant-pseudo-interface Information:
  Name      Status  Redundancy-group
  ----      -
  lo0       Up      0

Interface Monitoring:
  Interface  Weight  Status  Redundancy-group
  ----
  ge-8/0/2   120    Up      1
  ge-8/0/1   150    Up      1
  ge-0/0/2   140    Up      1
  ge-0/0/1   130    Up      1
  ge-8/0/3   255    Up      2
  ge-0/0/3   255    Up      2
```

## Meaning

The sample output confirms that monitoring interfaces are up and that the weight of each interface being monitored is displayed correctly as configured. These values do not change if the interface goes up or down. The weights only change for the redundant group and can be viewed when you use the `show chassis cluster information` command.

## Verifying Chassis Cluster Information

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster information` command.

```
{primary:node0}
user@host> show chassis cluster information

node0:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: primary, Weight: 255

Time           From           To           Reason
Feb 24 22:56:27 hold           secondary    Hold timer expired
Feb 24 22:56:34 secondary      primary      Better priority (254/1)

Redundancy Group 1 , Current State: primary, Weight: 255

Time           From           To           Reason
Feb 24 23:16:12 hold           secondary    Hold timer expired
Feb 24 23:16:12 secondary      primary      Remote yield (0/0)

Redundancy Group 2 , Current State: primary, Weight: 255

Time           From           To           Reason
Feb 24 23:16:12 hold           secondary    Hold timer expired
```

```

Feb 24 23:16:13 secondary      primary      Remote yield (0/0)

Chassis cluster LED information:
  Current LED color: Green
  Last LED change reason: No failures

node1:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

  Time          From          To          Reason
  Feb 24 22:56:34 hold          secondary   Hold timer expired

Redundancy Group 1 , Current State: secondary, Weight: 255

  Time          From          To          Reason
  Feb 24 23:16:10 hold          secondary   Hold timer expired

Redundancy Group 2 , Current State: secondary, Weight: 255

  Time          From          To          Reason
  Feb 24 23:16:10 hold          secondary   Hold timer expired

Chassis cluster LED information:
  Current LED color: Green
  Last LED change reason: No failures

```

## Meaning

The sample output confirms that node 0 and node 1 are healthy, and the green LED on the device indicates that there are no failures. Also, the default weight of the redundancy group (255) is displayed. The default weight is deducted whenever an interface mapped to the corresponding redundancy group goes down.

Refer to subsequent verification sections to see how the redundancy group value varies when a monitoring interface goes down or comes up.

## Verifying Interface ge-0/0/1 Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130

### Purpose

Verify that the interface ge-0/0/1 is disabled on node 0.

### Action

From configuration mode, enter the `set interface ge-0/0/1 disable` command.

```
{primary:node0}
user@host# set interface ge-0/0/1 disable
user@host# commit

node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

{primary:node0}
user@host# show interfaces ge-0/0/1
disable;
gigether-options {
    redundant-parent reth0;
}
```

### Meaning

The sample output confirms that interface ge-0/0/1 is disabled.

## Verifying Chassis Cluster Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

## Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary no no None
node1 1 secondary no no None

Redundancy group: 1 , Failover count: 1
node0 200 primary no no None
node1 100 secondary no no None

Redundancy group: 2 , Failover count: 1
node0 200 primary no no None
node1 100 secondary no no None
```

## Meaning

Use the `show chassis cluster status` command to confirm that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the secondary node.

## Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
  ----  -
  0      em0        Up                 Disabled
  1      em1        Down               Disabled

Fabric link status: Up

Fabric interfaces:
  Name      Child-interface  Status
              (Physical/Monitored)
  ----      -
  fab0      ge-0/0/0        Up / Up
  fab0
  fab1      ge-8/0/0        Up / Up
  fab1

Redundant-ethernet Information:
  Name      Status  Redundancy-group
  ----      -
  reth0     Down   1
  reth1     Up     1
  reth2     Up     2

Redundant-pseudo-interface Information:
  Name      Status  Redundancy-group
  ----      -
  lo0       Up     0

Interface Monitoring:
```



Interface	Weight	Status	Redundancy-group
ge-8/0/2	120	Up	1
ge-8/0/1	150	Up	1
ge-0/0/2	140	Up	1
ge-0/0/1	130	Down	1
ge-8/0/3	255	Up	2
ge-0/0/3	255	Up	2

## Meaning

The sample output confirms that monitoring interface ge-0/0/1 is down.

## Verifying Chassis Cluster Information After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster information` command.

```
{primary:node0}
user@host> show chassis cluster information

node0:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: primary, Weight: 255

Time          From          To          Reason
Feb 24 22:56:27 hold          secondary   Hold timer expired
Feb 24 22:56:34 secondary    primary     Better priority (254/1)

Redundancy Group 1 , Current State: primary, Weight: 125

Time          From          To          Reason
Feb 24 23:16:12 hold          secondary   Hold timer expired
```

Feb 24 23:16:12 secondary primary Remote yield (0/0)

Redundancy Group 2 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:13	secondary	primary	Remote yield (0/0)

Chassis cluster LED information:

Current LED color: Green

Last LED change reason: No failures

Failure Information:

Interface Monitoring Failure Information:

Redundancy Group 1, Monitoring status: Unhealthy

Interface	Status
ge-0/0/1	Down

node1:

-----  
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:34	hold	secondary	Hold timer expired

Redundancy Group 1 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired

Redundancy Group 2 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

## Meaning

The sample output confirms that in node 0, the RG1 weight is reduced to 125 (that is, 255 minus 130) because monitoring interface ge-0/0/1 (weight of 130) went down. The monitoring status is unhealthy, the device LED is amber, and the interface status of ge-0/0/1 is down.

If interface ge-0/0/1 is brought back up, the weight of RG1 in node 0 becomes 255. Conversely, if interface ge-0/0/2 is also disabled, the weight of RG1 in node 0 becomes 0 or less (in this example, 125 minus 140 = -15) and triggers failover, as indicated in the next verification section.

## Verifying Interface ge-0/0/2 Is Disabled

### Purpose

Verify that interface ge-0/0/2 is disabled on node 0.

### Action

From configuration mode, enter the `set interface ge-0/0/2 disable` command.

```
{primary:node0}
user@host# set interface ge-0/0/2 disable
user@host# commit

node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

{primary:node0}
user@host# show interfaces ge-0/0/2
disable;
gigether-options {
  redundant-parent reth1;
}
```

## Meaning

The sample output confirms that interface ge-0/0/2 is disabled.

## Verifying Chassis Cluster Status After Disabling Interface ge-0/0/2

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary no no None
node1 1 secondary no no None

Redundancy group: 1 , Failover count: 2
node0 0 secondary no no IF
node1 100 primary no no None

Redundancy group: 2 , Failover count: 1
node0 200 primary no no None
node1 100 secondary no no None
```

### Meaning

Use the `show chassis cluster status` command to confirm that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the

secondary node. On RG1, you see interface failure, because both interfaces mapped to RG1 on node 0 failed during interface monitoring.

## Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/2

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
  0      em0        Up                 Disabled
  1      em1        Down               Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
          (Physical/Monitored)
  fab0    ge-0/0/0         Up / Up
  fab0
  fab1    ge-8/0/0         Up / Up
  fab1

Redundant-ethernet Information:
  Name    Status  Redundancy-group
  reth0   Up      1
  reth1   Up      1
  reth2   Up      2

Redundant-pseudo-interface Information:
  Name    Status  Redundancy-group
  lo0     Up      0
```

**Interface Monitoring:**

Interface	Weight	Status	Redundancy-group
ge-8/0/2	120	Up	1
ge-8/0/1	150	Up	1
ge-0/0/2	140	Down	1
ge-0/0/1	130	Down	1
ge-8/0/3	255	Up	2
ge-0/0/3	255	Up	2

**Meaning**

The sample output confirms that monitoring interfaces ge-0/0/1 and ge-0/0/2 are down.

**Verifying Chassis Cluster Information After Disabling Interface ge-0/0/2****Purpose**

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

**Action**

From operational mode, enter the `show chassis cluster information` command.

```
{primary:node0}
user@host> show chassis cluster information

node0:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: primary, Weight: 255

Time          From          To          Reason
Feb 24 22:56:27 hold          secondary   Hold timer expired
Feb 24 22:56:34 secondary    primary     Better priority (254/1)

Redundancy Group 1 , Current State: secondary, Weight: -15

Time          From          To          Reason
Feb 24 23:16:12 hold          secondary   Hold timer expired
```

```

Feb 24 23:16:12 secondary    primary    Remote yield (0/0)
Feb 24 23:31:36 primary      secondary-hold Monitor failed: IF
Feb 24 23:31:37 secondary-hold secondary    Ready to become secondary

```

Redundancy Group 2 , Current State: primary, Weight: 255

```

Time          From          To          Reason
Feb 24 23:16:12 hold          secondary    Hold timer expired
Feb 24 23:16:13 secondary    primary      Remote yield (0/0)

```

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

Failure Information:

Interface Monitoring Failure Information:

Redundancy Group 1, Monitoring status: Failed

```

Interface          Status
ge-0/0/2           Down
ge-0/0/1           Down

```

node1:

-----  
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

```

Time          From          To          Reason
Feb 24 22:56:34 hold          secondary    Hold timer expired

```

Redundancy Group 1 , Current State: primary, Weight: 255

```

Time          From          To          Reason
Feb 24 23:16:10 hold          secondary    Hold timer expired
Feb 24 23:31:36 secondary    primary      Remote is in secondary hold

```

Redundancy Group 2 , Current State: secondary, Weight: 255

```

Time          From          To          Reason
Feb 24 23:16:10 hold          secondary    Hold timer expired

```

Chassis cluster LED information:

```
Current LED color: Amber
Last LED change reason: Monitored objects are down
```

## Meaning

The sample output confirms that in node 0, monitoring interfaces ge-0/0/1 and ge-0/0/2 are down. The weight of RG1 on node 0 reached zero value, which triggered RG1 failover during use of the `show chassis cluster status` command.

For RG2, the default weight of 255 is set for redundant Ethernet interface 2 (reth2). When interface monitoring is required, we recommend that you use the default weight when you do not have backup links like those in RG1. That is, if interface ge-0/0/3 is disabled, it immediately triggers failover because the weight becomes 0 (255 minus 225), as indicated in the next verification section.

## Verifying Interface Status After Disabling ge-0/0/3

### Purpose

Verify that interface ge-0/0/3 is disabled on node 0.

### Action

From configuration mode, enter the `set interface ge-0/0/3 disable` command.

```
{primary:node0}
user@host# set interface ge-0/0/3 disable
user@host# commit

node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

{primary:node0}
user@host# show interfaces ge-0/0/3
disable;
gigether-options {
```



```

    redundant-parent reth2;
}

```

## Meaning

The sample output confirms that interface ge-0/0/3 is disabled.

## Verifying Chassis Cluster Status After Disabling Interface ge-0/0/3

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```

{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary no no None
node1 1 secondary no no None

Redundancy group: 1 , Failover count: 2
node0 0 secondary no no IF
node1 100 primary no no None

Redundancy group: 2 , Failover count: 2

```

node0	0	secondary	no	no	IF
node1	100	primary	no	no	None

## Meaning

Use the `show chassis cluster status` command to confirm that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the secondary node.

## Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/3

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
  ----  -
  0      em0        Up                 Disabled
  1      em1        Down               Disabled

Fabric link status: Up

Fabric interfaces:
  Name      Child-interface  Status
              (Physical/Monitored)
  ----      -
  fab0     ge-0/0/0        Up / Up
  fab0
  fab1     ge-8/0/0        Up / Up
  fab1

Redundant-ethernet Information:
  Name      Status  Redundancy-group
  ----      -
  reth0     Up      1
```

```

reth1      Up      1
reth2      Up      2

```

Redundant-pseudo-interface Information:

```

Name       Status    Redundancy-group
lo0        Up        0

```

Interface Monitoring:

```

Interface   Weight   Status   Redundancy-group
ge-8/0/2    120     Up        1
ge-8/0/1    150     Up        1
ge-0/0/2    140     Down      1
ge-0/0/1    130     Down      1
ge-8/0/3    255     Up        2
ge-0/0/3    255     Down      2

```

## Meaning

The sample output confirms that monitoring interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 are down.

## Verifying Chassis Cluster Information After Disabling Interface ge-0/0/3

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster information` command.

```

{primary:node0}
user@host> show chassis cluster information

```

```

node0:
-----

```

Redundancy Group Information:

```

Redundancy Group 0 , Current State: primary, Weight: 255

```

```

Time          From          To          Reason

```

```

Feb 24 22:56:27 hold          secondary  Hold timer expired
Feb 24 22:56:34 secondary    primary    Better priority (254/1)

```

Redundancy Group 1 , Current State: secondary, Weight: -15

```

Time          From          To          Reason
Feb 24 23:16:12 hold          secondary    Hold timer expired
Feb 24 23:16:12 secondary    primary      Remote yield (0/0)
Feb 24 23:31:36 primary      secondary-hold Monitor failed: IF
Feb 24 23:31:37 secondary-hold secondary    Ready to become secondary

```

Redundancy Group 2 , Current State: secondary, Weight: 0

```

Time          From          To          Reason
Feb 24 23:16:12 hold          secondary    Hold timer expired
Feb 24 23:16:13 secondary    primary      Remote yield (0/0)
Feb 24 23:35:57 primary      secondary-hold Monitor failed: IF
Feb 24 23:35:58 secondary-hold secondary    Ready to become secondary

```

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

Failure Information:

Interface Monitoring Failure Information:

Redundancy Group 1, Monitoring status: Failed

```

Interface          Status
ge-0/0/2           Down
ge-0/0/1           Down

```

Redundancy Group 2, Monitoring status: Failed

```

Interface          Status
ge-0/0/3           Down

```

node1:

-----  
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

```

Time          From          To          Reason
Feb 24 22:56:34 hold          secondary    Hold timer expired

```

```
Redundancy Group 1 , Current State: primary, Weight: 255
```

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired
Feb 24 23:31:36	secondary	primary	Remote is in secondary hold

```
Redundancy Group 2 , Current State: primary, Weight: 255
```

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired
Feb 24 23:35:57	secondary	primary	Remote is in secondary hold

```
Chassis cluster LED information:
```

```
Current LED color: Amber
```

```
Last LED change reason: Monitored objects are down
```

## Meaning

The sample output confirms that in node 0, monitoring interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 are down.

In regard to RG1, allowing any interface in node 0 go up triggers a failover only if the preempt option is enabled. In the example, preempt is not enabled. Therefore the node should return to normal, with no monitor failure showing for RG1.

## Verifying That Interface ge-0/0/2 Is Enabled

### Purpose

Verify that interface ge-0/0/2 is enabled on node 0.

### Action

From configuration mode, enter the delete interfaces ge-0/0/2 disable command.

```
{primary:node0}
user@host# delete interfaces ge-0/0/2 disable
user@host# commit

node0:
```

```

configuration check succeeds
node1:
commit complete
node0:
commit complete

```

## Meaning

The sample output confirms that interface ge-0/0/2 disable is deleted.

## Verifying Chassis Cluster Status After Enabling Interface ge-0/0/2

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```

{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring      MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary no no None
node1 1 secondary no no None

Redundancy group: 1 , Failover count: 2
node0 200 secondary no no None

```

```

node1 100      primary      no      no      None

Redundancy group: 2 , Failover count: 2
node0 0       secondary   no      no      IF
node1 100     primary     no      no      None

```

## Meaning

Use the `show chassis cluster status` command to confirm that devices in the chassis cluster are communicating properly, with as one device functioning as the primary node and the other as the secondary node.

## Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/2

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```

{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
  ----  -
  0      em0        Up                 Disabled
  1      em1        Down                Disabled

Fabric link status: Up

Fabric interfaces:
  Name      Child-interface  Status
              (Physical/Monitored)
  ----      -
  fab0     ge-0/0/0        Up / Up
  fab0
  fab1     ge-8/0/0        Up / Up
  fab1

```

## Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1
reth2	Up	2

## Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

## Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-8/0/2	120	Up	1
ge-8/0/1	150	Up	1
ge-0/0/2	140	Up	1
ge-0/0/1	130	Down	1
ge-8/0/3	255	Up	2
ge-0/0/3	255	Down	2

## Meaning

The sample output confirms that monitoring interfaces ge-0/0/1 and ge-0/0/3 are down. Monitoring interface ge-0/0/2 is up after the disable has been deleted.

## Verifying Chassis Cluster Information After Enabling Interface ge-0/0/2

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster information` command.

```
{primary:node0}
user@host> show chassis cluster information
```

```
node0:
```

```
-----
Redundancy Group Information:
```



Redundancy Group 0 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:27	hold	secondary	Hold timer expired
Feb 24 22:56:34	secondary	primary	Better priority (254/1)

Redundancy Group 1 , Current State: secondary, Weight: 125

Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:12	secondary	primary	Remote yield (0/0)
Feb 24 23:31:36	primary	secondary-hold	Monitor failed: IF
Feb 24 23:31:37	secondary-hold	secondary	Ready to become secondary

Redundancy Group 2 , Current State: secondary, Weight: 0

Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:13	secondary	primary	Remote yield (0/0)
Feb 24 23:35:57	primary	secondary-hold	Monitor failed: IF
Feb 24 23:35:58	secondary-hold	secondary	Ready to become secondary

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

Failure Information:

Interface Monitoring Failure Information:

Redundancy Group 1, Monitoring status: Unhealthy

Interface	Status
ge-0/0/1	Down

Redundancy Group 2, Monitoring status: Failed

Interface	Status
ge-0/0/3	Down

node1:

-----  
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:34	hold	secondary	Hold timer expired

Redundancy Group 1 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired
Feb 24 23:31:36	secondary	primary	Remote is in secondary hold

Redundancy Group 2 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired
Feb 24 23:35:57	secondary	primary	Remote is in secondary hold

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

## Meaning

The sample output confirms that in node 0, monitoring interfaces ge-0/0/1 and ge-0/0/3 are down. Monitoring interface ge-0/0/2 is active after the disable has been deleted.

## Verifying Chassis Cluster RG2 Preempt

### Purpose

Verify that the chassis cluster RG2 is preempted on node 0.

### Action

From configuration mode, enter the `set chassis cluster redundancy-group 2 preempt` command.

```
{primary:node0}
user@host# set chassis cluster redundancy-group 2 preempt
user@host# commit

node0:
configuration check succeeds
```

```
node1:
commit complete
node0:
commit complete
```

## Meaning

The sample output confirms that chassis cluster RG2 preempted on node 0.

In the next section, you check that RG2 fails over back to node 0 when preempt is enabled when the disabled node 0 interface is brought online.

## Verifying Chassis Cluster Status After Preempting RG2

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring     IP IP monitoring
  LB Loopback monitoring      MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary no no None
node1 1 secondary no no None
```

```

Redundancy group: 1 , Failover count: 2
node0 200      secondary    no      no      None
node1 100      primary      no      no      None

Redundancy group: 2 , Failover count: 2
node0 0        secondary    yes     no      IF
node1 100      primary      yes     no      None

```

## Meaning

Use the `show chassis cluster status` command to confirm that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the secondary node.

## Verifying That Interface ge-0/0/3 Is Enabled

### Purpose

Verify that interface ge-0/0/3 is enabled on node 0.

### Action

From configuration mode, enter the `delete interfaces ge-0/0/3 disable` command.

```

{primary:node0}
user@host# delete interfaces ge-0/0/3 disable
user@host# commit

node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

```

## Meaning

The sample output confirms that interface ge-0/0/3 disable has been deleted.

## Verifying Chassis Cluster Status After Enabling Interface ge-0/0/3

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring            SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary no no None
node1 1 secondary no no None

Redundancy group: 1 , Failover count: 2
node0 200 secondary no no None
node1 100 primary no no None

Redundancy group: 2 , Failover count: 3
node0 200 primary yes no None
node1 100 secondary yes no None
```

### Meaning

Use the `show chassis cluster status` command to confirm that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the secondary node.

## Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/3

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
  0      em0        Up                 Disabled
  1      em1        Down               Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
          (Physical/Monitored)
  fab0    ge-0/0/0         Up / Up
  fab0
  fab1    ge-8/0/0         Up / Up
  fab1

Redundant-ethernet Information:
  Name    Status  Redundancy-group
  reth0   Up      1
  reth1   Up      1
  reth2   Up      2

Redundant-pseudo-interface Information:
  Name    Status  Redundancy-group
  lo0     Up      0

Interface Monitoring:
  Interface  Weight  Status  Redundancy-group
  ge-8/0/2   120    Up      1
```

ge-8/0/1	150	Up	1
ge-0/0/2	140	Up	1
ge-0/0/1	130	Down	1
ge-8/0/3	255	Up	2
ge-0/0/3	255	Up	2

## Meaning

The sample output confirms that monitoring interface ge-0/0/1 is down. Monitoring interfaces ge-0/0/2, and ge-0/0/3 are up after deleting the disable.

## Verifying Chassis Cluster Information After Enabling Interface ge-0/0/3

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster information` command.

```
{primary:node0}
user@host> show chassis cluster information

node0:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: primary, Weight: 255

Time           From           To           Reason
Feb 24 22:56:27 hold           secondary    Hold timer expired
Feb 24 22:56:34 secondary     primary      Better priority (254/1)

Redundancy Group 1 , Current State: secondary, Weight: 125

Time           From           To           Reason
Feb 24 23:16:12 hold           secondary    Hold timer expired
Feb 24 23:16:12 secondary     primary      Remote yield (0/0)
Feb 24 23:31:36 primary          secondary-hold Monitor failed: IF
```

Feb 24 23:31:37 secondary-hold secondary Ready to become secondary

Redundancy Group 2 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:13	secondary	primary	Remote yield (0/0)
Feb 24 23:35:57	primary	secondary-hold	Monitor failed: IF
Feb 24 23:35:58	secondary-hold	secondary	Ready to become secondary
Feb 24 23:45:45	secondary	primary	Remote is in secondary hold

Chassis cluster LED information:

Current LED color: Green

Last LED change reason: No failures

Failure Information:

Interface Monitoring Failure Information:

Redundancy Group 1, Monitoring status: Unhealthy

Interface	Status
ge-0/0/1	Down

node1:

-----

Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:34	hold	secondary	Hold timer expired

Redundancy Group 1 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired
Feb 24 23:31:36	secondary	primary	Remote is in secondary hold

Redundancy Group 2 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired
Feb 24 23:35:57	secondary	primary	Remote is in secondary hold
Feb 24 23:45:45	primary	secondary-hold	Preempt (100/200)



```
Feb 24 23:45:46 secondary-hold secondary      Ready to become secondary
```

```
Chassis cluster LED information:
```

```
Current LED color: Amber
```

```
Last LED change reason: Monitored objects are down
```

## Meaning

The sample output confirms that in node 0, monitoring interface ge-0/0/1 is down. RG2 on node 0 state is back to primary state (because of the preempt enable) with a healthy weight of 255 when interface ge-0/0/3 is back up.

## SEE ALSO

[Example: Configuring Chassis Cluster Redundancy Groups | 87](#)

## RELATED DOCUMENTATION

[Monitoring IP Addresses on a Chassis Cluster | 227](#)

[Configuring Cluster Failover Parameters | 248](#)

[Chassis Cluster Redundancy Group Failover | 254](#)

# Monitoring IP Addresses on a Chassis Cluster

## IN THIS SECTION

- [IP Monitoring Overview | 228](#)
- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring | 231](#)
- [Example: Configure Chassis Cluster Redundancy Group IP Address Monitoring | 233](#)
- [Example: Configuring IP Monitoring on SRX5000 Line Devices for IOC2 and IOC3 | 238](#)

Redundancy group IP address monitoring checks end-to-end connectivity and allows a redundancy group to fail over if reth interface fails to reach a configured IP address. Redundancy groups on both devices in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable. For more information, see the following topics:

## IP Monitoring Overview

### IN THIS SECTION

- [Benefits of Monitoring IP Addresses in a Chassis Cluster | 230](#)

IP monitoring checks the end-to-end connectivity of configured IP addresses and allows a redundancy group to automatically fail over when the monitored IP address is not reachable through the redundant Ethernet (reth) interface. Both the primary and secondary nodes in the chassis cluster monitor specific IP addresses to determine whether an upstream device in the network is reachable.

IP monitoring allows for failover based upon end to-end reachability of a configured monitored IP address. On SRX Series Firewalls, the reachability test is done by sending a ping to the monitored IP address from both the primary node and the secondary node through the reth interface and checking if a response is returned. The monitored IP address can be on a directly connected host in the same subnet as the reth interface or on a remote device reachable through a next-hop router.

The reachability states of the monitored IP address are reachable, unreachable, and unknown. The status is "unknown" if Packet Forwarding Engines are not yet up and running. The status changes to either "reachable" or "unreachable," depending on the corresponding message from the Packet Forwarding Engine.

We do not recommend configuring chassis cluster IP monitoring on Redundancy Group 0 (RG0) for SRX Series Firewalls.

[Table 19 on page 229](#) provides details of different combinations of monitored results from both the primary and secondary nodes, and the corresponding actions by the Juniper Services Redundancy Protocol (jsrpd) process.

**Table 19: IP Monitoring Results and Failover Action**

Primary Node Monitored Status	Secondary Node Monitored Status	Failover Action
Reachable	Reachable	No action
Unreachable	Reachable	Failover
Reachable	Unreachable	No action
Unreachable	Unreachable	No action

- You can configure up to 64 IP addresses for IP monitoring on SRX5000 line devices.
- On SRX Branch Series devices, when the reth interface has more than one physical interface configured, IP monitoring for redundant groups is not supported. The SRX uses the lowest interface in the bundle for tracking on the secondary node. If the peer forwards the reply on any other port except the one it received it on, the SRX drops it.
- The minimum interval of IP monitoring is 1 second and the maximum is 30 seconds. Default interval is 1 second.
- The minimum threshold of IP monitoring is 5 requests and the maximum is 15 requests. If the IP monitoring request does not receive a response for consecutive requests (exceeding the threshold value), IP monitoring reports that the monitored IP is unreachable. Default value for the threshold is 5.
- Reth interface not associated with Redundancy Group (RG) in IP monitoring CLI configuration is supported.

[Table 20 on page 229](#) provides details on multiple interface combinations of IOC2 and IOC3 with maximum MAC numbers.

**Table 20: Maximum MACs Supported for IP Monitoring on IOC2 and IOC3**

Cards	Interfaces	Maximum MACs Supported for IP Monitoring
IOC2 (SRX5K-MPC)	10XGE	10

**Table 20: Maximum MACs Supported for IP Monitoring on IOC2 and IOC3 (Continued)**

Cards	Interfaces	Maximum MACs Supported for IP Monitoring
	20GE	20
	2X40GE	2
	1X100GE	1
IOC3 (SRX5K-MPC3-40G10G or SRX5K-MPC3-100G10G)	24x10GE	24
	6x40GE	6
	2x100GE + 4x10GE	6

Note the following limitations for IP monitoring support on SRX5000 line IOC2 and IOC3:

- IP monitoring is supported through the reth or the RLAG interface. If your configuration does not specify either of these interfaces, the route lookup returns a non-reth/RLAG interface, which results in a failure report.
- Equal-cost multipath (ECMP) routing is not supported in IP monitoring.

### Benefits of Monitoring IP Addresses in a Chassis Cluster

- Helps determine the status of a specific IP address in a Chassis Cluster setup as unknown, reachable or unreachable.
- Initiates failover based upon end to-end reachability of a configured monitored IP address. If the monitored IP address becomes unreachable, the redundancy group can fail over to its backup to maintain service.

### SEE ALSO

[SRX5400, SRX5600, and SRX5800 Services Gateway Card Overview](#)

[Chassis Cluster Redundancy Groups | 82](#)

## Understanding Chassis Cluster Redundancy Group IP Address Monitoring

Redundancy group IP address monitoring checks end-to-end connectivity and allows a redundancy group to fail over because of the inability of a redundant Ethernet interface (known as a *reth*) to reach a configured IP address. Redundancy groups on both devices in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable. The redundancy group can be configured such that if the monitored IP address becomes unreachable, the redundancy group will fail over to its backup to maintain service. The primary difference between this monitoring feature and interface monitoring is that IP address monitoring allows for failover when the interface is still up but the network device it is connected to is not reachable for some reason. It may be possible under those circumstances for the other node in the cluster to route traffic around the problem.

If you want to dampen the failovers occurring because of IP address monitoring failures, use the `hold-down-interval` statement.

IP address monitoring configuration allows you to set not only the address to monitor and its failover weight but also a global IP address monitoring threshold and weight. Only after the IP address monitoring global-threshold is reached because of cumulative monitored address reachability failure will the IP address monitoring global-weight value be deducted from the redundant group's failover threshold. Thus, multiple addresses can be monitored simultaneously as well as monitored to reflect their importance to maintaining traffic flow. Also, the threshold value of an IP address that is unreachable and then becomes reachable again will be restored to the monitoring threshold. This will not, however, cause a failback unless the `preempt` option has been enabled.

When configured, the IP address monitoring failover value (global-weight) is considered along with interface monitoring—if set—and built-in failover monitoring, including SPU monitoring, cold-sync monitoring, and NPC monitoring (on supported platforms). The main IP addresses that should be monitored are router gateway addresses to ensure that valid traffic coming into the services gateway can be forwarded to the appropriate network router.

Starting in Junos OS Release 12.1X46-D35 and Junos OS Release 17.3R1, for all SRX Series Firewalls, the *reth* interface supports proxy ARP.

One Services Processing Unit (SPU) or Packet Forwarding Engine (PFE) per node is designated to send Internet Control Message Protocol (ICMP) ping packets for the monitored IP addresses on the cluster. The primary PFE sends ping packets using Address Resolution Protocol (ARP) requests resolved by the Routing Engine (RE). The source for these pings is the redundant Ethernet interface MAC and IP addresses. The secondary PFE resolves ARP requests for the monitored IP address itself. The source for these pings is the physical child MAC address and a secondary IP address configured on the redundant Ethernet interface. For the ping reply to be received on the secondary interface, the I/O card (IOC), central PFE processor, or Flex IOC adds both the physical child MAC address and the redundant Ethernet interface MAC address to its MAC table. The secondary PFE responds with the physical child MAC address to ARP requests sent to the secondary IP address configured on the redundant Ethernet interface.

**NOTE:** IP address monitoring is not supported on SRX5000 line devices if the redundant Ethernet interface is configured for a VPN routing and forwarding (VRF) instance.

The default interval to check the reachability of a monitored IP address is once per second. The interval can be adjusted using the `retry-interval` command. The default number of permitted consecutive failed ping attempts is 5. The number of allowed consecutive failed ping attempts can be adjusted using the `retry-count` command. After failing to reach a monitored IP address for the configured number of consecutive attempts, the IP address is determined to be unreachable and its failover value is deducted from the redundancy group's `global-threshold`.

On SRX5600 and SRX5800 devices, only two of the 10 ports on each PIC of 40-port 1-Gigabit Ethernet I/O cards (IOCs) can simultaneously enable IP address monitoring. Because there are four PICs per IOC, this permits a total of eight ports per IOC to be monitored. If more than two ports per PIC on 40-port 1-Gigabit Ethernet IOCs are configured for IP address monitoring, the commit will succeed but a log entry will be generated, and the accuracy and stability of IP address monitoring cannot be ensured. This limitation does not apply to any other IOCs or devices.

Once the IP address is determined to be unreachable, its weight is deducted from the `global-threshold`. If the recalculated `global-threshold` value is not 0, the IP address is marked unreachable, but the `global-weight` is not deducted from the redundancy group's threshold. If the redundancy group IP monitoring `global-threshold` reaches 0 and there are unreachable IP addresses, the redundancy group will continuously fail over and fail back between the nodes until either an unreachable IP address becomes reachable or a configuration change removes unreachable IP addresses from monitoring. Note that both default and configured `hold-down-interval` failover dampening is still in effect.

Every redundancy group  $x$  has a threshold tolerance value initially set to 255. When an IP address monitored by redundancy group  $x$  becomes unavailable, its weight is subtracted from the redundancy group  $x$ 's threshold. When redundancy group  $x$ 's threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

A redundancy group  $x$  failover occurs because the cumulative weight of the redundancy group  $x$ 's monitored IP addresses and other monitoring has brought its threshold value to 0. When the monitored IP addresses of redundancy group  $x$  on both nodes reach their thresholds at the same time, redundancy group  $x$  is primary on the node with the lower node ID, which is typically node 0.

Upstream device failure detection for the `chassis cluster` feature is supported on SRX Series Firewalls.

Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, configuring Address Resolution Protocol (ARP) request throttling is supported on SRX5000 line devices. This feature allows you to bypass the previously hard-coded ARP request throttling time default (10 seconds per SPU for each IP address) and set the time to a greater value (10 through 100 seconds). Setting the throttling time

to a greater value reduces the high utilization of the Routing Engine, allowing it to work more efficiently. You can configure the ARP request throttling time using the `set forwarding-options next-hop arp-throttle <seconds>` command.

Monitoring can be accomplished only if the IP address is reachable on a redundant Ethernet interface (known as a reth in CLI commands and interface listings), and IP addresses cannot be monitored over a tunnel. For an IP address to be monitored through a redundant Ethernet interface on a secondary cluster node, the interface must have a secondary IP address configured. IP address monitoring cannot be used on a chassis cluster running in transparent mode. The maximum number of monitoring IP addresses that can be configured per cluster is 64 for the SRX5000 line of devices, SRX1500, SRX1600, SRX2300, and SRX4000 line of devices.

Redundancy group IP address monitoring is not supported for IPv6 destinations.

## Example: Configure Chassis Cluster Redundancy Group IP Address Monitoring

### IN THIS SECTION

- [Requirements | 233](#)
- [Overview | 234](#)
- [Configuration | 234](#)
- [Verification | 237](#)

This example shows how to configure redundancy group IP address monitoring for an SRX Series Firewall in a chassis cluster.

### Requirements

Before you begin:

- Set the chassis cluster node ID and cluster ID. See ["Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster "](#) on page 45
- Configure the chassis cluster management interface. See ["Example: Configuring the Chassis Cluster Management Interface"](#) on page 51.

- Configure the chassis cluster fabric. See "[Example: Configuring the Chassis Cluster Fabric Interfaces](#)" on page 65.

## Overview

You can configure redundancy groups to monitor upstream resources by pinging specific IP addresses that are reachable through redundant Ethernet interfaces on either node in a cluster. You can also configure global threshold, weight, retry interval, and retry count parameters for a redundancy group. When a monitored IP address becomes unreachable, the weight of that monitored IP address is deducted from the redundancy group IP address monitoring global threshold. When the global threshold reaches 0, the global weight is deducted from the redundancy group threshold. The retry interval determines the ping interval for each IP address monitored by the redundancy group. The pings are sent as soon as the configuration is committed. The retry count sets the number of allowed consecutive ping failures for each IP address monitored by the redundancy group.

In this example, you configure the following settings for redundancy group 1:

- IP address to monitor—10.1.1.10
- IP address monitoring global-weight—100
- IP address monitoring global-threshold—200

The threshold applies cumulatively to all IP addresses monitored by the redundancy group.

- IP address retry-interval—3 seconds
- IP address retry-count—10
- Weight—100
- Redundant Ethernet interface—reth1.0
- Secondary IP address—10.1.1.101

## Configuration

### IN THIS SECTION

- [Procedure | 235](#)



## 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
user@host#
set chassis cluster redundancy-group 1 ip-monitoring global-weight 100
set chassis cluster redundancy-group 1 ip-monitoring global-threshold 200
set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
set chassis cluster redundancy-group 1 ip-monitoring family inet 10.1.1.10 weight 100 interface
reth1.0 secondary-ip-address 10.1.1.101
```

### Step-by-Step Procedure

To configure redundancy group IP address monitoring:

1. Specify a global monitoring weight.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-weight 100
```

2. Specify the global monitoring threshold.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-threshold 200
```

3. Specify the retry interval.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
```

#### 4. Specify the retry count.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
```

#### 5. Specify the IP address to be monitored, weight, redundant Ethernet interface, and secondary IP address.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 10.1.1.10 weight
100 interface reth1.0 secondary-ip-address 10.1.1.101
```

## Results

From configuration mode, confirm your configuration by entering the `show chassis cluster redundancy-group 1` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
{primary:node0}[edit]
user@host# show chassis cluster redundancy-group 1
ip-monitoring {
  global-weight 100;
  global-threshold 200;
  family {
    inet {
      10.1.1.10 {
        weight 100;
        interface reth1.0 secondary-ip-address 10.1.1.101;
      }
    }
  }
}
```

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

## Verification

### IN THIS SECTION

- [Verifying the Status of Monitored IP Addresses for a Redundancy Group | 237](#)

## Verifying the Status of Monitored IP Addresses for a Redundancy Group

### Purpose

Verify the status of monitored IP addresses for a redundancy group.

### Action

From operational mode, enter the `show chassis cluster ip-monitoring status` command. For information about a specific group, enter the `show chassis cluster ip-monitoring status redundancy-group` command.

```
{primary:node0}
user@host> show chassis cluster ip-monitoring status
node0:
-----

Redundancy group: 1
Global threshold: 200
Current threshold: -120

IP address          Status      Failure count Reason  Weight
10.1.1.10           reachable  0         n/a    100
10.1.1.101          reachable  0         n/a    100

node1:
-----

Redundancy group: 1
Global threshold: 200
Current threshold: -120

IP address          Status      Failure count Reason  Weight
```

10.1.1.10	reachable	0	n/a	100
10.1.1.101	reachable	0	n/a	100

## Example: Configuring IP Monitoring on SRX5000 Line Devices for IOC2 and IOC3

### IN THIS SECTION

- [Requirements | 238](#)
- [Overview | 238](#)
- [Configuration | 239](#)
- [Verification | 246](#)

This example shows how to monitor IP address on an SRX5000 line device with chassis cluster enabled.

### Requirements

This example uses the following hardware and software:

- Two SRX5400 Services Gateways with MIC (SRX-MIC-10XG-SFPP [IOC2]), and one Ethernet switch
- Junos OS Release 15.1X49-D30

The procedure mentioned in this example is also applicable to IOC3.

Before you begin:

- Physically connect the two SRX5400 devices (back-to-back for the fabric and control ports).
- Configure the two devices to operate in a chassis cluster.

### Overview

#### IN THIS SECTION

- [Topology | 239](#)

IP address monitoring checks end-to-end reachability of the configured IP address and allows a redundancy group to automatically fail over when it is not reachable through the child link of redundant Ethernet (reth) interface. Redundancy groups on both devices, or nodes, in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable.

## Topology

In this example, two SRX5400 devices in a chassis cluster are connected to an Ethernet switch. The example shows how the redundancy groups can be configured to monitor key upstream resources reachable through redundant Ethernet interfaces on either node in a cluster.

You set the system to send pings every second, with 10 losses required to declare unreachability to peer. You also set up a secondary IP address to allow testing from the secondary node.

In this example, you configure the following settings for redundancy group 1:

- IP address to be monitored—192.0.2.2, 198.51.100.2, 203.0.113.2
- IP monitoring global-weight—255
- IP monitoring global-threshold—240
- IP monitoring retry-interval—3 seconds
- IP monitoring retry-count—10
- Weight for monitored IP address—80
- Secondary IP addresses— 192.0.2.12, 198.51.100.12, 203.0.113.12

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 240](#)
- [Configuring IP Monitoring on a 10x10GE SFP+ MIC | 241](#)

## 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 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.

```
set chassis cluster reth-count 10
set chassis cluster control-ports fpc 3 port 0
set chassis cluster control-ports fpc 0 port 0
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 199
set chassis cluster redundancy-group 1 ip-monitoring global-weight 255
set chassis cluster redundancy-group 1 ip-monitoring global-threshold 240
set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
set chassis cluster redundancy-group 1 ip-monitoring family inet 192.0.2.2 weight 80
set chassis cluster redundancy-group 1 ip-monitoring family inet 192.0.2.2 interface reth0.0
secondary-ip-address 192.0.2.12
set chassis cluster redundancy-group 1 ip-monitoring family inet 198.51.100.2 weight 80
set chassis cluster redundancy-group 1 ip-monitoring family inet 198.51.100.2 interface reth1.0
secondary-ip-address 198.51.100.12
set chassis cluster redundancy-group 1 ip-monitoring family inet 203.0.113.2 weight 80
set chassis cluster redundancy-group 1 ip-monitoring family inet 203.0.113.2 interface reth2.0
secondary-ip-address 203.0.113.12
set interfaces xe-1/2/1 gigether-options redundant-parent reth0
set interfaces xe-1/2/2 gigether-options redundant-parent reth2
set interfaces xe-1/2/3 gigether-options redundant-parent reth1
set interfaces xe-4/2/1 gigether-options redundant-parent reth0
set interfaces xe-4/2/2 gigether-options redundant-parent reth2
set interfaces xe-4/2/3 gigether-options redundant-parent reth1
set interfaces fab0 fabric-options member-interfaces xe-1/2/0
set interfaces fab1 fabric-options member-interfaces xe-4/2/0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.0.2.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 198.51.100.1/24
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth2 unit 0 family inet address 203.0.113.1/24
set security zones security-zone HOST host-inbound-traffic system-services any-service
```

```
set security zones security-zone HOST host-inbound-traffic protocols all
set security zones security-zone HOST interfaces all
```

## Configuring IP Monitoring on a 10x10GE SFP+ MIC

### Step-by-Step Procedure

To configure IP monitoring on a 10x10GE SFP+ MIC:

1. Specify the number of redundant Ethernet interfaces.

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 10
```

2. Configure the control ports.

```
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 3 port 0
user@host# set chassis cluster control-ports fpc 0 port 0
```

3. Configure fabric interfaces.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces xe-1/2/0
user@host# set interfaces fab1 fabric-options member-interfaces xe-4/2/0
```

4. Specify a redundancy group's priority for primacy on each node of the cluster. The higher number takes precedence.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 200
user@host# set chassis cluster redundancy-group 1 node 1 priority 199
```

5. Configure IP monitoring under redundancy-group 1 with global weight, global threshold, retry interval and retry count.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-weight 255
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-threshold 240
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
```

6. Configure the redundant Ethernet interfaces to redundancy-group 1. Assign a weight to the IP address to be monitored, and configure a secondary IP address that will be used to send packets from the secondary node to track the IP address being monitored.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 192.0.2.2 weight 80
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 192.0.2.2 interface reth0.0 secondary-ip-address 192.0.2.12
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 198.51.100.2 weight 80
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 198.51.100.2 interface reth1.0 secondary-ip-address 198.51.100.12
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 203.0.113.2 weight 80
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 203.0.113.2 interface reth2.0 secondary-ip-address 203.0.113.12
```

7. Assign child interfaces for the redundant Ethernet interfaces from node 0, node 1, and node 2.

```
{primary:node0}[edit]
user@host# set interfaces xe-1/2/1 gigger-options redundant-parent reth0
user@host# set interfaces xe-1/2/2 gigger-options redundant-parent reth2
user@host# set interfaces xe-1/2/3 gigger-options redundant-parent reth1
user@host# set interfaces xe-4/2/1 gigger-options redundant-parent reth0
user@host# set interfaces xe-4/2/2 gigger-options redundant-parent reth2
user@host# set interfaces xe-4/2/3 gigger-options redundant-parent reth1
```



## 8. Configure the redundant Ethernet interfaces to redundancy-group 1.

```
{primary:node0}[edit]
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 192.0.2.1/24
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 198.51.100.1/24
user@host# set interfaces reth2 redundant-ether-options redundancy-group 1
user@host# set interfaces reth2 unit 0 family inet address 203.0.113.1/24
```

## 9. Create security zone and assign interfaces to zone.

```
user@host# set security zones security-zone HOST host-inbound-traffic system-services any-
service
user@host# set security zones security-zone HOST host-inbound-traffic protocols all
user@host# set security zones security-zone HOST interfaces all
```

## Results

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

```
chassis {
  cluster {
    reth-count 10;
    redundancy-group 0 {
      node 0 priority 254;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 200;
      node 1 priority 199;
      ip-monitoring {
        global-weight 255;
        global-threshold 240;
        retry-interval 3;
        retry-count 10;
        family {
```

```
        inet {
            192.0.2.2 {
                weight 80;
                interface reth0.0 secondary-ip-address 192.0.2.12;
            }
            198.51.100.2 {
                weight 80;
                interface reth1.0 secondary-ip-address 198.51.100.12;
            }
            203.0.113.2 {
                weight 80;
                interface reth2.0 secondary-ip-address 203.0.113.12;
            }
        }
    }
}
interfaces {
    xe-1/2/1 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-1/2/2 {
        gigether-options {
            redundant-parent reth2;
        }
    }
    xe-1/2/3 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    xe-4/2/1 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-4/2/2 {
        gigether-options {
```

```
        redundant-parent reth2;
    }
}
xe-4/2/3 {
    gigheter-options {
        redundant-parent reth1;
    }
}
fab0 {
    fabric-options {
        member-interfaces {
            xe-1/2/0;
        }
    }
}
fab1 {
    fabric-options {
        member-interfaces {
            xe-4/2/0;
        }
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 198.51.100.1/24;
        }
    }
}
reth2 {
```

```
redundant-ether-options {
    redundancy-group 1;
}
unit 0 {
    family inet {
        address 203.0.113.1/24;
    }
}
}
```

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

## Verification

### IN THIS SECTION

- [Verifying IP Monitoring Status | 246](#)

Confirm the configuration is working properly.

### Verifying IP Monitoring Status

#### Purpose

Verify the IP status being monitored from both nodes and the failure count for both nodes.

#### Action

From operational mode, enter the `show chassis cluster ip-monitoring status` command.

```
show chassis cluster ip-monitoring status
```

```
node0:
```

```
-----
```

```
Redundancy group: 1
```

```
Global weight: 255
```

```
Global threshold: 240
```

Current threshold: 240

IP address	Status	Failure count	Weight	Reason
203.0.113.2	reachable	1	80	n/a
198.51.100.2	reachable	1	80	n/a
192.0.2.2	reachable	1	80	n/a

node1:

-----

Redundancy group: 1  
 Global weight: 255  
 Global threshold: 240  
 Current threshold: 240

IP address	Status	Failure count	Weight	Reason
203.0.113.2	reachable	2	80	n/a
198.51.100.2	reachable	1	80	n/a
192.0.2.2	reachable	2	80	n/a

## Meaning

All the monitored IP addresses are reachable.

## 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
15.1X49-D60	Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, configuring Address Resolution Protocol (ARP) request throttling is supported on SRX5000 line devices.
12.1X46-D35	Starting in Junos OS Release 12.1X46-D35 and Junos OS Release 17.3R1, for all SRX Series Firewalls, the reth interface supports proxy ARP.

## RELATED DOCUMENTATION

[Chassis Cluster Redundancy Groups | 82](#)

[Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Firewalls | 342](#)

[Monitoring Chassis Cluster Interfaces | 185](#)

# Configuring Cluster Failover Parameters

## IN THIS SECTION

- [Understanding Chassis Cluster Control Link Heartbeats, Failure, and Recovery | 248](#)
- [Example: Configuring Chassis Cluster Control Link Recovery | 251](#)

SRX Series devices in a chassis cluster uses heartbeat transmissions to determine the “health” of the control link. If the number of missed heartbeats has reached the configured threshold, the system assesses whether a failure condition exists. For more information, see the following topics:

## Understanding Chassis Cluster Control Link Heartbeats, Failure, and Recovery

### IN THIS SECTION

- [Understanding Chassis Cluster Control Link Heartbeats | 248](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery | 249](#)

## Understanding Chassis Cluster Control Link Heartbeats

You specify the heartbeat threshold and heartbeat interval when you configure the *chassis cluster*.

The system monitors the control link’s status by default.

For dual control links, which are supported on SRX5600 and SRX5800 lines, the Juniper Services Redundancy Protocol process (jsrpd) sends and receives the control heartbeat messages on both control links. As long as heartbeats are received on one of the control links, Junos OS considers the other node to be alive.

The product of the `heartbeat-threshold` option and the `heartbeat-interval` option defines the wait time before failover is triggered. The default values of these options produce a wait time of 3 seconds. A

heartbeat-threshold of 5 and a heartbeat-interval of 1000 milliseconds would yield a wait time of 5 seconds. Setting the heartbeat-threshold to 4 and the heartbeat-interval to 1250 milliseconds would also yield a wait time of 5 seconds.

In a chassis cluster environment, if more than 1000 logical interfaces are used, the cluster heartbeat timers are recommended to be increased from the default of 3 seconds. At maximum capacity on an SRX4600, SRX5400, SRX5600 or an SRX5800 device, we recommend that you increase the configured time before failover to at least 5 seconds.

## Understanding Chassis Cluster Control Link Failure and Recovery

If the control link fails, Junos OS changes the operating state of the secondary node to ineligible for a 180-second countdown. If the fabric link also fails during the 180 seconds, Junos OS changes the secondary node to primary; otherwise, after 180 seconds the secondary node state changes to disabled.

When the control link is down, a system log message is generated.

A control link failure is defined as not receiving heartbeats over the control link while heartbeats are still being received over the fabric link.

In the event of a legitimate control link failure, redundancy group 0 remains primary on the node on which it is currently primary, inactive redundancy groups  $x$  on the primary node become active, and the secondary node enters a disabled state.

When the secondary node is disabled, you can still log in to the management port and run diagnostics.

To determine if a legitimate control link failure has occurred, the system relies on redundant liveness signals sent across both the control link and the fabric link.

The system periodically transmits probes over the fabric link and heartbeat signals over the control link. Probes and heartbeat signals share a common sequence number that maps them to a unique time event. Junos OS identifies a legitimate control link failure if the following two conditions exist:

- The threshold number of heartbeats were lost.
- At least one probe with a sequence number corresponding to that of a missing heartbeat signal was received on the fabric link.

If the control link fails, the 180-second countdown begins and the secondary node state is ineligible. If the fabric link fails before the 180-second countdown reaches zero, the secondary node becomes primary because the loss of both links is interpreted by the system to indicate that the other node is dead. Because concurrent loss of both control and fabric links means that the nodes are no longer synchronizing states nor comparing priorities, both nodes might thus temporarily become primary, which is not a stable operating state. However, once the control link is reestablished, the node with the higher priority value automatically becomes primary, the other node becomes secondary, and the cluster returns to normal operation.

When a legitimate control link failure occurs, the following conditions apply:

- Redundancy group 0 remains primary on the node on which it is currently primary (and thus its Routing Engine remains active), and all redundancy groups  $x$  on the node become primary.

If the system cannot determine which Routing Engine is primary, the node with the higher priority value for redundancy group 0 is primary and its Routing Engine is active. (You configure the priority for each node when you configure the `redundancy-group` statement for redundancy group 0.)

- The system disables the secondary node.

To recover a device from the disabled mode, you must reboot the device. When you reboot the disabled node, the node synchronizes its dynamic state with the primary node.

If you make any changes to the configuration while the secondary node is disabled, execute the `commit` command to synchronize the configuration after you reboot the node. If you did not make configuration changes, the configuration file remains synchronized with that of the primary node.

You cannot enable preemption for redundancy group 0. If you want to change the primary node for redundancy group 0, you must do a manual failover.

When you use dual control links (supported on SRX5600 and SRX5800 devices), note the following conditions:

- Host inbound or outbound traffic can be impacted for up to 3 seconds during a control link failure. For example, consider a case where redundancy group 0 is primary on node 0 and there is a Telnet session to the Routing Engine through a network interface port on node 1. If the currently active control link fails, the Telnet session will lose packets for 3 seconds, until this failure is detected.
- A control link failure that occurs while the `commit` process is running across two nodes might lead to commit failure. In this situation, run the `commit` command again after 3 seconds.

For SRX5600 and SRX5800 devices, dual control links require a second Routing Engine on each node of the *chassis cluster*.

You can specify that control link recovery be done automatically by the system by setting the `control-link-recovery` statement. In this case, once the system determines that the control link is healthy, it issues an automatic reboot on the disabled node. When the disabled node reboots, the node joins the cluster again.



## Example: Configuring Chassis Cluster Control Link Recovery

### IN THIS SECTION

- Requirements | 251
- Overview | 251
- Configuration | 252

This example shows how to enable control link recovery, which allows the system to automatically take over after the control link recovers from a failure.

### Requirements

Before you begin:

- Understand chassis cluster control links. See "[Understanding Chassis Cluster Control Plane and Control Links](#)" on page 72.
- Understand chassis cluster dual control links. See "[Understanding Chassis Cluster Dual Control Links](#)" on page 163.
- Connect dual control links in a chassis cluster. See "[Dual Control Link Connections for SRX Series Firewalls in a Chassis Cluster](#)" on page 165.

### Overview

You can enable the system to perform control link recovery automatically. After the control link recovers, the system takes the following actions:

- It checks whether it receives at least three consecutive heartbeats on the control link or, in the case of dual control links (SRX5600 and SRX5800 devices only), on either control link. This is to ensure that the control link is not flapping and is healthy.
- After it determines that the control link is healthy, the system issues an automatic reboot irrespective of the state of the node (ineligible or disable) when the control link failed. When the node reboots, it can rejoin the cluster. There is no need for any manual intervention.

In this example, you enable chassis cluster control link recovery.

## Configuration

### IN THIS SECTION

- [Procedure | 252](#)

### Procedure

#### Step-by-Step Procedure

To enable chassis cluster control-link-recovery:

1. Enable control link recovery.

```
{primary:node0}[edit]  
user@host# set chassis cluster control-link-recovery
```

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

```
{primary:node0}[edit]  
user@host# commit
```

### RELATED DOCUMENTATION

[Chassis Cluster Dual Control Links | 163](#)

[Example: Configure Chassis Cluster Control Ports for Dual Control Links | 169](#)

# Understanding Chassis Cluster Resiliency

## IN THIS SECTION

- [Layer 1 for Detecting Hardware Faults and Software Failures | 253](#)
- [Layer 2 for Probing Critical Paths | 253](#)
- [Layer 3 for Detecting Control Link and Fabric Link Failure | 254](#)
- [Benefits | 254](#)

Junos OS uses a layered model to provide resiliency on SRX Series Firewalls that are in a chassis cluster setup. In the event of a software or hardware component failure, the layered model ensures that the system performance is not impacted.

## Layer 1 for Detecting Hardware Faults and Software Failures

Layer 1 identifies and detects the components that are causing the software failures and impacting the system performance. An alarm, syslog, or an SNMP trap is triggered to provide notifications about the failures.

## Layer 2 for Probing Critical Paths

Layer 2 probes the system's critical paths to detect hardware and software failures that are not detected by Layer 1.

Heartbeat communications validate the state of the paths between the two endpoints of the path. If any component in the path fails, communication is lost and the system health status is communicated using heartbeat messages sent from one end of the path to the other end.

## Layer 3 for Detecting Control Link and Fabric Link Failure

Layer 3 determines the system health information from Layer 1 and Layer 2, shares the health status between two nodes over the control links and fabric links, and makes the failover decision based on the health status of the two nodes and the heartbeat status of the control links and fabric links. An alarm, syslog, or an SNMP trap is triggered to provide notifications about the failures.

Layer 3 addresses the following software issues:

- em0 flapping
- Control path hardware or software component fails
- Fabric link is down and control link is alive
- Control link is down and fabric link is alive
- Both the control link and fabric link are down

## Benefits

- Improve the failover time and stability.
- Identify the exact location of the fault or failure.

### RELATED DOCUMENTATION

| [cluster \(Chassis\)](#)

# Chassis Cluster Redundancy Group Failover

## IN THIS SECTION

- [Understanding Chassis Cluster Redundancy Group Failover | 255](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover | 260](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover | 261](#)

- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers | 264](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover | 265](#)
- [Verifying Chassis Cluster Failover Status | 266](#)
- [Clearing Chassis Cluster Failover Status | 268](#)

A redundancy group (RG) includes and manages a collection of objects on both nodes of a cluster to provide high-availability. Each redundancy group acts as an independent unit of failover and is primary on only one node at a time. For more information, see the following topics:

## Understanding Chassis Cluster Redundancy Group Failover

### IN THIS SECTION

- [Preemptive Failover Delay Timer | 256](#)

*Chassis cluster* employs a number of highly efficient failover mechanisms that promote high availability to increase your system's overall reliability and productivity.

A redundancy group is a collection of objects that fail over as a group. Each redundancy group monitors a set of objects (physical interfaces), and each monitored object is assigned a weight. Each redundancy group has an initial threshold of 255. When a monitored object fails, the weight of the object is subtracted from the threshold value of the redundancy group. When the threshold value reaches zero, the redundancy group fails over to the other node. As a result, all the objects associated with the redundancy group fail over as well. Graceful restart of the routing protocols enables the SRX Series Firewall to minimize traffic disruption during a failover.

Back-to-back failovers of a redundancy group in a short interval can cause the cluster to exhibit unpredictable behavior. To prevent such unpredictable behavior, configure a dampening time between failovers. On failover, the previous primary node of a redundancy group moves to the secondary-hold state and stays in the secondary-hold state until the hold-down interval expires. After the hold-down interval expires, the previous primary node moves to the secondary state.

Configuring the hold-down interval prevents back-to-back failovers from occurring within the duration of hold-down interval.

The hold-down interval affects manual failovers, as well as automatic failovers associated with monitoring failures.

The default dampening time for a redundancy group 0 is 300 seconds (5 minutes) and is configurable to up to 1800 seconds with the `hold-down-interval` statement. For some configurations, such as those with a large number of routes or logical interfaces, the default interval or the user-configured interval might not be sufficient. In such cases, the system automatically extends the dampening time in increments of 60 seconds until the system is ready for failover.

Redundancy groups  $x$  (redundancy groups numbered 1 through 128) have a default dampening time of 1 second, with a range from 0 through 1800 seconds.

On SRX Series Firewalls, chassis cluster failover performance is optimized to scale with more logical interfaces. Previously, during redundancy group failover, gratuitous arp (GARP) is sent by the Juniper Services Redundancy Protocol (jsrpd) process running in the Routing Engine on each *logical interface* to steer the traffic to the appropriate node. With logical interface scaling, the Routing Engine becomes the checkpoint and GARP is directly sent from the Services Processing Unit (SPU).

## Preemptive Failover Delay Timer

A redundancy group is in the primary state (active) on one node and in the secondary state (backup) on the other node at any given time.

You can enable the preemptive behavior on both nodes in a redundancy group and assign a priority value for each node in the redundancy group. The node in the redundancy group with the higher configured priority is initially designated as the primary in the group, and the other node is initially designated as the secondary in the redundancy group.

When a redundancy group swaps the state of its nodes between primary and secondary, there is a possibility that a subsequent state swap of its nodes can happen again soon after the first state swap. This rapid change in states results in flapping of the primary and secondary systems.

Starting with Junos OS Release 17.4R1, a failover delay timer is introduced on SRX Series Firewalls in a chassis cluster to limit the flapping of redundancy group state between the secondary and the primary nodes in a preemptive failover.

To prevent the flapping, you can configure the following parameters:

- **Preemptive delay** –The preemptive delay time is the amount of time a redundancy group in a secondary state waits when the primary state is down in a preemptive failover before switching to the primary state. This delay timer delays the immediate failover for a configured period of time-- between 1 and 21,600 seconds.

- Preemptive limit–The preemptive limit restricts the number of preemptive failovers (between 1 to 50) during a configured preemptive period, when preemption is enabled for a redundancy group.
- Preemptive period–Time period (1 to 1440 seconds) during which the preemptive limit is applied, that is, number of configured preemptive failovers are applied when preempt is enabled for a redundancy group.

Consider the following scenario where you have configured a preemptive period as 300 seconds and preemptive limit as 50.

When the preemptive limit is configured as 50, the count starts at 0 and increments with a first preemptive failover; this process continues until the count reaches the configured preemptive limit, that is 50, before the preemptive period expires. When the preemptive limit (50) is exceeded, you must manually reset the preempt count to allow preemptive failovers to occur again.

When you have configured the preemptive period as 300 seconds, and if the time difference between the first preemptive failover and the current failover has already exceeded 300 seconds, and the preemptive limit (50) is not yet reached, then the preemptive period will be reset. After resetting, the last failover is considered as the first preemptive failover of the new preemptive period and the process starts all over again.

The preemptive delay can be configured independent of the failover limit. Configuring the preemptive delay timer does not change the existing preemptive behavior.

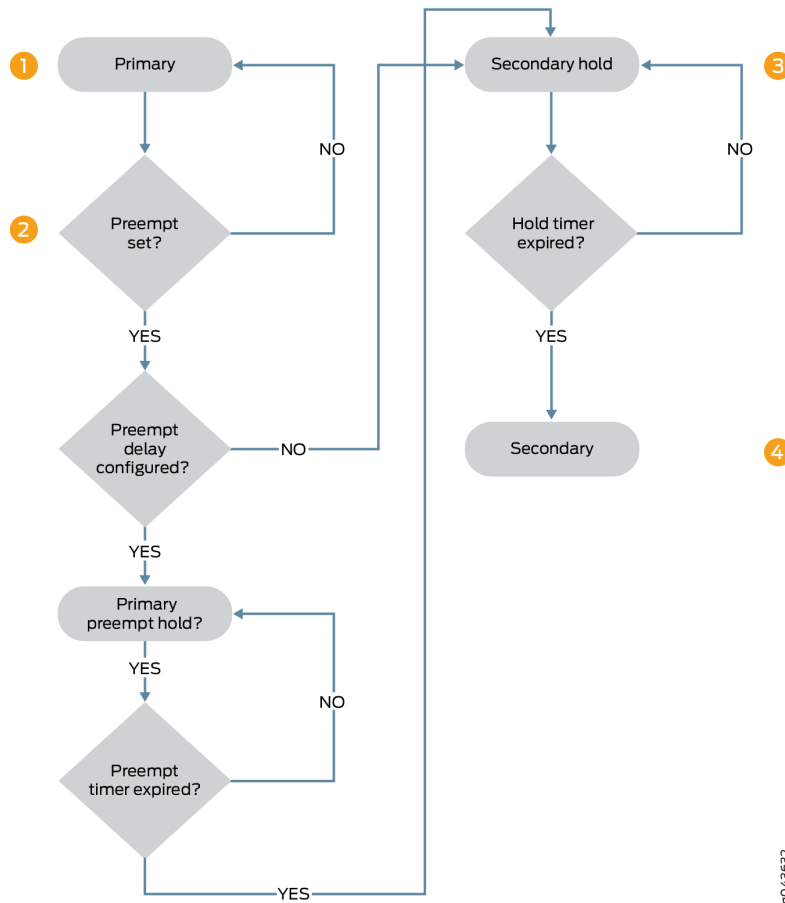
This enhancement enables the administrator to introduce a failover delay, which can reduce the number of failovers and result in a more stable network state due to the reduction in active /standby flapping within the redundancy group.

### **Understanding Transition from Primary State to Secondary State with Preemptive Delay**

Consider the following example, where a redundancy group, that is primary on the node 0 is ready for preemptive transition to the secondary state during a failover. Priority is assigned to each node and the preemptive option is also enabled for the nodes.

[Figure 37 on page 258](#) illustrates the sequence of steps in transition from the primary state to the secondary state when a preemptive delay timer is configured.

Figure 37: Transition from Primary State to Secondary State with Preemptive Delay



1. The node in the primary state is ready for preemptive transition to secondary state if the preemptive option is configured, and the node in secondary state has the priority over the node in primary state. If the preemptive delay is configured, the node in the primary state transitions to primary-preempt-hold state . If preemptive delay is not configured, then instant transition to the secondary state happens.
2. The node is in primary-preempt-hold state waiting for the preemptive delay timer to expire. The preemptive delay timer is checked and transition is held until the timer expires. The primary node stays in the primary-preempt-hold state until the timer expires, before transitioning to the secondary state.
3. The node transitions from primary-preempt-hold state into secondary-hold state and then to the secondary state.
4. The node stays in the secondary-hold state for the default time (1 second) or the configured time (a minimum of 300 seconds), and then the node transitions to the secondary state.



If your chassis cluster setup experiences an abnormal number of flaps, you must check your link and monitoring timers to make sure they are set correctly. Be careful when while setting timers in high latency networks to avoid getting false positives.

## Configuring Preemptive Delay Timer

This topic explains how to configure the delay timer on SRX Series Firewalls in a chassis cluster. Back-to-back redundancy group failovers that occur too quickly can cause a chassis cluster to exhibit unpredictable behavior. Configuring the delay timer and failover rate limit delays immediate failover for a configured period of time.

To configure the preemptive delay timer and failover rate limit between redundancy group failovers:

1. Enable preemptive failover for a redundancy group.

You can set the delay timer between 1 and 21,600 seconds. Default value is 1 second.

```
{primary:node1}
[edit chassis cluster redundancy-group number preempt]
user@host# set delay interval
```

2. Set up a limit for preemptive failover.

You can set maximum number of preemptive failovers between 1 to 50 and time period during which the limit is applied between 1 to 1440 seconds.

```
{primary:node1}[edit chassis cluster redundancy-group number preempt]
user@host# set limit limit period period
```

In the following example, you are setting the preemptive delay timer to 300 seconds, and the preemptive limit to 10 for a preemptive period of 600 seconds. That is, this configuration delays immediate failover for 300 seconds, and it limits a maximum of 10 preemptive failovers in a duration of 600 seconds.

```
{primary:node1}[edit chassis cluster redundancy-group 1 preempt]
user@host# set delay 300 limit 10 period 600
```

You can use the `clear chassis clusters preempt-count` command to clear the preempt failover counter for all redundancy groups. When a preempt limit is configured, the counter starts with a first preemptive failover and the count is reduced; this process continues until the count reaches zero before the timer expires. You can use this command to clear the preempt failover counter and reset it to start again.

**SEE ALSO**

[Chassis Cluster Redundancy Groups](#) | 82

## Understanding Chassis Cluster Redundancy Group Manual Failover

You can initiate a redundancy group  $x$  (redundancy groups numbered 1 through 128) failover manually. A manual failover applies until a failback event occurs.

For example, suppose that you manually do a redundancy group 1 failover from node 0 to node 1. Then an interface that redundancy group 1 is monitoring fails, dropping the threshold value of the new primary redundancy group to zero. This event is considered a failback event, and the system returns control to the original redundancy group.

You can also initiate a redundancy group 0 failover manually if you want to change the primary node for redundancy group 0. You cannot enable preemption for redundancy group 0.

If `preempt` is added to a redundancy group configuration, the device with the higher priority in the group can initiate a failover to become primary. By default, preemption is disabled. For more information on preemption, see *preempt (Chassis Cluster)*.

When you do a manual failover for redundancy group 0, the node in the primary state transitions to the secondary-hold state. The node stays in the secondary-hold state for the default or configured time (a minimum of 300 seconds) and then transitions to the secondary state.

State transitions in cases where one node is in the secondary-hold state and the other node reboots, or the control link connection or fabric link connection is lost to that node, are described as follows:

- Reboot case—The node in the secondary-hold state transitions to the primary state; the other node goes dead (inactive).
- Control link failure case—The node in the secondary-hold state transitions to the ineligible state and then to a disabled state; the other node transitions to the primary state.
- Fabric link failure case—The node in the secondary-hold state transitions directly to the ineligible state.

Starting with Junos OS Release 12.1X46-D20 and Junos OS Release 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.

Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.

Keep in mind that during an in-service software upgrade (ISSU), the transitions described here cannot happen. Instead, the other (primary) node transitions directly to the secondary state because Juniper Networks releases earlier than 10.0 do not interpret the secondary-hold state. While you start an ISSU, if one of the nodes has one or more redundancy groups in the secondary-hold state, you must wait for them to move to the secondary state before you can do manual failovers to make all the redundancy groups be primary on one node.

Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine failover, in which case all processes running on the primary node are killed and then spawned on the new primary Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

In some Junos OS releases, for redundancy groups  $x$ , it is possible to do a manual failover on a node that has 0 priority. We recommend that you use the `show chassis cluster status` command to check the redundancy group node priorities before doing the manual failover. However, from Junos OS Releases 12.1X44-D25, 12.1X45-D20, 12.1X46-D10, and 12.1X47-D10 and later, the readiness check mechanism for manual failover is enhanced to be more restrictive, so that you cannot set manual failover to a node in a redundancy group that has 0 priority. This enhancement prevents traffic from being dropped unexpectedly due to a failover attempt to a 0 priority node, which is not ready to accept traffic.

## Initiating a Chassis Cluster Manual Redundancy Group Failover

Before you begin, complete the following tasks:

- ["Example: Configuring Chassis Cluster Redundancy Groups" on page 87](#)
- ["Example: Configuring Chassis Cluster Redundant Ethernet Interfaces" on page 95](#)

You can initiate a failover manually with the `request` command. A manual failover bumps up the priority of the redundancy group for that member to 255.

Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine (RE) failover, in which case all processes running on the primary node are killed and then spawned on the new primary Routing Engine (RE). This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

Unplugging the power cord and holding the power button to initiate a chassis cluster redundancy group failover might result in unpredictable behavior.

For redundancy groups  $x$  (redundancy groups numbered 1 through 128), it is possible to do a manual failover on a node that has 0 priority. We recommend that you check the redundancy group node priorities before doing the manual failover.

Use the `show` command to display the status of nodes in the cluster:

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 0
Cluster ID: 9
Node                Priority      Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              254         primary   no       no
  node1              1          secondary no       no
```

Output to this command indicates that node 0 is primary.

Use the `request` command to trigger a failover and make node 1 primary:

```
{primary:node0}
user@host> request chassis cluster failover redundancy-group 0 node 1
-----
Initiated manual failover for redundancy group 0
```

Use the `show` command to display the new status of nodes in the cluster:

```
{secondary-hold:node0}
user@host> show chassis cluster status redundancy-group 0
Cluster ID: 9
Node                Priority      Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 2
  node0              254         secondary-hold no       yes
  node1              255         primary   no       yes
```

Output to this command shows that node 1 is now primary and node 0 is in the secondary-hold state. After 5 minutes, node 0 will transition to the secondary state.

You can reset the failover for redundancy groups by using the `request` command. This change is propagated across the cluster.

```
{secondary-hold:node0}
user@host> request chassis cluster failover reset redundancy-group 0
node0:
-----
```

```
No reset required for redundancy group 0.
```

```
node1:
```

```
-----  
Successfully reset manual failover for redundancy group 0
```

You cannot trigger a back-to-back failover until the 5-minute interval expires.

```
{secondary-hold:node0}
```

```
user@host> request chassis cluster failover redundancy-group 0 node 0
```

```
node0:
```

```
-----  
Manual failover is not permitted as redundancy-group 0 on node0 is in secondary-hold state.
```

Use the `show` command to display the new status of nodes in the cluster:

```
{secondary-hold:node0}
```

```
user@host> show chassis cluster status redundancy-group 0
```

```
Cluster ID: 9
```

```
Node                Priority      Status      Preempt  Manual failover
```

```
Redundancy group: 0 , Failover count: 2
```

node0	254	secondary-hold	no	no
node1	1	primary	no	no

Output to this command shows that a back-to-back failover has not occurred for either node.

After doing a manual failover, you must issue the `reset failover` command before requesting another failover.

When the primary node fails and comes back up, election of the primary node is done based on regular criteria (priority and preempt).

## Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers

### IN THIS SECTION

- [Requirements | 264](#)
- [Overview | 264](#)
- [Configuration | 264](#)

This example shows how to configure the dampening time between back-to-back redundancy group failovers for a chassis cluster. Back-to-back redundancy group failovers that occur too quickly can cause a chassis cluster to exhibit unpredictable behavior.

### Requirements

Before you begin:

- Understand redundancy group failover. See "[Understanding Chassis Cluster Redundancy Group Failover](#)" on page 255.
- Understand redundancy group manual failover. See "[Understanding Chassis Cluster Redundancy Group Manual Failover](#)" on page 260.

### Overview

The dampening time is the minimum interval allowed between back-to-back failovers for a redundancy group. This interval affects manual failovers and automatic failovers caused by interface monitoring failures.

In this example, you set the minimum interval allowed between back-to-back failovers to 420 seconds for redundancy group 0.

### Configuration

#### IN THIS SECTION

- [Procedure | 265](#)

## Procedure

### Step-by-Step Procedure

To configure the dampening time between back-to-back redundancy group failovers:

1. Set the dampening time for the redundancy group.

```
{primary:node0}[edit]  
user@host# set chassis cluster redundancy-group 0 hold-down-interval 420
```

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

```
{primary:node0}[edit]  
user@host# commit
```

## Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover

Chassis clustering supports SNMP traps, which are triggered whenever there is a redundancy group failover.

The trap message can help you troubleshoot failovers. It contains the following information:

- The cluster ID and node ID
- The reason for the failover
- The redundancy group that is involved in the failover
- The redundancy group's previous state and current state

These are the different states that a cluster can be in at any given instant: hold, primary, secondary-hold, secondary, ineligible, and disabled. Traps are generated for the following state transitions (only a transition from a hold state does not trigger a trap):

- primary <-> secondary
- primary -> secondary-hold
- secondary-hold -> secondary

- secondary -> ineligible
- ineligible -> disabled
- ineligible -> primary
- secondary -> disabled

A transition can be triggered because of any event, such as interface monitoring, SPU monitoring, failures, and manual failovers.

The trap is forwarded over the control link if the outgoing interface is on a node different from the node on the Routing Engine that generates the trap.

You can specify that a trace log be generated by setting the `traceoptions flag snmp` statement.

## Verifying Chassis Cluster Failover Status

### IN THIS SECTION

- [Purpose | 266](#)
- [Action | 266](#)

### Purpose

Display the failover status of a chassis cluster.

### Action

From the CLI, enter the `show chassis cluster status` command:

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 3
  Node name          Priority  Status  Preempt  Manual failover

Redundancy-group: 0, Failover count: 1
node0                254     primary no        no
```



```

node1                2          secondary no    no

Redundancy-group: 1, Failover count: 1
node0                254         primary  no    no
node1                1          secondary no    no

```

```

{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
Node                Priority      Status      Preempt  Manual failover

Redundancy group: 0 , Failover count: 5
  node0              200         primary    no       no
  node1              0          lost       n/a     n/a

Redundancy group: 1 , Failover count: 41
  node0              101        primary    no       no
  node1              0          lost       n/a     n/a

```

```

{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
Node                Priority      Status      Preempt  Manual failover

Redundancy group: 0 , Failover count: 5
  node0              200         primary    no       no
  node1              0          unavailable n/a     n/a

Redundancy group: 1 , Failover count: 41
  node0              101        primary    no       no
  node1              0          unavailable n/a     n/a

```

## Clearing Chassis Cluster Failover Status

To clear the failover status of a chassis cluster, enter the `clear chassis cluster failover-count` command from the CLI:

```
{primary:node1}
user@host> clear chassis cluster failover-count
Cleared failover-count for all redundancy-groups
```

### 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
17.4R1	Starting with Junos OS Release 17.4R1, a failover delay timer is introduced on SRX Series Firewalls in a chassis cluster to limit the flapping of redundancy group state between the secondary and the primary nodes in a preemptive failover.
12.1X47-D10	Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.
12.1X46-D20	Starting with Junos OS Release 12.1X46-D20 and Junos OS Release 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.

### RELATED DOCUMENTATION

[Monitoring of Global-Level Objects in a Chassis Cluster | 181](#)

[Monitoring Chassis Cluster Interfaces | 185](#)

[Monitoring IP Addresses on a Chassis Cluster | 227](#)

# 4

CHAPTER

## Chassis Cluster Operations

---

Aggregated Ethernet Interfaces in a Chassis Cluster | 270

NTP Time Synchronization on Chassis Cluster | 332

Active/Passive Chassis Cluster Deployments | 341

Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster | 407

Example: Configuring an Active/Active Layer 3 Cluster Deployment | 433

Multicast Routing and Asymmetric Routing on Chassis Cluster | 456

Ethernet Switching on Chassis Cluster | 477

Media Access Control Security (MACsec) on Chassis Cluster | 483

Understanding SCTP Behavior in Chassis Cluster | 506

Example: Encrypting Messages Between Two Nodes in a Chassis Cluster | 507

---

# Aggregated Ethernet Interfaces in a Chassis Cluster

## IN THIS SECTION

- [Understanding Link Aggregation Groups in a Chassis Cluster | 270](#)
- [Example: Configuring Link Aggregation Groups in a Chassis Cluster | 272](#)
- [Understanding Link Aggregation Group Failover in a Chassis Cluster | 277](#)
- [Understanding LACP on Chassis Clusters | 279](#)
- [Example: Configuring LACP on Chassis Clusters | 282](#)
- [Example: Configuring Chassis Cluster Minimum Links | 293](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on an SRX5000 Line Device with IOC2 or IOC3 | 296](#)
- [Understanding VRRP on SRX Series Firewalls | 301](#)
- [VRRP failover-delay Overview | 305](#)
- [Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces | 307](#)
- [Example: Configuring VRRP for IPv6 | 320](#)

IEEE 802.3ad link aggregation enables you to group Ethernet interfaces to form a single link layer interface, also known as a link aggregation group (LAG) or bundle. Reth LAG interfaces combine characteristics of reth interfaces and LAG interfaces. For more information, see the following topics:

## Understanding Link Aggregation Groups in a Chassis Cluster

Support for Ethernet link aggregation groups (LAGs) based on IEEE 802.3ad makes it possible to aggregate physical interfaces on a standalone device. LAGs on standalone devices provide increased interface bandwidth and link availability. Aggregation of links in a *chassis cluster* allows a redundant Ethernet interface to add more than two physical child interfaces thereby creating a redundant Ethernet interface LAG. For SRX4600 and SRX5000 line of devices, a redundant Ethernet interface LAG can have up to eight links per redundant Ethernet interface per node (for a total of 16 links per redundant Ethernet interface). For SRX300 Series, SRX1500, SRX1600, SRX2300, SRX4100/SRX4200, and SRX4300 devices, a redundant Ethernet interface LAG can have up to maximum four links per redundant Ethernet interface per node (for a total of 8 links per redundant Ethernet interface).

The aggregated links in a redundant Ethernet interface LAG provide the same bandwidth and redundancy benefits of a LAG on a standalone device with the added advantage of chassis cluster redundancy. A redundant Ethernet interface LAG has two types of simultaneous redundancy. The aggregated links within the redundant Ethernet interface on each node are redundant; if one link in the primary aggregate fails, its traffic load is taken up by the remaining links. If enough child links on the primary node fail, the redundant Ethernet interface LAG can be configured so that all traffic on the entire redundant Ethernet interface fails over to the aggregate link on the other node. You can also configure interface monitoring for LACP-enabled redundancy group reth child links for added protection.

Aggregated Ethernet interfaces, known as local LAGs, are also supported on either node of a chassis cluster but cannot be added to redundant Ethernet interfaces. Local LAGs are indicated in the system interfaces list using an ae- prefix. Likewise any child interface of an existing local LAG cannot be added to a redundant Ethernet interface and vice versa. Note that it is necessary for the switch (or switches) used to connect the nodes in the cluster to have a LAG link configured and 802.3ad enabled for each LAG on both nodes so that the aggregate links are recognized as such and correctly pass traffic. The total maximum number of combined individual node LAG interfaces (ae) and redundant Ethernet (reth) interfaces per cluster is 128.

The redundant Ethernet interface LAG child links from each node in the chassis cluster must be connected to a different LAG at the peer devices. If a single peer switch is used to terminate the redundant Ethernet interface LAG, two separate LAGs must be used in the switch.

Links from different PICs or IOCs and using different cable types (for example, copper and fiber-optic) can be added to the same redundant Ethernet interface LAG but the speed of the interfaces must be the same and all interfaces must be in full duplex mode. We recommend, however, that for purposes of reducing traffic processing overhead, interfaces from the same PIC or IOC be used whenever feasible. Regardless, all interfaces configured in a redundant Ethernet interface LAG share the same virtual MAC address.

SRX Series Firewalls interface-monitoring feature allows monitoring of redundant Ethernet/aggregated Ethernet interfaces.

Redundant Ethernet interface configuration also includes a minimum-links setting that allows you to set a minimum number of physical child links on the primary node in a given redundant Ethernet interface that must be working for the interface to be up. The default minimum-links value is 1. Note that the minimum-links setting only monitors child links on the primary node. Redundant Ethernet interfaces do not use physical interfaces on the backup node for either ingress or egress traffic.

Note the following support details:

- *Quality of service* (QoS) is supported in a redundant Ethernet interface LAG. Guaranteed bandwidth is, however, duplicated across all links. If a link is lost, there is a corresponding loss of guaranteed bandwidth.
- Layer 2 transparent mode and Layer 2 security features are supported in redundant Ethernet interface LAGs.

- Link Aggregation Control Protocol (LACP) is supported in chassis cluster deployments, where aggregated Ethernet interfaces and redundant Ethernet interfaces are supported simultaneously.
- Chassis cluster management, control, and fabric interfaces cannot be configured as redundant Ethernet interface LAGs or added to a redundant Ethernet interface LAG.
- Network processor (NP) bundling can coexist with redundant Ethernet interface LAGs on the same cluster. However, assigning an interface simultaneously to a redundant Ethernet interface LAG and a network processor bundle is not supported.

IOC2 cards do not have network processors but IOC1 cards do have them.

- Single flow throughput is limited to the speed of a single physical link regardless of the speed of the aggregate interface.

On SRX300, SRX320, SRX340, SRX345, and SRX380 devices, the speed mode and link mode configuration is available for member interfaces of a reth interface.

For more information about Ethernet interface link aggregation and LACP, see the “Aggregated Ethernet” information in the [Interfaces User Guide for Security Devices](#).

## SEE ALSO

| [Understanding Link Aggregation Control Protocol](#)

## Example: Configuring Link Aggregation Groups in a Chassis Cluster

### IN THIS SECTION

- [Requirements | 273](#)
- [Overview | 273](#)
- [Configuration | 274](#)
- [Verification | 276](#)

This example shows how to configure a redundant Ethernet interface link aggregation group for a chassis cluster. Chassis cluster configuration supports more than one child interface per node in a redundant Ethernet interface. When at least two physical child interface links from each node are

included in a redundant Ethernet interface configuration, the interfaces are combined within the redundant Ethernet interface to form a redundant Ethernet interface link aggregation group.

## Requirements

Before you begin:

- Configure chassis cluster redundant interfaces. See ["Example: Configuring Chassis Cluster Redundant Ethernet Interfaces" on page 95](#).
- Understand chassis cluster redundant Ethernet interface link aggregation groups. See ["Understanding Link Aggregation Groups in a Chassis Cluster" on page 270](#).

## Overview

For aggregation to take place, the switch used to connect the nodes in the cluster must enable IEEE 802.3ad link aggregation for the redundant Ethernet interface physical child links on each node. Because most switches support IEEE 802.3ad and are also LACP capable, we recommend that you enable LACP on SRX Series Firewalls. In cases where LACP is not available on the switch, you must not enable LACP on SRX Series Firewalls.

In this example, you assign six Ethernet interfaces to reth1 to form the Ethernet interface link aggregation group:

- ge-1/0/1—reth1
- ge-1/0/2—reth1
- ge-1/0/3—reth1
- ge-12/0/1—reth1
- ge-12/0/2—reth1
- ge-12/0/3—reth1

A maximum of eight physical interfaces per node in a cluster, for a total of 16 child interfaces, can be assigned to a single redundant Ethernet interface when a redundant Ethernet interface LAG is being configured.

Junos OS supports LACP and LAG on a redundant Ethernet interface, which is called RLAG.

## Configuration

### IN THIS SECTION

- [Procedure | 274](#)

### 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, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]

set interfaces ge-1/0/1 gigether-options redundant-parent reth1
set interfaces ge-1/0/2 gigether-options redundant-parent reth1
set interfaces ge-1/0/3 gigether-options redundant-parent reth1
set interfaces ge-12/0/1 gigether-options redundant-parent reth1
set interfaces ge-12/0/2 gigether-options redundant-parent reth1
set interfaces ge-12/0/3 gigether-options redundant-parent reth1
```

#### Step-by-Step Procedure

To configure a redundant Ethernet interface link aggregation group:

- Assign Ethernet interfaces to `reth1`.

```
{primary:node0}[edit]
user@host# set interfaces ge-1/0/1 gigether-options redundant-parent reth1
user@host# set interfaces ge-1/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-1/0/3 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/1 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/3 gigether-options redundant-parent reth1
```



## Results

From configuration mode, confirm your configuration by entering the `show interfaces reth1` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show interfaces reth1
...
ge-1/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-1/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-1/0/3 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-12/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-12/0/2 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-12/0/3 {
    gigether-options {
        redundant-parent reth1;
    }
}
...
```

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

## Verification

### IN THIS SECTION

- [Verifying the Redundant Ethernet Interface LAG Configuration | 276](#)

## Verifying the Redundant Ethernet Interface LAG Configuration

### Purpose

Verify the redundant Ethernet interface LAG configuration.

### Action

From operational mode, enter the `show interfaces terse | match reth` command.

```
{primary:node0}
user@host> show interfaces terse | match reth
ge-1/0/1.0          up    down aenet  --> reth1.0
ge-1/0/2.0          up    down aenet  --> reth1.0
ge-1/0/3.0          up    down aenet  --> reth1.0
ge-12/0/1.0         up    down aenet  --> reth1.0
ge-12/0/2.0         up    down aenet  --> reth1.0
ge-12/0/3.0         up    down aenet  --> reth1.0
reth0               up    down
reth0.0             up    down inet    10.10.37.214/24
reth1               up    down
reth1.0             up    down inet
```

### SEE ALSO

| [Example: Configuring Aggregated Ethernet Device with LAG and LACP \(CLI Procedure\)](#)

## Understanding Link Aggregation Group Failover in a Chassis Cluster

### IN THIS SECTION

- [Scenario 1: Monitored Interface Weight Is 255 | 278](#)
- [Scenario 2: Monitored Interface Weight Is 75 | 279](#)
- [Scenario 3: Monitored Interface Weight Is 100 | 279](#)

You control failover of redundant Ethernet (reth) interfaces in two ways:

- Using the `minimum-links` configuration setting. This parameter determines how many physical members of a redundancy group must be up before the group is declared down. By default this parameter is set to one, which means the redundancy group remains active if a single physical interface is up on the primary node.

The default value for minimum links is 1.

- Using the `interface-monitor` configuration statement along with a `weight` value for each member in the LAG. The interface weighting mechanism works by subtracting a failed interface's configured weight from the redundancy group. The group begins with a weight of 255, and when the group falls to, or below 0, the redundancy group is declared down.

**NOTE:** It's worth noting that the `minimum-links` and `interface-monitor` configuration statements work independently. Crossing either the threshold of minimum links (on the primary node), or the threshold of 0 on the redundancy group, triggers a switchover.

In most cases it's a best practice to configure the weights of interface monitoring according to the `minimum-links` setting. This configuration requires that the weights be equally distributed among the monitored links such that when the number of active physical interface links falls below the `minimum-links` setting, the computed weight for that redundancy group also falls to, or below, zero. This triggers a failover of the redundant Ethernet interfaces link aggregation group (LAG) because both the number of physical links falls below the `minimum-links` value and the LAG group's weight falls below 0.

To demonstrate this interaction, consider a `reth0` interface LAG with four underlying physical links:

- The LAG is configured with a `minimum-links` setting of 2. With this setting failover is triggered when the number of active physical links on the primary node is less than 2.

**NOTE:** When the physical link is Up and LACP is Down, a failover of the redundant Ethernet interfaces link aggregation group (LAG) is triggered.

- The Interface-monitor weight values are used to monitor LAG link status and correctly calculate failover weight.

Configure the underlying interface attached to the redundant Ethernet LAG.

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/4 gigether-options redundant-parent reth0
user@host# set interfaces ge-0/0/5 gigether-options redundant-parent reth0
user@host# set interfaces ge-0/0/6 gigether-options redundant-parent reth0
user@host# set interfaces ge-0/0/7 gigether-options redundant-parent reth0
```

Specify the minimum number of links for the redundant Ethernet interface as 2.

```
{primary:node0}[edit]
user@host# set interfaces reth0 redundant-ether-options minimum-links 2
```

Configure interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.

These scenarios provide examples of how redundant Ethernet LAG failover operates:

### Scenario 1: Monitored Interface Weight Is 255

Specify the monitored interface weight as 255 for each underlying interface.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/4 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/5 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/6 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/7 weight 255
```

When 1 of the 4 interfaces fails there are still 3 active physical links in the redundant Ethernet LAG. While this number exceeds the configured minimum links parameter, the loss of one interface with a weight of 255 causes the group's weight to fall to 0, triggering a failover.

## Scenario 2: Monitored Interface Weight Is 75

Specify the monitored interface weight as 75 for each underlying interface.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/4 weight 75
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/5 weight 75
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/6 weight 75
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/7 weight 75
```

In this case, when three physical links are down, the redundant Ethernet interface will go down due to falling below the `minimum-links` value configured. Note that in this scenario the LAG group weight remains above 0.

## Scenario 3: Monitored Interface Weight Is 100

Specify the monitored interface weight as 100 for each underlying interface.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/4 weight 100
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/5 weight 100
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/6 weight 100
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/7 weight 100
```

In this case, when 3 of 4 physical links are down, the redundant Ethernet interface is declared down both because the `minimum-links` value is not met, and due to the interface monitoring weights causing the LAG group's weight to reach 0.

Of all the three scenarios, scenario 3 illustrates the most ideal way to manage redundant Ethernet LAG failover and there will be minimum traffic loss.

## Understanding LACP on Chassis Clusters

### IN THIS SECTION

- [Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups | 280](#)
- [Sub-LAGs | 281](#)

- [Supporting Hitless Failover | 281](#)
- [Managing Link Aggregation Control PDUs | 282](#)

You can combine multiple physical Ethernet ports to form a logical point-to-point link, known as a link aggregation group (LAG) or bundle, such that a media access control (MAC) client can treat the LAG as if it were a single link.

LAGs can be established across nodes in a *chassis cluster* to provide increased interface bandwidth and link availability.

The Link Aggregation Control Protocol (LACP) provides additional functionality for LAGs. LACP is supported in standalone deployments, where aggregated Ethernet interfaces are supported, and in chassis cluster deployments, where aggregated Ethernet interfaces and redundant Ethernet interfaces are supported simultaneously.

You configure LACP on a redundant Ethernet interface by setting the LACP mode for the parent link with the `lacp` statement. The LACP mode can be off (the default), active, or passive.

This topic contains the following sections:

## Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

A redundant Ethernet interface has active and standby links located on two nodes in a chassis cluster. All active links are located on one node, and all standby links are located on the other node. You can configure up to eight active links and eight standby links per node.

When at least two physical child interface links from each node are included in a redundant Ethernet interface configuration, the interfaces are combined within the redundant Ethernet interface to form a redundant Ethernet interface LAG.

Having multiple active redundant Ethernet interface links reduces the possibility of failover. For example, when an active link is out of service, all traffic on this link is distributed to other active redundant Ethernet interface links, instead of triggering a redundant Ethernet active/standby failover.

Aggregated Ethernet interfaces, known as local LAGs, are also supported on either node of a chassis cluster but cannot be added to redundant Ethernet interfaces. Likewise, any child interface of an existing local LAG cannot be added to a redundant Ethernet interface, and vice versa. The total maximum number of combined individual node LAG interfaces (`ae`) and redundant Ethernet (`reth`) interfaces per cluster is 128.

However, aggregated Ethernet interfaces and redundant Ethernet interfaces can coexist, because the functionality of a redundant Ethernet interface relies on the Junos OS aggregated Ethernet framework.

For more information, see ["Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups"](#) on page 270.

## Minimum Links

Redundant Ethernet interface configuration includes a `minimum-links` setting that allows you to set a minimum number of physical child links in a redundant Ethernet interface LAG that must be working on the primary node for the interface to be up. The default `minimum-links` value is 1. When the number of physical links on the primary node in a redundant Ethernet interface falls below the `minimum-links` value, the interface might be down even if some links are still working. For more information, see ["Example: Configuring Chassis Cluster Minimum Links"](#) on page 293.

## Sub-LAGs

LACP maintains a point-to-point LAG. Any port connected to the third point is denied. However, a redundant Ethernet interface does connect to two different systems or two remote aggregated Ethernet interfaces by design.

To support LACP on redundant Ethernet interface active and standby links, a redundant Ethernet interface is created automatically to consist of two distinct sub-LAGs, where all active links form an active sub-LAG and all standby links form a standby sub-LAG.

In this model, LACP selection logic is applied and limited to one sub-LAG at a time. In this way, two redundant Ethernet interface sub-LAGs are maintained simultaneously while all the LACP advantages are preserved for each sub-LAG.

It is necessary for the switches used to connect the nodes in the cluster to have a LAG link configured and 802.3ad enabled for each LAG on both nodes so that the aggregate links are recognized as such and correctly pass traffic.

The redundant Ethernet interface LAG child links from each node in the chassis cluster must be connected to a different LAG at the peer devices. If a single peer switch is used to terminate the redundant Ethernet interface LAG, two separate LAGs must be used in the switch.

## Supporting Hitless Failover

With LACP, the redundant Ethernet interface supports hitless failover between the active and standby links in normal operation. The term *hitless* means that the redundant Ethernet interface state remains up during a failover.

The `lacpd` process manages both the active and standby links of the redundant Ethernet interfaces. A redundant Ethernet interface state remains up when the number of active up links is equal to or more than the number of minimum links configured. Therefore, to support hitless failover, the LACP state on the redundant Ethernet interface standby links must be collected and distributed before failover occurs.

## Managing Link Aggregation Control PDUs

The protocol data units (PDUs) contain information about the state of the link. By default, aggregated and redundant Ethernet links do not exchange link aggregation control PDUs.

You can configure PDUs exchange in the following ways:

- Configure Ethernet links to actively transmit link aggregation control PDUs
- Configure Ethernet links to passively transmit PDUs, sending out link aggregation control PDUs only when they are received from the remote end of the same link

The local end of a child link is known as the actor and the remote end of the link is known as the partner. That is, the actor sends link aggregation control PDUs to its protocol partner that convey what the actor knows about its own state and that of the partner's state.

You configure the interval at which the interfaces on the remote side of the link transmit link aggregation control PDUs by configuring the `periodic` statement on the interfaces on the local side. It is the configuration on the local side that specifies the behavior of the remote side. That is, the remote side transmits link aggregation control PDUs at the specified interval. The interval can be `fast` (every second) or `slow` (every 30 seconds).

For more information, see ["Example: Configuring LACP on Chassis Clusters" on page 282](#).

By default, the actor and partner transmit link aggregation control PDUs every second. You can configure different periodic rates on active and passive interfaces. When you configure the active and passive interfaces at different rates, the transmitter honors the receiver's rate.

## Example: Configuring LACP on Chassis Clusters

### IN THIS SECTION

- [Requirements | 283](#)
- [Overview | 283](#)
- [Configuration | 284](#)
- [Verification | 290](#)

This example shows how to configure LACP on chassis clusters.



## Requirements

Before you begin:

Complete the tasks such as enabling the chassis cluster, configuring interfaces and redundancy groups. See ["SRX Series Chassis Cluster Configuration Overview" on page 14](#) and ["Example: Configuring Chassis Cluster Redundant Ethernet Interfaces" on page 95](#) for more details.

## Overview

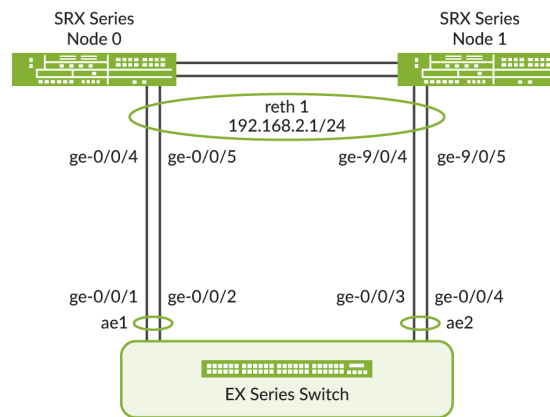
You can combine multiple physical Ethernet ports to form a logical point-to-point link, known as a link aggregation group (LAG) or bundle. You configure LACP on a redundant Ethernet interface of SRX Series Firewall in chassis cluster.

In this example, you set the LACP mode for the reth1 interface to active and set the link aggregation control PDU transmit interval to slow, which is every 30 seconds.

When you enable LACP, the local and remote sides of the aggregated Ethernet links exchange protocol data units (PDUs), which contain information about the state of the link. You can configure Ethernet links to actively transmit PDUs, or you can configure the links to passively transmit them (sending out LACP PDUs only when they receive them from another link). One side of the link must be configured as active for the link to be up.

[Figure 38 on page 283](#) shows the topology used in this example.

**Figure 38: Topology for LAGs Connecting SRX Series Firewalls in Chassis Cluster to an EX Series Switch**



In the [Figure 38 on page 283](#), SRX1500 devices are used to configure the interfaces on node0 and node1. For more information on EX Series switch configuration, see [Configuring Aggregated Ethernet LACP \(CLI Procedure\)](#).

## Configuration

### IN THIS SECTION

- [Configuring LACP on Chassis Cluster | 284](#)
- [Configuring LACP on EX Series Switch | 286](#)

### Configuring LACP on Chassis Cluster

#### Step-by-Step Procedure

To configure LACP on chassis clusters:

1. Specify the number of redundant Ethernet interfaces.

```
[edit chassis cluster]
user@host# set reth-count 2
```

2. Specify a redundancy group's priority for primacy on each node of the cluster. The higher number takes precedence.

```
[edit chassis cluster]
user@host# set redundancy-group 1 node 0 priority 200
user@host# set redundancy-group 1 node 1 priority 100
```

3. Create security zone and assign interfaces to zone.

```
[edit security zones]
user@host# set security-zone trust host-inbound-traffic system-services all
user@host# set security-zone trust interfaces reth1.0
```

4. Bind redundant child physical interfaces to reth1.

```
[edit interfaces]
user@host# set ge-0/0/4 gigheter-options redundant-parent reth1
user@host# set ge-0/0/5 gigheter-options redundant-parent reth1
```

```
user@host# set ge-9/0/4 gicther-options redundant-parent reth1
user@host# set ge-9/0/5 gicther-options redundant-parent reth1
```

5. Add reth1 to redundancy group 1.

```
[edit interfaces]
user@host# set reth1 redundant-ether-options redundancy-group 1
```

6. Set the LACP on reth1.

```
[edit interfaces]
user@host# set reth1 redundant-ether-options lacp active
user@host# set reth1 redundant-ether-options lacp periodic slow
```

7. Assign an IP address to reth1.

```
[edit interfaces]
user@host# set reth1 unit 0 family inet address 192.168.2.1/24
```

8. Configure LACP on aggregated Ethernet interfaces (ae1).
9. Configure LACP on aggregated Ethernet interfaces (ae2).
10. If you are done configuring the device, commit the configuration.

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

## Results

From configuration mode, confirm your configuration by entering the `show chassis`, `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.

```
[edit]
user@host#show chassis
cluster {
    reth-count 2;
```

```
    redundancy-group 1 {
        node 0 priority 200;
        node 1 priority 100;
    }
}
[edit]
user@host#show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        reth1.0;
    }
}
[edit]
user@host#show interfaces
reth1 {
    redundant-ether-options {
        redundancy-group 1;
        lacp {
            active;
            periodic slow;
        }
    }
    unit 0 {
        family inet {
            address 192.168.2.1/24;
        }
    }
}
```

## Configuring LACP on EX Series Switch

### Step-by-Step Procedure

Configure LACP on EX Series switch.

1. Set the number of aggregated Ethernet interfaces.

```
[edit chassis]
user@host# set aggregated-devices ethernet device-count 3
```

2. Associate physical interfaces with aggregated Ethernet interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 gigether-options 802.3ad ae1
user@host# set ge-0/0/2 gigether-options 802.3ad ae1
user@host# set ge-0/0/3 gigether-options 802.3ad ae2
user@host# set ge-0/0/4 gigether-options 802.3ad ae2
```

3. Configure LACP on aggregated Ethernet interfaces (ae1).

```
[edit interfaces]
user@host# set interfaces ae1 unit 0 family ethernet-switching interface-mode access

user@host# set interfaces ae1 unit 0 family ethernet-switching vlan members RETH0_VLAN
```

4. Configure LACP on aggregated Ethernet interfaces (ae2).

```
[edit interfaces]

user@host# set interfaces ae2 unit 0 family ethernet-switching interface-mode access

user@host# set interfaces ae2 unit 0 family ethernet-switching vlan members RETH0_VLAN
```

5. Configure VLAN.

```
user@host#set vlans RETH0_VLAN vlan-id 10
user@host# set vlans RETH0_VLAN l3-interface vlan.10
user@host# set interfaces vlan unit 10 family inet address 192.168.2.254/24
```

## Results

From configuration mode, confirm your configuration by entering the `show chassis` and `show interfaces` 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
aggregated-devices {
  ethernet {
    device-count 3;
  }
}
user@host# show vlans
RETH0_VLAN {
  vlan-id 10;
  l3-interface vlan.10;
}
user@host> show vlans RETH0_VLAN
Routing instance      VLAN name      Tag      Interfaces
default-switch       RETH0_VLAN    10
                                                              ae1.0*
                                                              ae2.0*

user@host> show ethernet-switching interface ae1
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                        LH - MAC limit hit, DN - interface down,
                        MMAS - Mac-move action shutdown,
                        SCTL - shutdown by Storm-control )
Logical      Vlan      TAG      MAC      STP      Logical      Tagging
interface    members
ae1.0
                RETH0_VLAN  10      131072   Forwarding
                interface flags
                untagged
                untagged

user@host> show ethernet-switching interface ae2
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                        LH - MAC limit hit, DN - interface down,
                        MMAS - Mac-move action shutdown,
                        SCTL - shutdown by Storm-control )

Logical      Vlan      TAG      MAC      STP      Logical      Tagging
interface    members
                interface flags
```

```

ae2.0          131072          untagged
                RETH0_VLAN  10    131072    Forwarding    untagged

```

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

```

ge-0/0/1 {
  ether-options {
    802.3ad ae1;
  }
}
ge-0/0/2 {
  ether-options {
    802.3ad ae1;
  }
}
ge-0/0/3 {
  ether-options {
    802.3ad ae2;
  }
}
ge-0/0/4 {
  ether-options {
    802.3ad ae2;
  }
}
ae1 {
  aggregated-ether-options {
    lacp {
      active;
      periodic slow;
    }
  }
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members RETH0_VLAN;
      }
    }
  }
}
ae2 {
  aggregated-ether-options {
    lacp {
      active;

```

```
        periodic slow;
    }
}
unit 0 {
    family ethernet-switching {
        interface-mode access;
        vlan {
            members RETH0_VLAN;
        }
    }
}
vlan {
    unit 10 {
        family inet {
            address 192.168.2.254/24 {
            }
        }
    }
}
}
```

## Verification

### IN THIS SECTION

- [Verifying LACP on Redundant Ethernet Interfaces | 290](#)

## Verifying LACP on Redundant Ethernet Interfaces

### Purpose

Display LACP status information for redundant Ethernet interfaces.



## Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}[edit]
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring    RE Relinquish monitoring
  IS IRQ storm

Cluster ID: 1
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 1      primary          no    no    None
node1 1      secondary        no    no    None

Redundancy group: 1 , Failover count: 1
node0 200    primary          no    no    None
node1 100    secondary        no    no    None
```

```
{primary:node0}[edit]
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      fxp1      Up                Disabled     Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
              (Physical/Monitored)
  fab0    ge-0/0/2        Up / Up
              Security
              Enabled
```

```

fab0
fab1    ge-9/0/2        Up    / Up        Enabled
fab1

```

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Down	Not configured
reth1	Up	1

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

From operational mode, enter the `show lacp interfaces reth1` command.

```

{primary:node0}[edit]
user@host> show lacp interfaces reth1
Aggregated interface: reth1
LACP state:      Role  Exp  Def  Dist  Col  Syn  Aggr  Timeout  Activity
  ge-0/0/4       Actor No   No   Yes  Yes  Yes  Yes    Slow    Active
  ge-0/0/4       Partner No   No   Yes  Yes  Yes  Yes    Slow    Active
  ge-0/0/5       Actor No   No   Yes  Yes  Yes  Yes    Slow    Active
  ge-0/0/5       Partner No   No   Yes  Yes  Yes  Yes    Slow    Active
  ge-9/0/4       Actor No   No   Yes  Yes  Yes  Yes    Slow    Active
  ge-9/0/4       Partner No   No   Yes  Yes  Yes  Yes    Slow    Active
  ge-9/0/5       Actor No   No   Yes  Yes  Yes  Yes    Slow    Active
  ge-9/0/5       Partner No   No   Yes  Yes  Yes  Yes    Slow    Active
LACP protocol:  Receive State  Transmit State  Mux State
  ge-0/0/4      Current      Slow periodic  Collecting distributing
  ge-0/0/5      Current      Slow periodic  Collecting distributing
  ge-9/0/4      Current      Slow periodic  Collecting distributing
  ge-9/0/5      Current      Slow periodic  Collecting distributing

```

The output shows redundant Ethernet interface information, such as the following:

- The LACP state—Indicates whether the link in the bundle is an actor (local or near-end of the link) or a partner (remote or far-end of the link).
- The LACP mode—Indicates whether both ends of the aggregated Ethernet interface are enabled (active or passive)—at least one end of the bundle must be active.
- The periodic link aggregation control PDU transmit rate.

- The LACP protocol state—Indicates the link is up if it is collecting and distributing packets.

## Example: Configuring Chassis Cluster Minimum Links

### IN THIS SECTION

- Requirements | 293
- Overview | 293
- Configuration | 294
- Verification | 294

This example shows how to specify a minimum number of physical links assigned to a redundant Ethernet interface on the primary node that must be working for the interface to be up.

### Requirements

Before you begin:

- Configure redundant Ethernet interfaces. See ["Example: Configuring Chassis Cluster Redundant Ethernet Interfaces" on page 95](#).
- Understand redundant Ethernet interface link aggregation groups. See ["Example: Configuring Link Aggregation Groups in a Chassis Cluster" on page 272](#).

### Overview

When a redundant Ethernet interface has more than two child links, you can set a minimum number of physical links assigned to the interface on the primary node that must be working for the interface to be up. When the number of physical links on the primary node falls below the minimum-links value, the interface will be down even if some links are still working.

In this example, you specify that three child links on the primary node and bound to reth1 (minimum-links value) be working to prevent the interface from going down. For example, in a redundant Ethernet interface LAG configuration in which six interfaces are assigned to reth1, setting the minimum-links value to 3 means that all reth1 child links on the primary node must be working to prevent the interface's status from changing to down.

Although it is possible to set a minimum-links value for a redundant Ethernet interface with only two child interfaces (one on each node), we do not recommend it.

## Configuration

### IN THIS SECTION

- [Procedure | 294](#)

## Procedure

### Step-by-Step Procedure

To specify the minimum number of links:

1. Specify the minimum number of links for the redundant Ethernet interface.

```
{primary:node0}[edit]  
user@host# set interfaces reth1 redundant-ether-options minimum-links 3
```

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

```
{primary:node0}[edit]  
user@host# commit
```

## Verification

### IN THIS SECTION

- [Verifying the Chassis Cluster Minimum Links Configuration | 295](#)

## Verifying the Chassis Cluster Minimum Links Configuration

### Purpose

To verify the configuration is working properly, enter the `show interface reth1` command.

### Action

From operational mode, enter the `show show interfaces reth1` command.

```
{primary:node0}[edit]
user@host> show interfaces reth1
Physical interface: reth1, Enabled, Physical link is Down
  Interface index: 129, SNMP ifIndex: 548
  Link-level type: Ethernet, MTU: 1514, Speed: Unspecified, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled, Minimum links needed: 3, Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Current address: 00:10:db:ff:10:01, Hardware address: 00:10:db:ff:10:01
  Last flapped   : 2010-09-15 15:54:53 UTC (1w0d 22:07 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)

Logical interface reth1.0 (Index 68) (SNMP ifIndex 550)
  Flags: Hardware-Down Device-Down SNMP-Traps 0x0 Encapsulation: ENET2
  Statistics          Packets      pps      Bytes      bps
  Bundle:
    Input :           0           0           0           0
    Output:           0           0           0           0
  Security: Zone: untrust
  Allowed host-inbound traffic : bootp bfd bgp dns dvmrp igmp ldp msdp nhrp
  ospf pgm pim rip router-discovery rsvp sap vrrp dhcp finger ftp tftp
  ident-reset http https ike netconf ping reverse-telnet reverse-ssh rlogin
  rpm rsh snmp snmp-trap ssh telnet traceroute xnm-clear-text xnm-ssl lsping
  ntp sip
  Protocol inet, MTU: 1500
  Flags: Sendbcst-pkt-to-re
```

## Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on an SRX5000 Line Device with IOC2 or IOC3

### IN THIS SECTION

- Requirements | 296
- Overview | 296
- Configuration | 297
- Verification | 300

Support for Ethernet link aggregation groups (LAGs) based on IEEE 802.3ad makes it possible to aggregate physical interfaces on a standalone device. LAGs on standalone devices provide increased interface bandwidth and link availability. Aggregation of links in a chassis cluster allows a redundant Ethernet interface to add more than two physical child interfaces, thereby creating a redundant Ethernet interface LAG.

### Requirements

This example uses the following software and hardware components:

- Junos OS Release 15.1X49-D40 or later for SRX Series Firewalls.
- SRX5800 with IOC2 or IOC3 with Express Path enabled on IOC2 and IOC3. For details, see [Example: Configuring SRX5K-MPC3-100G10G \(IOC3\) and SRX5K-MPC3-40G10G \(IOC3\) on an SRX5000 Line Device to Support Express Path](#).

### Overview

This example shows how to configure a redundant Ethernet interface link aggregation group and configure LACP on chassis clusters on an SRX Series Firewall using the ports from either IOC2 or IOC3 in Express Path mode. Note that configuring child interfaces by mixing links from both IOC2 and IOC3 is not supported.

A redundant Ethernet interface or aggregated Ethernet interface (aex) must contain child interfaces from the same IOC type for IOC2 and IOC3. For example, if one child link is from 10-Gigabit Ethernet on IOC2, the second child link should also be from IOC2. This limitation is not applicable for IOC3 and IOC4 child interfaces if the child interfaces have the same speed.

The following combination is not supported:

- Node 0-100GbE from IOC2 and 10GbE/40GbE/100GbE from IOC3
- Node 1-100GbE from IOC2 and 10GbE/40GbE/100GbE from IOC3

The following combination is supported (with the same interface speed):

- Node 0-100GbE from IOC3 and 100GbE from IOC4
- Node 1-100GbE from IOC3 and 100GbE from IOC4

The following member links are used in this example:

- xe-1/0/0
- xe-3/0/0
- xe-14/0/0
- xe-16/0/0

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 297](#)
- [Procedure | 298](#)

### 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, delete, and then copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set chassis cluster reth-count 5
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 redundant-ether-options lacp active
set interfaces reth0 redundant-ether-options lacp periodic fast
set interfaces reth0 redundant-ether-options minimum-links 1
set interfaces reth0 unit 0 family inet address 192.0.2.1/24
set interfaces xe-1/0/0 gigheter-options redundant-parent reth0
set interfaces xe-3/0/0 gigheter-options redundant-parent reth0
```

```
set interfaces xe-14/0/0 gigether-options redundant-parent reth0
set interfaces xe-16/0/0 gigether-options redundant-parent reth0
```

## Procedure

### Step-by-Step Procedure

To configure LAG Interfaces:

1. Specify the number of aggregated Ethernet interfaces to be created.

```
[edit chassis]
user@host# set chassis cluster reth-count 5
```

2. Bind redundant child physical interfaces to reth0.

```
[edit interfaces]
user@host# set xe-1/0/0 gigether-options redundant-parent reth0
user@host# set xe-3/0/0 gigether-options redundant-parent reth0
user@host# set xe-14/0/0 gigether-options redundant-parent reth0
user@host# set xe-16/0/0 gigether-options redundant-parent reth0
```

3. Add reth0 to redundancy group 1.

```
user@host#set reth0 redundant-ether-options redundancy-group 1
```

4. Assign an IP address to reth0.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.1/24
```

5. Set the LACP on reth0.

```
[edit interfaces]
user@host# set reth0 redundant-ether-options lacp active
user@host# set reth0 redundant-ether-options lacp periodic fast
user@host# set reth0 redundant-ether-options minimum-links 1
```



## Results

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

```
[edit]
user@host# show interfaces
xe-1/0/0 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-3/0/0 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-14/0/0 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-16/0/0 {
  gigether-options {
    redundant-parent reth0;
  }
}
reth0 {
  redundant-ether-options {
    lacp {
      active;
      periodic fast;
    }
    minimum-links 1;
  }
  unit 0 {
    family inet {
      address 192.0.2.1/24;
    }
  }
}
```

```
ae1 {
  aggregated-ether-options {
    lacp {
      active;
    }
  }
  unit 0 {
    family inet {
      address 192.0.2.2/24;
    }
  }
}
```

```
[edit]
user@host# show chassis
chassis cluster {
  reth-count 5;
}
```

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

## Verification

### IN THIS SECTION

- [Verifying LACP on Redundant Ethernet Interfaces | 300](#)

## Verifying LACP on Redundant Ethernet Interfaces

### Purpose

Display LACP status information for redundant Ethernet interfaces.

## Action

From operational mode, enter the `show lacp interfaces` command to check that LACP has been enabled as active on one end.

```
user@host> show lacp interfaces
```

```
Aggregated interface: reth0
```

LACP state:	Role	Exp	Def	Dist	Col	Syn	Aggr	Timeout	Activity
xe-16/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-16/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-14/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-14/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-1/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-1/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-3/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-3/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active

LACP protocol:	Receive State	Transmit State	Mux State
xe-16/0/0	Current	Fast periodic	Collecting distributing
xe-14/0/0	Current	Fast periodic	Collecting distributing
xe-1/0/0	Current	Slow periodic	Collecting distributing
xe-3/0/0	Current	Slow periodic	Collecting distributing

The output indicates that LACP has been set up correctly and is active at one end.

## Understanding VRRP on SRX Series Firewalls

### IN THIS SECTION

- [Overview of VRRP on SRX Series Firewalls | 302](#)
- [Benefits of VRRP | 303](#)
- [Sample VRRP Topology | 303](#)
- [SRX Series Firewalls Support for VRRPv3 | 304](#)
- [Limitations of VRRPv3 Features | 304](#)

SRX Series Firewalls support the Virtual Router Redundancy Protocol (VRRP) and VRRP for IPv6. This topic covers:

## Overview of VRRP on SRX Series Firewalls

Configuring end hosts on your network with static default routes minimizes configuration effort and complexity and reduces processing overhead on the end hosts. When hosts are configured with static routes, the failure of the default gateway normally results in a catastrophic event, isolating all hosts that are unable to detect available alternate paths to their gateway. Using Virtual Router Redundancy Protocol (VRRP) enables you to dynamically provide alternative gateways for end hosts if the primary gateway fails.

You can configure the Virtual Router Redundancy Protocol (VRRP) or VRRP for IPv6 on Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, and logical interfaces on SRX Series Firewalls. VRRP enables hosts on a LAN to make use of redundant devices on that LAN without requiring more than the static configuration of a single default route on the hosts. Devices configured with VRRP share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP configured devices is the primary (active) and the others are backups. If the primary device fails, then one of the backup devices becomes the new primary, providing a virtual default device and enabling traffic on the LAN to be routed without relying on a single device. Using VRRP, a backup SRX Series Firewall can take over a failed default device within a few seconds. This is done with minimum loss of VRRP traffic and without any interaction with the hosts. Virtual Router Redundancy Protocol is not supported on management interfaces.

VRRP for IPv6 provides a much faster switchover to an alternate default device than IPv6 Neighbor Discovery (ND) procedures. VRRP for IPv6 does not support the authentication-type or authentication-key statements.

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 the backup device at a regular intervals. The default interval is 1 second. If the backup device do not receive an advertisement for a set period, then the backup device with the highest priority takes over as primary and begins forwarding packets.

The backup devices do not attempt to preempt the primary device 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 device becoming primary device of any device associated with addresses it owns.

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.

Priority 255 cannot be set for routed VLAN interfaces (RVIs).

VRRP is defined in RFC 3768, *Virtual Router Redundancy Protocol*.

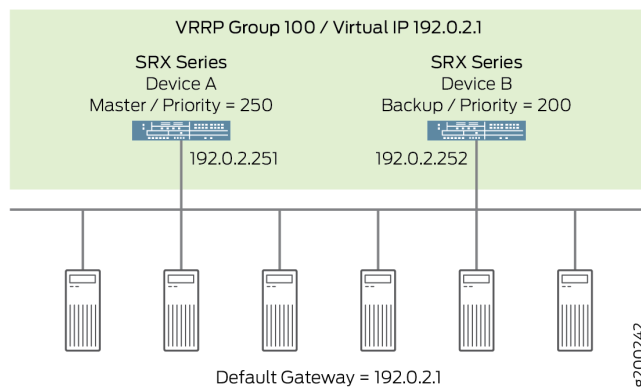
## Benefits of VRRP

- VRRP provides dynamic failover of IP addresses from one device to another in the event of failure.
- You can implement VRRP to provide a highly available default path to a gateway without needing to configure dynamic routing or router discovery protocols on end hosts.

## Sample VRRP Topology

Figure 39 on page 303 illustrates a basic VRRP topology with SRX Series Firewalls. In this example, Devices A and B are running VRRP and share the virtual IP address 192.0.2.1. The default gateway for each of the clients is 192.0.2.1.

Figure 39: Basic VRRP on SRX Series Switches



The following illustrates basic VRRP behavior using Figure 39 on page 303 for reference:

1. When any of the servers wants to send traffic out of the LAN, it sends the traffic to the default gateway address of 192.0.2.1. This is a virtual IP address (VIP) owned by VRRP group 100. Because Device A is the primary of the group, the VIP is associated with the “real” address 192.0.2.251 on Device A, and traffic from the servers is actually sent to this address. (Device A is the primary because it has been configured with a higher priority value.)
2. If there is a failure on Device A that prevents it from forwarding traffic to or from the servers—for example, if the interface connected to the LAN fails—Device B becomes the primary and assumes ownership of the VIP. The servers continue to send traffic to the VIP, but because the VIP is now associated with the “real” address 192.0.2.252 on Device B (because of change of primary), the traffic is sent to Device B instead of Device A.

3. If the problem that caused the failure on Device A is corrected, Device A becomes the primary again and reasserts ownership of the VIP. In this case, the servers resume sending traffic to Device A.

Notice that no configuration changes are required on the servers for them to switch between sending traffic to Device A and Device B. When the VIP moves between 192.0.2.251 and 192.0.2.252, the change is detected by normal TCP-IP behavior and no configuration or intervention is required on the servers.

## SRX Series Firewalls Support for VRRPv3

The advantage of using VRRPv3 is that VRRPv3 supports both IPv4 and IPv6 address families, whereas VRRP supports only IPv4 addresses.

Enable VRRPv3 in your network only if VRRPv3 can be enabled on all the devices configured with VRRP in your network because VRRPv3 (IPv4) does not interoperate with the previous versions of VRRP. For example, if VRRP IPv4 advertisement packets are received by a device on which VRRPv3 is enabled, then the device 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 devices on the LAN.

## Limitations of VRRPv3 Features

Below are some VRRPv3 features limitations.

### 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).

## SEE ALSO

[Junos OS High Availability Configuration Guide](#)

`show vrrp`

## VRRP failover-delay Overview

### IN THIS SECTION

- [When failover-delay Is Not Configured | 306](#)
- [When failover-delay Is Configured | 306](#)

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 from 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 milliseconds—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 milliseconds.
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.

## SEE ALSO

| [failover-delay](#)

## Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces

### IN THIS SECTION

- [Requirements | 308](#)
- [Overview | 308](#)
- [Configuration VRRP | 309](#)

- [Verification | 317](#)

When Virtual Router Redundancy Protocol (VRRP) is configured, the VRRP groups multiple devices into a virtual device. At any time, one of the devices configured with VRRP is the primary (active) and the other devices are backups. If the primary fails, one of the backup devices becomes the new primary device.

This example describes how to configure VRRP on redundant interface:

## Requirements

This example uses the following hardware and software components:

- Junos OS Release 18.1 R1 or later for SRX Series Firewalls.
- Two SRX Series Firewalls connected in a chassis cluster.
- One SRX Series Firewall connected as standalone device.

## Overview

### IN THIS SECTION

- [Topology | 309](#)

You configure VRRP by configuring VRRP groups on redundant interfaces on a chassis cluster devices and on Gigabit Ethernet interface on standalone device. A redundant interface of chassis cluster devices and Gigabit Ethernet interface of standalone device can be a member of one or more VRRP groups. Within a VRRP group, the primary redundant interface of chassis cluster devices and the backup Gigabit Ethernet interface of standalone device must be configured.

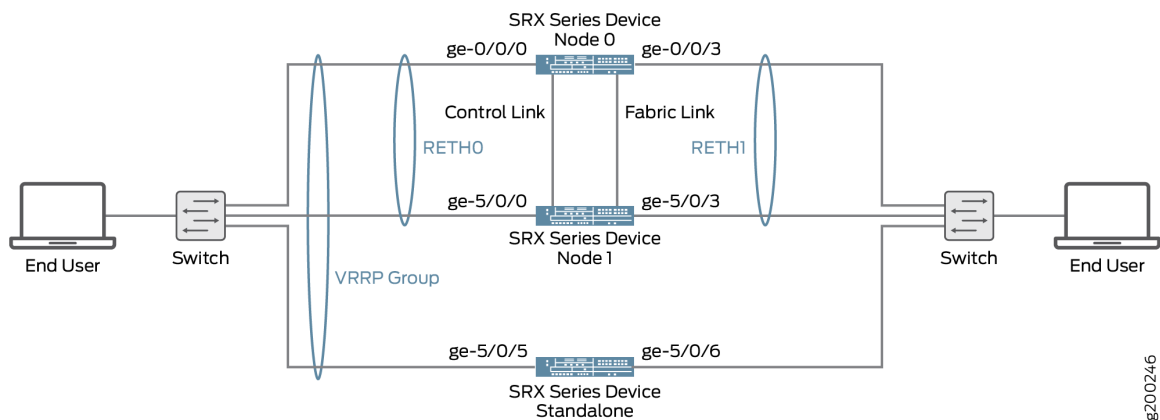
To configure VRRP group, you must configure group identifier, and virtual IP address to the redundant interfaces and Gigabit Ethernet interfaces that are members of VRRP group. The virtual IP address must be the same for all the interfaces in the VRRP group. Then you configure the priority to the redundant interfaces and Gigabit Ethernet interfaces to become the primary interface.

You can force assignment of primary and backup redundant interfaces and Gigabit Ethernet interfaces using priorities from 1 through 255, where 255 is the highest priority.

## Topology

Figure 40 on page 309 shows the topology used in this example.

Figure 40: VRRP on Redundant interface



## Configuration VRRP

### IN THIS SECTION

- [Configuring VRRPv3, VRRP Groups, and Priority on Chassis Cluster Redundant Ethernet Interfaces | 309](#)
- [Configuring VRRP Groups on Standalone Device | 314](#)

### Configuring VRRPv3, VRRP Groups, and Priority on Chassis Cluster Redundant Ethernet Interfaces

#### 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.

```
set protocols vrrp traceoptions file vrrp.log
set protocols vrrp traceoptions file size 10000000
set protocols vrrp traceoptions flag all
```

```

set protocols vrrp version-3
set protocols vrrp ignore-nonstop-routing
set interfaces ge-0/0/0 gigether-options redundant-parent reth0
set interfaces ge-0/0/3 gigether-options redundant-parent reth1
set interfaces ge-5/0/0 gigether-options redundant-parent reth0
set interfaces ge-5/0/3 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 virtual-address 192.0.2.3
set interfaces reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 priority 255
set interfaces reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 accept-data
set interfaces reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 virtual-inet6-address
2001:db8::3
set interfaces reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 priority 255
set interfaces reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 accept-data
set interfaces reth1 redundant-ether-options redundancy-group 2
set interfaces reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 virtual-address 192.168.120.3
set interfaces reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 priority 150
set interfaces reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 accept-data
set interfaces reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3 virtual-inet6-address
2001:db8::4
set interfaces reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3 priority 150
set interfaces reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3 accept-data

```

## Step-by-Step Procedure

To configure VRRPv3, VRRP Groups, and priority on chassis cluster devices:

1. Configure a filename to the traceoptions to trace VRRP protocol traffic.

```

[edit protocols vrrp]
user@host# set traceoptions file vrrp.log

```

2. Specify the maximum trace file size.

```

[edit protocols vrrp]
user@host# set traceoptions file size 10000000

```

3. Enable vrrp traceoptions.

```
[edit protocols vrrp]
user@host# set traceoptions flag all
```

4. Set vrrp version to 3.

```
[edit protocols vrrp]
user@host# set version-3
```

5. Configure this command to support graceful Routing Engine switchover (GRES) for VRRP and for nonstop active routing when there is VRRP reth failover. Using vrrp, a secondary node can take over a failed primary node within a few seconds and this is done with minimum VRRP traffic and without any interaction with the hosts

```
[edit protocols vrrp]
user@host# set ignore-nonstop-routing
```

6. Set up the redundant Ethernet (reth) interfaces and assign the redundant interface to a zone.

```
[edit interfaces]
user@host# set ge-0/0/0 gigether-options redundant-parent reth0
user@host# set ge-0/0/3 gigether-options redundant-parent reth1
user@host# set ge-5/0/0 gigether-options redundant-parent reth0
user@host# set ge-5/0/3 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth1 redundant-ether-options redundancy-group 2
```

7. Configure the family inet address and virtual address for the redundant interface 0 unit 0.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 virtual-address
192.168.110.3
user@host# set reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 virtual-inet6-
address 2001:db8::3
```

8. Configure the family inet address and virtual address for the redundant interface 1 unit 0.

```
[edit interfaces]
user@host# set reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 virtual-address
192.168.120.3
user@host# set reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3 virtual-inet6-
address 2001:db8::4
```

9. Set the priority of the redundant interface 0 unit 0 to 255.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 priority 255
user@host# set reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 priority 255
```

10. Set the priority of the redundant interface 1 unit 0 to 150.

```
[edit interfaces]
user@host# set reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 priority 150
user@host# set reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3 priority 150
```

11. Configure the redundant interface 0 unit 0 to accept all packets sent to the virtual IP address.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 accept-data
user@host# set reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 accept-data
```

12. Configure the redundant interface 1 unit 0 to accept all packets sent to the virtual IP address.

```
[edit interfaces]
user@host# set reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 accept-data
user@host# set reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3 accept-data
```

## Results

From configuration mode, confirm your configuration by entering the `show interfaces reth0` and `show interfaces reth1` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces reth0
redundant-ether-options {
  redundancy-group 1;
}
unit 0 {
  family inet {
    address 192.0.2.2/24 {
      vrrp-group 0 {
        virtual-address 192.0.2.3;
        priority 255;
        accept-data;
      }
    }
  }
  family inet6 {
    address 2001:db8::2/32 {
      vrrp-inet6-group 2 {
        virtual-inet6-address 2001:db8::3;
        priority 255;
        accept-data;
      }
    }
  }
}
```

```
[edit]
user@host# show interfaces reth1
redundant-ether-options {
  redundancy-group 2;
}
unit 0 {
  family inet {
    address 192.0.2.4/24 {
      vrrp-group 1 {
```

```

        virtual-address
192.0.2.5;
        priority 150;
        accept-data;
    }
}
}
family inet6 {
    address 2001:db8::3/32
{
    vrrp-inet6-group 3
{
    virtual-inet6-address
2001:db8::4;
    priority
150;
    accept-
data;
    }
}
}
}
}
}

```

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

## Configuring VRRP Groups on Standalone Device

### 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.

```

set protocols vrrp version-3
set interfaces xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 virtual-address 192.0.2.3
set interfaces xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 priority 50
set interfaces xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 accept-data
set interfaces xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 virtual-inet6-address
2001:db8::3
set interfaces xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 priority 50

```



```

set interfaces xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 accept-data
set interfaces xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 virtual-address 192.0.2.5
set interfaces xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 priority 50
set interfaces xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 accept-data
set interfaces xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 virtual-inet6-address
2001:db8::4
set interfaces xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 priority 50
set interfaces xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 accept-data

```

## Step-by-Step Procedure

To configure VRRP groups on standalone device:

1. Set vrrp version to 3.

```

[edit protocols vrrp]
user@host# set version-3

```

2. Configure the family inet address and virtual address for the Gigabit Ethernet interface unit 0.

```

[edit interfaces]
user@host# set xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 virtual-address 192.0.2.3
user@host# set xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 virtual-inet6-
address 2001:db8::3
user@host# set xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 virtual-address 192.0.2.5
user@host# set xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 virtual-inet6-
address 2001:db8::4

```

3. Set the priority of the Gigabit Ethernet interface unit 0 to 50.

```

[edit interfaces]
user@host# set xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 priority 50
user@host# set xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 priority 50
user@host# set xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 priority 50
user@host# set xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 priority 50

```

#### 4. Configure the Gigabit Ethernet interface unit 0 to accept all packets sent to the virtual IP address.

```
[edit interfaces]
user@host# set xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 accept-data
user@host# set xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 accept-data
user@host# set xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 accept-data
user@host# set xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 accept-data
```

## Results

From configuration mode, confirm your configuration by entering the `show interfaces xe-5/0/5` and `show interfaces xe-5/0/6` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces xe-5/0/5
unit 0 {
  family inet {
    address 192.0.2.1/24 {
      vrrp-group 0 {
        virtual-address 192.0.2.3;
        priority 50;
        accept-data;
      }
    }
  }
  family inet6 {
    address 2001:db8::1/32 {
      vrrp-inet6-group 2 {
        virtual-inet6-address 2001:db8::3;
        priority 50;
        accept-data;
      }
    }
  }
}
}
```

```
[edit]
user@host# show interfaces xe-5/0/6
```

```
unit 0 {
  family inet {
    address 192.0.2.1/24 {
      vrrp-group 1 {
        virtual-address 192.0.2.5;
        priority 50;
        accept-data;
      }
    }
  }
  family inet6 {
    address 2001:db8::5/32 {
      vrrp-inet6-group 3 {
        virtual-inet6-address 2001:db8::4;
        priority 50;
        accept-data;
      }
    }
  }
}
```

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

## Verification

### IN THIS SECTION

- [Verifying the VRRP on Chassis Cluster Devices | 317](#)
- [Verifying the VRRP on standalone device | 318](#)

Confirm that the configuration is working properly.

### Verifying the VRRP on Chassis Cluster Devices

#### Purpose

Verify that VRRP on chassis cluster devices has been configured properly.

## Action

From operational mode, enter the `show vrrp brief` command to display the status of VRRP on chassis cluster devices.

```
user@host> show vrrp brief
Interface      State      Group  VR state VR Mode  Timer   Type   Address
  reth0.0      up                0  master  Active   A 0.149 lcl  192.0.2.3
  vip 192.0.2.3
  reth0.0      up                2  master  Active   A 0.155 lcl  2001:db8::2
  vip 2001:db8:5eff:fe00:202
  vip 2001:db8::2
  reth1.0      up                1  master  Active   A 0.445 lcl  192.0.2.4
  vip 192.0.2.4
  reth1.0      up                3  master  Active   A 0.414 lcl  2001:db8::4
  vip 2001:db8:5eff:fe00:203
  vip 2001:db8::4
```

## Meaning

The sample output shows that the four VRRP groups are active and that the redundant interfaces has assumed the correct primary roles. The lcl address is the physical address of the interface and the vip address is the virtual address shared by redundant interfaces. The Timer value (A 0.149, A 0.155, A 0.445, and A 0.414) indicates the remaining time (in seconds) in which the redundant interfaces expects to receive a VRRP advertisement from the Gigabit Ethernet interfaces. If an advertisement for group 0, 1, 2, and 3 does not arrive before the timer expires, Chassis cluster devices asserts itself as the primary.

## Verifying the VRRP on standalone device

### Purpose

Verify that VRRP has been configured properly on a standalone device.

## Action

From operational mode, enter the `show vrrp brief` command to display the status of VRRP on standalone device.

```
user@host> show vrrp brief
Interface      State      Group  VR state VR Mode  Timer   Type   Address
```

```

    xe-5/0/5.0  up           0  backup  Active    D 3.093 lcl  192.0.2.2.1
vip  192.0.2.2
mas  192.0.2.2.2
    xe-5/0/5.0  up           2  backup  Active    D 3.502 lcl  2001:db8::2:1
vip  2001:db8:200:5eff:fe00:202
vip  2001:db8::2
mas  2001:db8:210:dbff:feff:1000
    xe-5/0/6.0  up           1  backup  Active    D 3.499 lcl  192.0.2.5.1
vip  192.0.2.5
mas  192.0.2.5.2
    xe-5/0/6.0  up           3  backup  Active    D 3.282 lcl  2001:db8::5
vip  2001:db8:200:5eff:fe00:203
vip  2001:db8::4
mas  2001:db8:210:dbff:feff:1001

```

## Meaning

The sample output shows that the four VRRP groups are active and that the Gigabit Ethernet interfaces has assumed the correct backup roles. The lcl address is the physical address of the interface and the vip address is the virtual address shared by Gigabit Ethernet interfaces. The Timer value (D 3.093, D 3.502, D 3.499, and D 3.282) indicates the remaining time (in seconds) in which the Gigabit Ethernet interfaces expects to receive a VRRP advertisement from the redundant interfaces. If an advertisement for group 0, 1, 2, and 3 does not arrive before the timer expires, then the standalone device continues to be a backup device.

## SEE ALSO

---

*authentication-type*

---

*authentication-key*

---

*show vrrp*

## Example: Configuring VRRP for IPv6

### IN THIS SECTION

- Requirements | 320
- Overview | 320
- Configuring VRRP | 321
- Verification | 328

This example shows how to configure VRRP properties for IPv6.

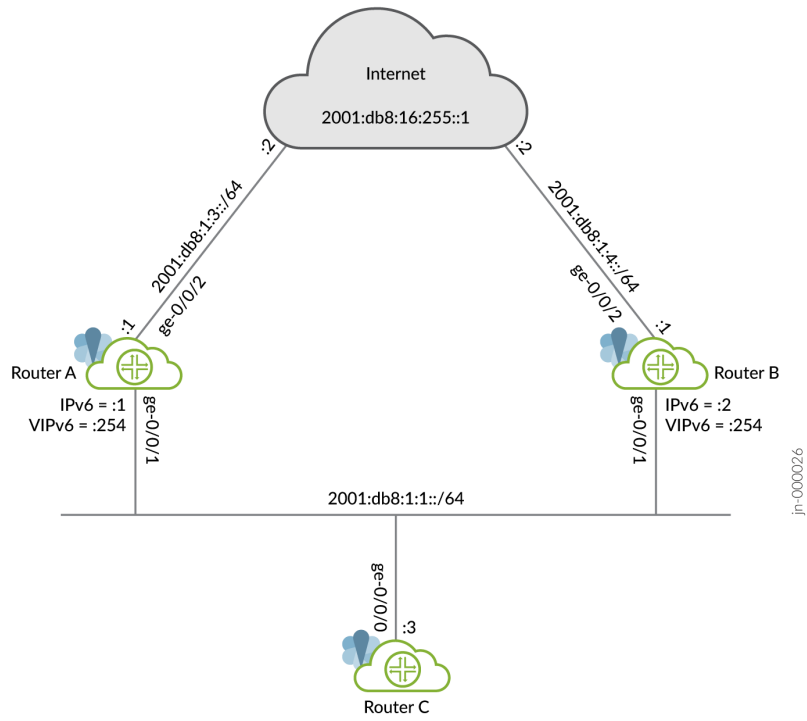
### Requirements

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 | 322](#)
- [Configuring Router B | 325](#)
- [Configuring Router C | 328](#)

## 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 | 328](#)
- [Verifying That VRRP Is Working on Router B | 329](#)
- [Verifying Router C Reaches the Internet Transiting Router A | 330](#)
- [Verifying Router B Becomes Primary for VRRP | 331](#)

### 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 (0 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 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 0.119 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.

## SEE ALSO

[Understanding VRRP](#)

[Configuring VRRP](#)

[Configuring VRRP Route Tracking](#)

## RELATED DOCUMENTATION

[Chassis Cluster Redundant Ethernet Interfaces | 91](#)

[Configuring Chassis Clustering on SRX Series Devices | 120](#)

# NTP Time Synchronization on Chassis Cluster

## IN THIS SECTION

- [NTP Time Synchronization on SRX Series Devices | 333](#)
- [Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP | 334](#)

Network Time Protocol (NTP) is used to synchronize the time between the Packet Forwarding Engine and the Routing Engine in a standalone device and between two devices in a chassis cluster. For more information, see the following topics:

## NTP Time Synchronization on SRX Series Devices

In both standalone and chassis cluster modes, the primary Routing Engine runs the NTP process to get the time from the external NTP server. Although the secondary Routing Engine runs the NTP process in an attempt to get the time from the external NTP server, this attempt fails because of network issues. For this reason, the secondary Routing Engine uses NTP to get the time from the primary Routing Engine.

Use NTP to:

- Send the time from the primary Routing Engine to the secondary Routing Engine through the chassis cluster control link.
- Get the time from an external NTP server to the primary or a standalone Routing Engine.
- Get the time from the Routing Engine NTP process to the Packet Forwarding Engine.

Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, configuring the NTP time adjustment threshold is supported on SRX300, SRX320, SRX340, SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX Virtual Firewall instances. This feature allows you to configure and enforce the NTP adjustment threshold for the NTP service and helps in improve the security and flexibility of the NTP service protocol.

Starting with Junos OS Release 23.4R1 and Junos OS Release 24.2R1, configuring the NTP time adjustment threshold is supported on SRX1600, SRX2300, and SRX4300 devices.

### SEE ALSO

[NTP Overview](#)

---

*ntp threshold*

---

*show system ntp threshold*

---

*set date ntp*

## Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP

### IN THIS SECTION

- [Requirements | 334](#)
- [Overview | 335](#)
- [Configuration | 335](#)
- [Verification | 337](#)

This example shows how to simplify management by synchronizing the time between two SRX Series Firewalls operating in a chassis cluster. Using a Network Time Protocol (NTP) server, the primary node can synchronize time with the secondary node. NTP is used to synchronize the time between the Packet Forwarding Engine and the Routing Engine in a standalone device and between two devices in a chassis cluster. You need to synchronize the system clocks on both nodes of the SRX Series Firewalls in a chassis cluster in order to manage the following items:

- Real-time objects (RTO)
- Licenses
- Software updates
- Node failovers
- Analyzing system logs (syslogs)

### Requirements

This example uses the following hardware and software components:

- SRX Series Firewalls operating in a chassis cluster
- Junos OS Release 12.1X47-D10 or later

Before you begin:

- Understand the basics of the Network Time Protocol. See [NTP Overview](#).

## Overview

### IN THIS SECTION

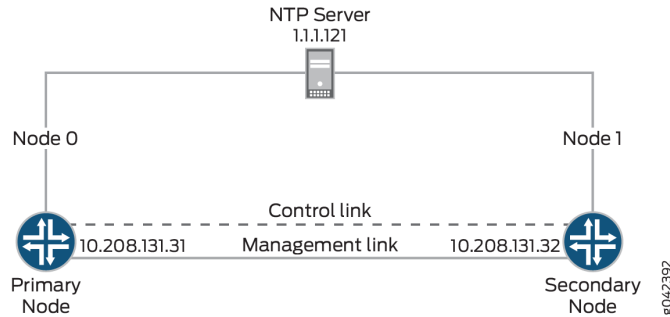
- [Topology | 335](#)

When SRX Series Firewalls are operating in chassis cluster mode, the secondary node cannot access the external NTP server through the revenue port. Junos OS Release 12.1X47 or later supports synchronization of secondary node time with the primary node through the control link by configuring the NTP server on the primary node.

### Topology

[Figure 41 on page 335](#) shows the time synchronization from the peer node using the control link.

**Figure 41: Synchronizing Time From Peer Node Through Control Link**



In the primary node, the NTP server is reachable. The NTP process on the primary node can synchronize the time from the NTP server, and the secondary node can synchronize the time with the primary node from the control link.

## Configuration

### IN THIS SECTION

- [CLI Quick Configuration | 336](#)
- [Synchronizing Time from the NTP server | 336](#)

## CLI Quick Configuration

To quickly configure this example, and synchronize the time from the NTP server, 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.

```
set system ntp server 1.1.1.121
```

## Synchronizing Time from the NTP server

### Step-by-Step Procedure

In this example, you configure the primary node to get its time from an NTP server at IP address 1.1.1.121. To synchronize the time from the NTP server:

1. Configure the NTP server.

```
{primary:node0}[edit]  
[edit system]  
user@host# set ntp server 1.1.1.121
```

2. Commit the configuration.

```
user@host#commit
```

## Results

From configuration mode, confirm your configuration by entering the `show system ntp` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
{primary:node0}[edit]
user@host# show system ntp
server 1.1.1.121
```

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

## Verification

### IN THIS SECTION

- [Verifying the NTP Configuration on the Primary Node | 337](#)
- [Verifying the NTP Configuration on the Secondary Node | 339](#)

Confirm that the configuration is working properly.

### Verifying the NTP Configuration on the Primary Node

#### Purpose

Verify that the configuration is working properly.

#### Action

From operational mode, enter the `show ntp associations` command:

```
user@host> show ntp associations
remote      refid      st t  when poll reach  delay  offset  jitter
=====
*1-1-1-121-dynami 10.208.0.50    4 -   63   64   65   4.909  -12.067  2.014
```

From operational mode, enter the `show ntp status` command:

```
user@host> show ntp status
status=0664 leap_none, sync_ntp, 6 events, event_peer/strat_chg,
version="ntpd 4.2.0-a Fri Mar 21 00:50:30 PDT 2014 (1)",
processor="i386", system="JUNOS12.1I20140320_srx_12q1_x47.1-637245",
leap=00, stratum=5, precision=-20, rootdelay=209.819,
rootdispersion=513.087, peer=14596, refid=1.1.1.121,
reftime=d6dbb2f9.b3f41ff7 Tue, Mar 25 2014 15:47:05.702, poll=6,
clock=d6dbb47a.72918b20 Tue, Mar 25 2014 15:53:30.447, state=4,
offset=-6.066, frequency=-55.135, jitter=4.343, stability=0.042
```

## Meaning

The output on the primary and secondary node shows the NTP association as follows:

- `remote`—Address or name of the remote NTP peer.
- `refid`—Reference identifier of the remote peer. If the reference identifier is not known, this field shows a value of 0.0.0.0.
- `st`—Stratum of the remote peer.
- `t`—Type of peer: b (broadcast), l (local), m (multicast), or u (unicast).
- `when`—When the last packet from the peer was received.
- `poll`—Polling interval, in seconds.
- `reach`—Reachability register, in octal.
- `delay`—Current estimated delay of the peer, in milliseconds.
- `offset`—Current estimated offset of the peer, in milliseconds.
- `jitter`—Magnitude of jitter, in milliseconds.

The output on the primary and secondary node shows the NTP status as follows:

- `status`—System status word, a code representing the status items listed.
- `x events`—Number of events that have occurred since the last code change. An event is often the receipt of an NTP polling message.
- `version`—A detailed description of the version of NTP being used.



- `processor`—Current hardware platform and version of the processor.
- `system`—Detailed description of the name and version of the operating system in use.
- `leap`—Number of leap seconds in use.
- `stratum`—Stratum of the peer server. Anything greater than 1 is a secondary reference source, and the number roughly represents the number of hops away from the stratum 1 server. Stratum 1 is a primary reference, such as an atomic clock.
- `precision`—Precision of the peer clock, how precisely the frequency and time can be maintained with this particular timekeeping system.
- `rootdelay`—Total roundtrip delay to the primary reference source, in seconds.
- `rootdispersion`—Maximum error relative to the primary reference source, in seconds.
- `peer`—Identification number of the peer in use.
- `refid`—Reference identifier of the remote peer. If the reference identifier is not known, this field shows a value of 0.0.0.0.
- `reftime`—Local time, in timestamp format, when the local clock was last updated. If the local clock has never been synchronized, the value is zero.
- `poll`—NTP broadcast message polling interval, in seconds.
- `clock`—Current time on the local router clock.
- `state`—Current mode of NTP operation, where 1 is symmetric active, 2 is symmetric passive, 3 is client, 4 is server, and 5 is broadcast.
- `offset`—Current estimated offset of the peer, in milliseconds. Indicates the time difference between the reference clock and the local clock.
- `frequency`—Frequency of the clock.
- `jitter`—Magnitude of jitter, in milliseconds.
- `stability`—Measurement of how well this clock can maintain a constant frequency.

## Verifying the NTP Configuration on the Secondary Node

### Purpose

Verify that the configuration is working properly.

## Action

From operational mode, enter the `show ntp associations` command:

```
user@host> show ntp associations
remote  refid  st t   when poll reach delay  offset jitter
=====
1-1-1-121-dynami .INIT.      16 -   - 1024   0   0.000  0.000 4000.00
*129.96.0.1      1.1.1.121    5 u   32   64 377   0.417  0.760  1.204
```

From operational mode, enter the `show ntp status` command:

```
user@host> show ntp status
status=0664 leap_none, sync_ntp, 6 events, event_peer/strat_chg,
version="ntpd 4.2.0-a Thu Mar 13 01:53:03 PDT 2014 (1)",
processor="i386", system="JUNOS12.1I20140312_srx_12q1_x47.2-635305",
leap=00, stratum=12, precision=-20, rootdelay=2.408,
rootdispersion=892.758, peer=51948, refid=1.1.1.121,
reftime=d6d646bb.853d2f42 Fri, Mar 21 2014 13:03:55.520, poll=6,
clock=d6d647bc.e8f28b2f Fri, Mar 21 2014 13:08:12.909, state=4,
offset=-1.126, frequency=-62.564, jitter=0.617, stability=0.002
```

## 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
15.1X49-D70	Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, configuring the NTP time adjustment threshold is supported on SRX300, SRX320, SRX340, SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX Virtual Firewall instances. This feature allows you to configure and enforce the NTP adjustment threshold for the NTP service and helps in improve the security and flexibility of the NTP service protocol.

## RELATED DOCUMENTATION

[Time Management Routing Guide for Administration Devices](#)

[Verifying Chassis Cluster Configuration Synchronization Status](#) | 143

# Active/Passive Chassis Cluster Deployments

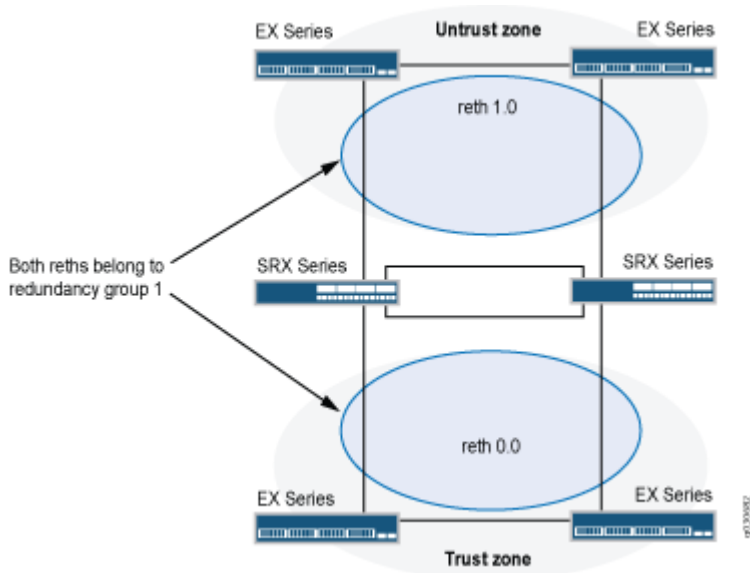
## IN THIS SECTION

- [Understanding Active/Passive Chassis Cluster Deployment | 341](#)
- [Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Firewalls | 342](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair \(SRX1500 or SRX1600\) | 361](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair \(J-Web\) | 378](#)
- [Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel | 380](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel | 382](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel \(J-Web\) | 404](#)

## Understanding Active/Passive Chassis Cluster Deployment

In this case, a single device in the cluster is used to route all traffic while the other device is used only in the event of a failure (see [Figure 42 on page 342](#)). When a failure occurs, the backup device becomes primary and controls all forwarding.

Figure 42: Active/Passive Chassis Cluster Scenario



An active/passive *chassis cluster* can be achieved by using redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. If any of the interfaces in an active group in a node fails, the group is declared inactive and all the interfaces in the group fail over to the other node.

This configuration minimizes the traffic over the fabric link because only one node in the cluster forwards traffic at any given time.

## SEE ALSO

[Chassis Cluster Overview | 2](#)

## Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Firewalls

### IN THIS SECTION

- [Requirements | 343](#)
- [Overview | 343](#)
- [Configuration | 345](#)

This example shows how to set up basic active/passive chassis clustering on an SRX5800 firewalls.

## Requirements

Before you begin:

- You need two SRX5800 Firewalls with identical hardware configurations, and optionally one MX480 edge router, and one EX9214 Ethernet Switch for sending end to end data traffic.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models.
- Before the cluster is formed, you must configure control ports for each device, as well as assign a cluster ID and node ID to each device, and then reboot. When the system boots, both the nodes come up as a cluster.

Control port configuration is required for SRX5400, SRX5600, and SRX5800 firewalls.

Now the devices are a pair. From this point forward, configuration of the cluster is synchronized between the node members, and the two separate devices function as one device.

## Overview

This example shows how to set up basic active/passive chassis clustering on an SRX Series Firewall. The basic active/passive example is the most common type of chassis cluster.

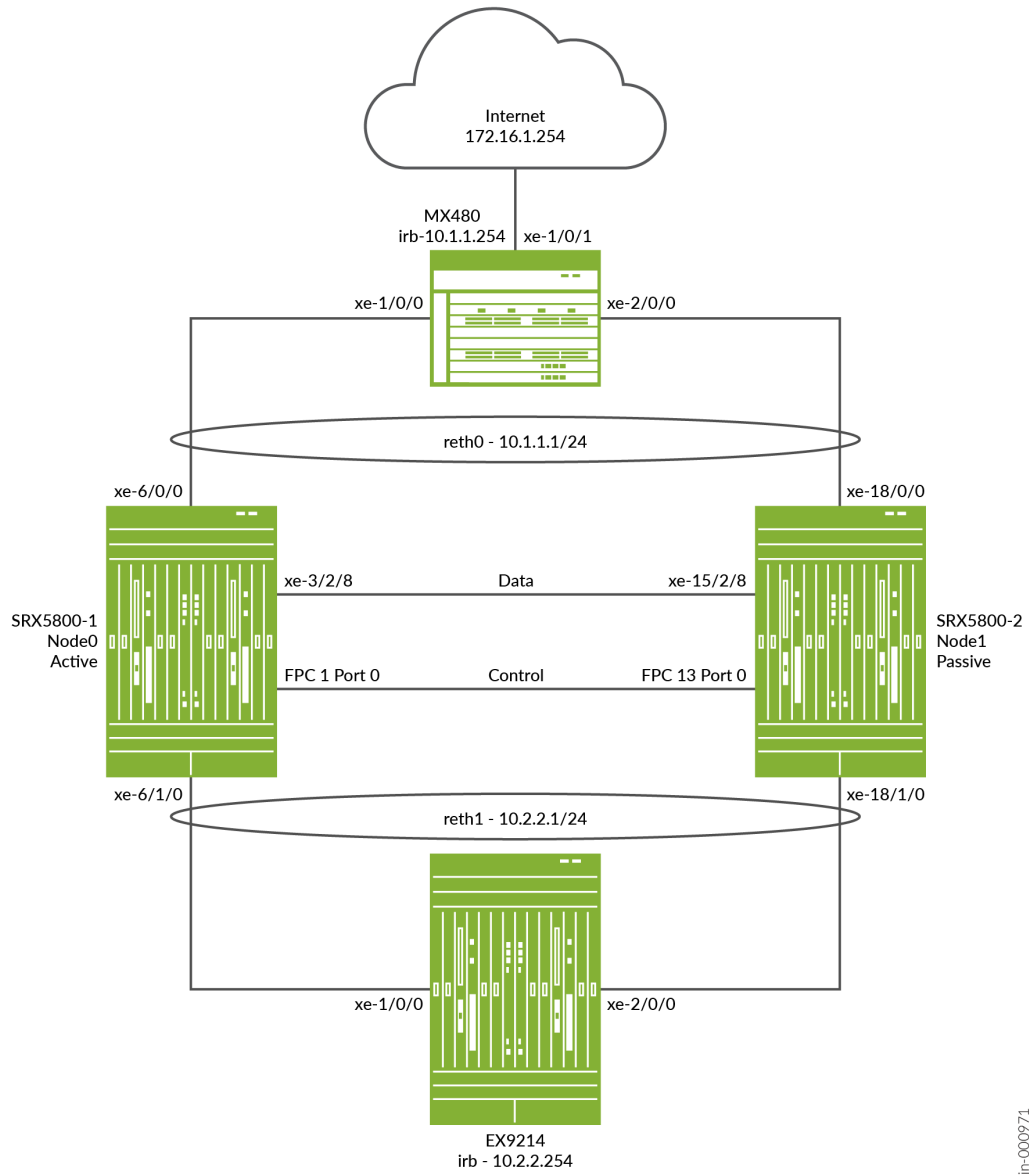
The basic active/passive chassis cluster consists of two devices:

- One device actively provides routing, firewall, NAT, VPN, and security services, along with maintaining control of the chassis cluster.
- The other device passively maintains its state for cluster failover capabilities in case the active device becomes inactive.

This active/passive mode example for the SRX5800 Firewall does not describe in detail miscellaneous configurations such as how to configure NAT, security policies, or VPNs. They are essentially the same as they would be for standalone configurations. However, if you are performing proxy ARP in chassis cluster configurations, you must apply the proxy ARP configurations to the reth interfaces rather than the member interfaces because the RETH interfaces hold the logical configurations. See [Configuring Proxy ARP for NAT \(CLI Procedure\)](#). You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800 Firewall. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

Figure 43 on page 344 shows the topology used in this example.

Figure 43: Basic Active/Passive Chassis Clustering on an SRX Series Firewall Topology Example



jn-000971

## Configuration

### IN THIS SECTION

- [Configuring the Control Ports and Enabling Cluster Mode | 345](#)

## Configuring the Control Ports and Enabling Cluster Mode

### 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.

On {primary:node0}

```
[edit]
set groups node0 system host-name hostA
set groups node0 system backup-router 10.52.63.254
set groups node0 system backup-router destination 10.0.0.0/8
set groups node0 interfaces fxp0 unit 0 family inet address 10.52.43.57/19
set groups node1 system host-name hostB
set groups node1 system backup-router 10.52.63.254
set groups node1 system backup-router destination 10.0.0.0/8
set groups node1 interfaces fxp0 unit 0 family inet address 10.52.52.27/19
set apply-groups "${node}"
set chassis cluster control-ports fpc 1 port 0
set chassis cluster control-ports fpc 13 port 0
set chassis cluster control-link-recovery
set chassis cluster reth-count 2
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
set security policies from-zone trust to-zone untrust policy allow match source-address any
```

```

set security policies from-zone trust to-zone untrust policy allow match destination-address any
set security policies from-zone trust to-zone untrust policy allow match application any
set security policies from-zone trust to-zone untrust policy allow then permit
set security policies from-zone untrust to-zone trust policy allow match source-address any
set security policies from-zone untrust to-zone trust policy allow match destination-address any
set security policies from-zone untrust to-zone trust policy allow match application any
set security policies from-zone untrust to-zone trust policy allow then permit
set security zones security-zone trust host-inbound-traffic system-services ping
set security zones security-zone trust interfaces reth1.50
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces reth0.51
set interfaces xe-6/1/0 gigether-options redundant-parent reth1
set interfaces xe-6/0/0 gigether-options redundant-parent reth0
set interfaces xe-18/1/0 gigether-options redundant-parent reth1
set interfaces xe-18/0/0 gigether-options redundant-parent reth0
set interfaces fab0 fabric-options member-interfaces xe-3/2/8
set interfaces fab1 fabric-options member-interfaces xe-15/2/8
set interfaces reth0 vlan-tagging
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 51 vlan-id 51
set interfaces reth0 unit 51 family inet address 10.1.1.1/24
set interfaces reth1 vlan-tagging
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 50 vlan-id 50
set interfaces reth1 unit 50 family inet address 10.2.2.1/24
set routing-options static route 10.0.0.0/8 next-hop 10.52.63.254
set routing-options static route 172.16.1.0/24 next-hop 10.1.1.254

```

(Optional) To quickly configure an EX9214 Core Switch, 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.

On EX device

```

[edit]
set interfaces xe-1/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-1/0/0 unit 0 family ethernet-switching vlan members v50
set interfaces xe-2/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-2/0/0 unit 0 family ethernet-switching vlan members v50
set interfaces irb unit 50 family inet address 10.2.2.254/24
set routing-options static route 10.1.1.0/24 next-hop 10.2.2.1

```



```

set routing-options static route 172.16.1.0/24 next-hop 10.2.2.1
set vlans v50 vlan-id 50
set vlans v50 l3-interface irb.50

```

(Optional) To quickly configure an MX480 edge router, 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.

On MX device

```

[edit]
set interfaces xe-1/0/0 encapsulation ethernet-bridge
set interfaces xe-1/0/0 unit 0 family bridge interface-mode trunk
set interfaces xe-1/0/0 unit 0 family bridge vlan-id 51
set interfaces xe-1/0/1 unit 0 family inet address 172.16.1.1/24
set interfaces xe-2/0/0 encapsulation ethernet-bridge
set interfaces xe-2/0/0 unit 0 family bridge interface-mode trunk
set interfaces xe-2/0/0 unit 0 family bridge vlan-id 51
set interfaces irb unit 0 family inet address 10.1.1.254/24
set routing-options static route 10.2.2.0/24 next-hop 10.1.1.1
set bridge-domains v51 domain-type bridge
set bridge-domains v51 vlan-id 51
set bridge-domains v51 routing-interface irb.0

```

## Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure a chassis cluster on an SRX Series Firewall:

In cluster mode, the configuration is synchronized over the control link between the nodes when you execute a `commit` command. All commands are applied to both nodes regardless of from which device the command is configured.

1. Because the SRX5000 Firewall chassis cluster configuration is contained within a single common configuration, to assign some elements of the configuration to a specific member only, you must use the Junos OS node-specific configuration method called groups. The `set apply-groups ${node}` command uses the node variable to define how the groups are applied to the nodes; each node recognizes its number and accepts the configuration accordingly. You must also configure out-of-band management on the `fxp0` interface of the SRX5000 Firewall using separate IP addresses for the individual control planes of the cluster.

Configuring the backup router destination address as x.x.x.0/0 is not allowed.

```
user@hostA# set groups node0 system host-name hostA
user@hostA# set groups node0 system backup-router 10.52.63.254
user@hostA# set groups node0 system backup-router destination 10.0.0.0/8
user@hostA# set groups node0 interfaces fxp0 unit 0 family inet address 10.52.43.57/19
```

```
user@hostB# set groups node1 system host-name hostB
user@hostB# set groups node1 system backup-router 10.52.63.254
user@hostB# set groups node1 system backup-router destination 10.0.0.0/8
user@hostB# set groups node1 interfaces fxp0 unit 0 family inet address 10.52.52.27/19
```

The above groups node0 and node1 configuration is committed, but not applied. Once the device is up in cluster, these commands are applied using `set apply-groups "${node}"`.

2. Use the following commands to configure the node 0, which is primary. The node 1 is unreachable till the node configuration is committed. The node 0 will automatically sync the configuration through the control port to node 1 and it is not required to explicitly configure node 1.

```
user@hostA# set apply-groups "${node}"
```

3. Configure the control port for each device, and commit the configuration.

Ensure to have the physical control link connection between the SPC cards on both the nodes as per the configuration.

The control ports are derived based on the SPC location in the chassis and offset value is based on the platform. In the below example the SPC is present in revenue slot 1 and because offset of SRX5800 is 12, the control ports are 1, 13. You can view the Offset value for particular platform using `"jwhoami -c"` command in shell mode. You must enter the following commands on both devices. For example:

- On node 0:

```
user@hostA# set chassis cluster control-ports fpc 1 port 0
user@hostA# set chassis cluster control-ports fpc 13 port 0
user@hostA# commit
```

- On node 1:

```
user@hostB# set chassis cluster control-ports fpc 1 port 0
user@hostB# set chassis cluster control-ports fpc 13 port 0
user@hostB# commit
```

4. Set the two devices to cluster mode. A reboot is required to enter into cluster mode after the cluster ID and node ID are set. You can cause the system to boot automatically by including the `reboot` parameter in the CLI command line. You must enter the operational mode commands on both devices. For example:

- On node 0:

```
user@hostA> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@hostB> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster ID must be the same on both devices in a cluster, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 1 through 255. Setting a cluster ID to 0 is equivalent to disabling a cluster. But it is recommended to use `set chassis cluster disable` to break the nodes from cluster.

5. Configure redundancy groups for chassis clustering. Each node has interfaces in a redundancy group where interfaces are active in active redundancy groups (multiple active interfaces can exist in one redundancy group). Redundancy group 0 controls the control plane and redundancy group 1+ controls the data plane and includes the data plane ports. For this active/passive mode example, only one chassis cluster member is active at a time so you need to define redundancy groups 0 and 1 only. Besides redundancy groups, you must also define:
  - Redundant Ethernet groups—Configure how many redundant Ethernet interfaces (member links) will be active on the device so that the system can allocate the appropriate resources for it.
  - Priority for control plane and data plane—Define which device has priority (for chassis cluster, high priority is preferred) for the control plane, and which device is preferred to be active for the data plane.
    - In active/passive or active/active mode, the control plane (redundancy group 0) can be active on a chassis different from the data plane (redundancy group 1+ and groups) chassis. However, for this example we recommend having both the control and data plane active on

the same chassis member. When traffic passes through the fabric link to go to another member node, latency is introduced (z line mode traffic).

- On SRX Series Firewalls (SRX5000 line), the IPsec VPN is not supported in active/active chassis cluster configuration (that is, when there are multiple RG1+ redundancy groups) in Z mode.

```

user@hostA# set chassis cluster reth-count 2
user@hostA# set chassis cluster redundancy-group 1 node 0 priority 254
user@hostA# set chassis cluster redundancy-group 1 node 1 priority 1
user@hostA# set chassis cluster redundancy-group 0 node 0 priority 254
user@hostA# set chassis cluster redundancy-group 0 node 1 priority 1

```

6. Configure the fabric (data) ports of the cluster that are used to pass RTOs in active/passive mode. For this example, use one of the revenue ports. Define two fabric interfaces, one on each chassis, to connect together.

Configure the data interfaces on the platform so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly. Seamless transition to a new active node will occur with data plane failover. In case of control plane failover, all the daemons are restarted on the new node thus enabling a graceful restart to avoid losing neighborhood with peers (ospf, bgp). This promotes a seamless transition to the new node without any packet loss.

You must define the following items:

- Define the membership information of the member interfaces to the reth interface.
- Define which redundancy group the reth interface is a member of. For this active/passive example, it is always 1.
- Define reth interface information such as the IP address of the interface.

```

{primary:node0}[edit]
user@hostA# set interfaces xe-6/1/0 gigger-options redundant-parent reth1
user@hostA# set interfaces xe-6/0/0 gigger-options redundant-parent reth0
user@hostA# set interfaces xe-18/1/0 gigger-options redundant-parent reth1
user@hostA# set interfaces xe-18/0/0 gigger-options redundant-parent reth0
user@hostA# set interfaces reth0 vlan-tagging
user@hostA# set interfaces reth0 redundant-ether-options redundancy-group 1
user@hostA# set interfaces reth0 unit 51 vlan-id 51
user@hostA# set interfaces reth0 unit 51 family inet address 10.1.1.1/24
user@hostA# set interfaces reth1 vlan-tagging
user@hostA# set interfaces reth1 redundant-ether-options redundancy-group 1

```

```

user@hostA# set interfaces reth1 unit 50 vlan-id 50
user@hostA# set interfaces reth1 unit 50 family inet address 10.2.2.1/24
user@hostA# set interfaces fab0 fabric-options member-interfaces xe-3/2/8
user@hostA# set interfaces fab1 fabric-options member-interfaces xe-15/2/8

```

7. (Optional) Configure the chassis cluster behavior in case of a failure. For the SRX5800 Firewall, the failover threshold is set at 255. You can alter the weights to determine the impact on the chassis failover. You must also configure control link recovery. The recovery automatically causes the secondary node to reboot should the control link fail, and then come back online. Enter these commands on node 0.

```

{primary:node0}[edit]
user@hostA# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
user@hostA# set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255
user@hostA# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
user@hostA# set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
user@hostA# set chassis cluster control-link-recovery

```

This step completes the chassis cluster configuration part of the active/passive mode example for the SRX5800 Firewall. The rest of this procedure describes how to configure the zone, virtual router, routing, EX9214 Core Switch, and MX480 Edge Router to complete the deployment scenario.

8. (Optional) Configure and connect the reth interfaces to the appropriate zones and virtual routers. For this example, leave the reth0 and reth1 interfaces in the default virtual router inet.0, which does not require any additional configuration.

```

{primary:node0}[edit]
user@hostA# set security zones security-zone trust host-inbound-traffic system-services ping
user@hostA# set security zones security-zone trust interfaces reth1.50
user@hostA# set security zones security-zone untrust host-inbound-traffic system-services
ping
user@hostA# set security zones security-zone untrust interfaces reth0.51

```

9. Create the security policy to permit traffic from the trust zone to the untrust zone.

```

user@hostA# set security policies from-zone trust to-zone untrust policy allow match source-address any
user@hostA# set security policies from-zone trust to-zone untrust policy allow match

```

```

destination-address any
user@hostA# set security policies from-zone trust to-zone untrust policy allow match
application any
user@hostA# set security policies from-zone trust to-zone untrust policy allow then permit
user@hostA# set security policies from-zone untrust to-zone trust policy allow match source-
address any
user@hostA# set security policies from-zone untrust to-zone trust policy allow match
destination-address any
user@hostA# set security policies from-zone untrust to-zone trust policy allow match
application any
user@hostA# set security policies from-zone untrust to-zone trust policy allow then permit

```

10. (Optional) For the EX9214 Ethernet Switch, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Firewall; most notably the VLANs, routing, and interface configuration.

```

[edit]
user@switch# set interfaces xe-1/0/0 unit 0 family ethernet-switching interface-mode trunk
user@switch# set interfaces xe-1/0/0 unit 0 family ethernet-switching vlan members v50
user@switch# set interfaces xe-2/0/0 unit 0 family ethernet-switching interface-mode trunk
user@switch# set interfaces xe-2/0/0 unit 0 family ethernet-switching vlan members v50
user@switch# set interfaces irb unit 50 family inet address 10.2.2.254/24
user@switch# set routing-options static route 10.1.1.0/24 next-hop 10.2.2.1
user@switch# set routing-options static route 172.16.1.0/24 next-hop 10.2.2.1
user@switch# set vlans v50 vlan-id 50
user@switch# set vlans v50 l3-interface irb.50

```

11. (Optional) For the MX480 edge router, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Firewall; most notably you must use an IRB interface within a virtual switch instance on the switch.

```

[edit]
user@router# set interfaces xe-1/0/0 encapsulation ethernet-bridge
user@router# set interfaces xe-1/0/0 unit 0 family bridge interface-mode trunk
user@router# set interfaces xe-1/0/0 unit 0 family bridge vlan-id 51
user@router# set interfaces xe-1/0/1 unit 0 family inet address 172.16.1.1/24
user@router# set interfaces xe-2/0/0 encapsulation ethernet-bridge
user@router# set interfaces xe-2/0/0 unit 0 family bridge interface-mode trunk
user@router# set interfaces xe-2/0/0 unit 0 family bridge vlan-id 51
user@router# set interfaces irb unit 0 family inet address 10.1.1.254/24
user@router# set routing-options static route 10.2.2.0/24 next-hop 10.1.1.1

```

```

user@router# set bridge-domains v51 domain-type bridge
user@router# set bridge-domains v51 vlan-id 51
user@router# set bridge-domains v51 routing-interface irb.0

```

## Verification

### IN THIS SECTION

- [Verify Chassis Cluster Status | 353](#)
- [Verify Chassis Cluster Interfaces | 354](#)
- [Verify Chassis Cluster Statistics | 355](#)
- [Verify Chassis Cluster Control Plane Statistics | 357](#)
- [Verify Chassis Cluster Data Plane Statistics | 358](#)
- [Verify Ping from EX Device | 359](#)
- [Verify Chassis Cluster Redundancy Group Status | 360](#)
- [Troubleshooting with Logs | 360](#)

Confirm that the configuration is working properly.

### Verify Chassis Cluster Status

#### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

#### Action

From operational mode, enter the `show chassis cluster status` command.

```

{primary:node0}
user@hostA> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring

```

```

NH Nexthop monitoring      NP NPC monitoring
SP SPU monitoring         SM Schedule monitoring
CF Config Sync monitoring  RE Relinquish monitoring
IS IRQ storm

```

Cluster ID: 1

```

Node  Priority Status          Preempt Manual  Monitor-failures

```

Redundancy group: 0 , Failover count: 1

```

node0 254 primary          no   no   None
node1 1   secondary       no   no   None

```

Redundancy group: 1 , Failover count: 1

```

node0 254 primary          no   no   None
node1 1   secondary       no   no   None

```

## Verify Chassis Cluster Interfaces

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```

{primary:node0}
user@hostA> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      em0        Up                Disabled     Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status                Security
                (Physical/Monitored)
  fab0    xe-3/2/8        Up / Up              Disabled
  fab0

```



```
fab1 xe-15/2/8 Up / Up Disabled
fab1
```

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

Interface Monitoring:

Interface	Weight	Status (Physical/Monitored)	Redundancy-group
xe-18/1/0	255	Up / Up	1
xe-6/1/0	255	Up / Up	1
xe-18/0/0	255	Up / Up	1
xe-6/0/0	255	Up / Up	1

## Verify Chassis Cluster Statistics

### Purpose

Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

### Action

From operational mode, enter the `show chassis cluster statistics` command.

```
{primary:node0}
user@hostA> show chassis cluster statistics
```

Control link statistics:

```
Control link 0:
Heartbeat packets sent: 229414
Heartbeat packets received: 229385
Heartbeat packet errors: 0
```

Fabric link statistics:

## Child link 0

Probes sent: 459691

Probes received: 459679

## Child link 1

Probes sent: 0

Probes received: 0

## 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	0	0

## Verify Chassis Cluster Control Plane Statistics

### Purpose

Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

### Action

From operational mode, enter the `show chassis cluster control-plane statistics` command.

```
{primary:node0}
user@hostA> show chassis cluster control-plane statistics
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 229474
    Heartbeat packets received: 229445
    Heartbeat packet errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 459809
    Probes received: 459797
  Child link 1
```

```
Probes sent: 0
Probes received: 0
```

## Verify Chassis Cluster Data Plane Statistics

### Purpose

Verify information about the number of RTOs sent and received for services.

### Action

From operational mode, enter the `show chassis cluster data-plane statistics` command.

```
{primary:node0}
user@hostA> show chassis cluster 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
  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

## Verify Ping from EX Device

### Purpose

Verify the connection status from EX device.

### Action

From operational mode, enter the ping 172.16.1.254 count 2 command.

```

user@EX9214> ping 172.16.1.254 count 2
PING 172.16.1.254 (172.16.1.254): 56 data bytes
64 bytes from 172.16.1.254: icmp_seq=0 ttl=62 time=4.599 ms
64 bytes from 172.16.1.254: icmp_seq=1 ttl=62 time=3.192 ms

--- 172.16.1.254 ping statistics ---

```

```
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.192/3.896/4.599/0.704 ms
```

## Verify Chassis Cluster Redundancy Group Status

### Purpose

Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

### Action

From operational mode, enter the `chassis cluster status redundancy-group` command.

```
{primary:node0}
user@hostA> show chassis cluster status redundancy-group 1
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring            SM Schedule monitoring
  CF Config Sync monitoring     RE Relinquish monitoring

Cluster ID: 1
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 1 , Failover count: 1
node0 254    primary    no    no    None
node1 1      secondary no    no    None
```

## Troubleshooting with Logs

### Purpose

Use these logs to identify any chassis cluster issues. You must run these logs on both nodes.

## Action

From operational mode, enter these `show log` commands.

```
user@hostA> show log jsrpd
user@hostA> show log chassisd
user@hostA> show log messages
user@hostA> show log dcd
user@hostA> show traceoptions
```

## SEE ALSO

[Preparing Your Equipment for Chassis Cluster Formation | 35](#)

[Connecting SRX Series Firewalls to Create a Chassis Cluster | 39](#)

## Example: Configuring an Active/Passive Chassis Cluster Pair (SRX1500 or SRX1600)

### IN THIS SECTION

- [Requirements | 361](#)
- [Overview | 362](#)
- [Configuration | 366](#)
- [Verification | 372](#)

This example shows how to configure active/passive chassis clustering for SRX1500 or SRX1600 device.

## Requirements

Before you begin:

1. Physically connect a pair of devices together, ensuring that they are the same models.

2. Create a fabric link by connecting a Gigabit Ethernet interface on one device to another Gigabit Ethernet interface on the other device.
3. Create a control link by connecting the control port of the two SRX1500 devices.
4. Connect to one of the devices using the console port. (This is the node that forms the cluster.) and set the cluster ID and node number.

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

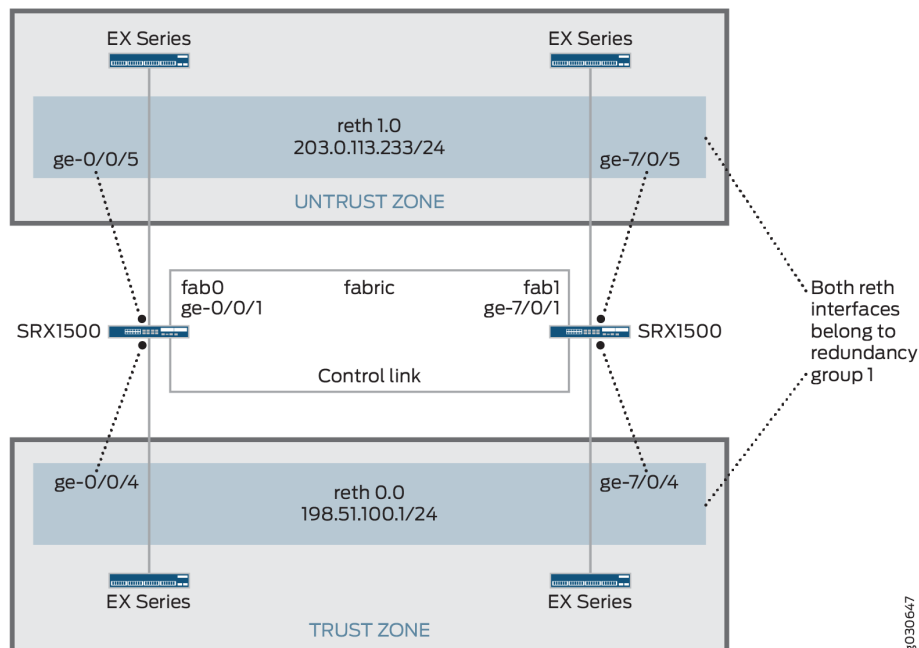
5. Connect to the other device using the console port and set the cluster ID and node number.

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

## Overview

In this example, a single device in the cluster is used to route all traffic, and the other device is used only in the event of a failure. (See [Figure 44 on page 362](#).) When a failure occurs, the backup device becomes primary and controls all forwarding.

**Figure 44: Active/Passive Chassis Cluster Topology**



8030647



You can create an active/passive chassis cluster by configuring redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. This configuration minimizes the traffic over the fabric link because only one node in the cluster forwards traffic at any given time.

In this example, you configure group (applying the configuration with the `apply-groups` command) and chassis cluster information. Then you configure security zones and security policies. See [Table 21 on page 363](#) through [Table 24 on page 365](#).

**Table 21: Group and Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: srx1500-A</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.0.2.110/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: srx1500-B</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.0.2.111/24</li> </ul> </li> </ul>

**Table 22: Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: ge-0/0/1
	fab1	Interface: ge-7/0/1
Heartbeat interval	-	1000
Heartbeat threshold	-	3

Table 22: Chassis Cluster Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Redundancy group	0	<ul style="list-style-type: none"> <li>• Priority:</li> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul>
	1	<ul style="list-style-type: none"> <li>• Priority:</li> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>• ge-0/0/4</li> <li>• ge-7/0/4</li> <li>• ge-0/0/5</li> <li>• ge-7/0/5</li> </ul>
Number of redundant Ethernet interfaces	-	2
Interfaces	ge-0/0/4	Redundant parent: reth0
	ge-7/0/4	Redundant parent: reth0
	ge-0/0/5	Redundant parent: reth1
	ge-7/0/5	Redundant parent: reth1
	reth0	Redundancy group: 1

Table 22: Chassis Cluster Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
		<ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 198.51.100.1/24</li> </ul>
	reth1	Redundancy group: 1  <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 203.0.113.233/24</li> </ul>

Table 23: Security Zone Configuration Parameters

Name	Configuration Parameters
trust	The reth1.0 interface is bound to this zone.
untrust	The reth0.0 interface is bound to this zone.

Table 24: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>• Match criteria:               <ul style="list-style-type: none"> <li>• source-address any</li> <li>• destination-address any</li> <li>• application any</li> </ul> </li> <li>• Action: permit</li> </ul>

## Configuration

### IN THIS SECTION

- [Procedure | 366](#)

### 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, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
[edit]
set groups node0 system host-name srx1500-A
set groups node0 interfaces fxp0 unit 0 family inet address 192.0.2.110/24
set groups node1 system host-name srx1500-B
set groups node1 interfaces fxp0 unit 0 family inet address 192.0.2.111/24
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-7/0/1
set chassis cluster heartbeat-interval 1000
set chassis cluster heartbeat-threshold 3
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/4 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-7/0/4 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/5 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-7/0/5 weight 255
set chassis cluster reth-count 2
set interfaces ge-0/0/5 gigether-options redundant-parent reth1
set interfaces ge-7/0/5 gigether-options redundant-parent reth1
set interfaces ge-0/0/4 gigether-options redundant-parent reth0
set interfaces ge-7/0/4 gigether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 198.51.100.1/24
```

```

set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 203.0.113.233/24
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone trust interfaces reth0.0
set security policies from-zone trust to-zone untrust policy ANY match source-address any
set security policies from-zone trust to-zone untrust policy ANY match destination-address any
set security policies from-zone trust to-zone untrust policy ANY match application any
set security policies from-zone trust to-zone untrust policy ANY then permit

```

## Step-by-Step Procedure

To configure an active/passive chassis cluster:

1. Configure the management interface.

```

{primary:node0}[edit]
user@host# set groups node0 system host-name srx1500-A
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 192.0.2.110/24
user@host# set groups node1 system host-name srx1500-B
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 192.0.2.111/24
user@host# set apply-groups "${node}"

```

2. Configure the fabric interface.

```

{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/1

```

3. Configure heartbeat settings.

```

{primary:node0}[edit]
user@host# set chassis cluster heartbeat-interval 1000
user@host# set chassis cluster heartbeat-threshold 3

```

4. Configure redundancy groups.

```

{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
user@host# set chassis cluster redundancy-group 0 node 1 priority 1

```

```

user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/4 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/4 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/5 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/5 weight 255

```

## 5. Configure redundant Ethernet interfaces.

```

{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
user@host# set interfaces ge-0/0/5 gigger-options redundant-parent reth1
user@host# set interfaces ge-7/0/5 gigger-options redundant-parent reth1
user@host# set interfaces ge-0/0/4 gigger-options redundant-parent reth0
user@host# set interfaces ge-7/0/4 gigger-options redundant-parent reth0
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 198.51.100.1/24
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 203.0.113.233/24

```

## 6. Configure security zones.

```

{primary:node0}[edit]
user@host# set security zones security-zone untrust interfaces reth1.0
user@host# set security zones security-zone trust interfaces reth0.0

```

## 7. Configure security policies.

```

{primary:node0}[edit]
user@host# set security policies from-zone trust to-zone untrust policy ANY match source-
address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match destination-
address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match application
any
user@host# set security policies from-zone trust to-zone untrust policy ANY then permit

```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name srx1500-A;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.0.2.110/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name srx1500-B;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.0.2.110/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
chassis {
```

```
cluster {
    reth-count 2;
    heartbeat-interval 1000;
    heartbeat-threshold 3;
    redundancy-group 0 {
        node 0 priority 100;
        node 1 priority 1;
    }
    redundancy-group 1 {
        node 0 priority 100;
        node 1 priority 1;
        interface-monitor {
            ge-0/0/4 weight 255;
            ge-7/0/4 weight 255;
            ge-0/0/5 weight 255;
            ge-7/0/5 weight 255;
        }
    }
}

interfaces {
    ge-0/0/4 {
        gigheter-options {
            redundant-parent reth0;
        }
    }
    ge-7/0/4 {
        gigheter-options {
            redundant-parent reth0;
        }
    }
    ge-0/0/5 {
        gigheter-options {
            redundant-parent reth1;
        }
    }
    ge-7/0/5 {
        gigheter-options {
            redundant-parent reth1;
        }
    }
    fab0 {
        fabric-options {
```



```
        member-interfaces {
            ge-0/0/1;
        }
    }
}
fab1 {
    fabric-options {
        member-interfaces {
            ge-7/0/1;
        }
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 198.51.100.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 203.0.113.233/24;
        }
    }
}
}
...
security {
    zones {
        security-zone untrust {
            interfaces {
                reth1.0;
            }
        }
    }
    security-zone trust {
        interfaces {
```

```
        reth0.0;
    }
}
policies {
    from-zone trust to-zone untrust {
        policy ANY {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
            }
        }
    }
}
}
```

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

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Status | 373](#)
- [Verifying Chassis Cluster Interfaces | 373](#)
- [Verifying Chassis Cluster Statistics | 374](#)
- [Verifying Chassis Cluster Control Plane Statistics | 375](#)
- [Verifying Chassis Cluster Data Plane Statistics | 376](#)
- [Verifying Chassis Cluster Redundancy Group Status | 377](#)
- [Troubleshooting with Logs | 377](#)

Confirm that the configuration is working properly.

## Verifying Chassis Cluster Status

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no

Redundancy group: 1 , Failover count: 1
  node0              100       primary  no       no
  node1              1         secondary no       no
```

## Verifying Chassis Cluster Interfaces

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces

Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Security
```

0	em0	Up	Disabled
1	em1	Down	Disabled

Fabric link status: Up

Fabric interfaces:

Name	Child-interface	Status	Security
fab0	ge-0/0/1	Up	Disabled
fab0			
fab1	ge-7/0/1	Up	Disabled
fab1			

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	1

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-0/0/4	255	Up	1
ge-7/0/4	255	Up	1
ge-0/0/5	255	Up	1
ge-7/0/5	255	Up	1

## Verifying Chassis Cluster Statistics

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster statistics` command.

```
{primary:node0}
user@host> show chassis cluster statistics
```

## Control link statistics:

## Control link 0:

Heartbeat packets sent: 2276

Heartbeat packets received: 2280

Heartbeat packets errors: 0

## Fabric link statistics:

## Child link 0

Probes sent: 2272

Probes received: 597

## Services Synchronized:

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

## Verifying Chassis Cluster Control Plane Statistics

### Purpose

Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

## Action

From operational mode, enter the `show chassis cluster control-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics
```

```
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

## Verifying Chassis Cluster Data Plane Statistics

### Purpose

Verify information about the number of RTOs sent and received for services.

### Action

From operational mode, enter the `show chassis cluster data-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics
```

```
Services Synchronized:
```

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0

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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

## Verifying Chassis Cluster Redundancy Group Status

### Purpose

Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

### Action

From operational mode, enter the chassis cluster status redundancy-group command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority  Status  Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
node0           100     primary no       no
node1           1       secondary no       no
```

## Troubleshooting with Logs

### Purpose

Use these logs to identify any chassis cluster issues. You must run these logs on both nodes.

## Action

From operational mode, enter these show commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

## SEE ALSO

[Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster | 45](#)

[Chassis Cluster Management Interfaces | 49](#)

[Chassis Cluster Fabric Interfaces | 59](#)

[Chassis Cluster Control Plane Interfaces | 72](#)

[Chassis Cluster Redundancy Groups | 82](#)

[Chassis Cluster Redundant Ethernet Interfaces | 91](#)

## Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web)

1. Enable clustering. See Step 1 in "[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)" on page 361.
2. Configure the management interface. See Step 2 in "[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)" on page 361.
3. Configure the fabric interface. See Step 3 in "[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)" on page 361.
4. Configure the redundancy groups.
  - Select Configure>Chassis Cluster.
  - Enter the following information, and then click Apply:
    - a. Redundant ether-Interface Count: 2
    - b. Heartbeat Interval: 1000



- c. Heartbeat Threshold: 3
- d. Nodes: 0
- e. Group Number: 0
- f. Priorities: 100
- Enter the following information, and then click Apply:
  - a. Nodes: 0
  - b. Group Number: 1
  - c. Priorities: 1
- Enter the following information, and then click Apply:
  - a. Nodes: 1
  - b. Group Number: 0
  - c. Priorities: 100

## 5. Configure the redundant Ethernet interfaces.

- Select Configure>Chassis Cluster.
- Select ge-0/0/4.
- Enter reth1 in the Redundant Parent box.
- Click Apply.
- Select ge-7/0/4.
- Enter reth1 in the Redundant Parent box.
  - a.
- Click Apply.
- Select ge-0/0/5.
- Enter reth0 in the Redundant Parent box.
  - a.
- Click Apply.
- Select ge-7/0/5.

- Enter `reth0` in the Redundant Parent box.
    - a.
  - Click Apply.
  - See Step 5 in "[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)" on page 361 for the last four configuration settings.
6. Configure the security zones. See Step 6 in "[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)" on page 361.
  7. Configure the security policies. See Step 7 in "[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)" on page 361.
  8. Click OK to check your configuration and save it as a candidate configuration, then click Commit Options>Commit.

#### SEE ALSO

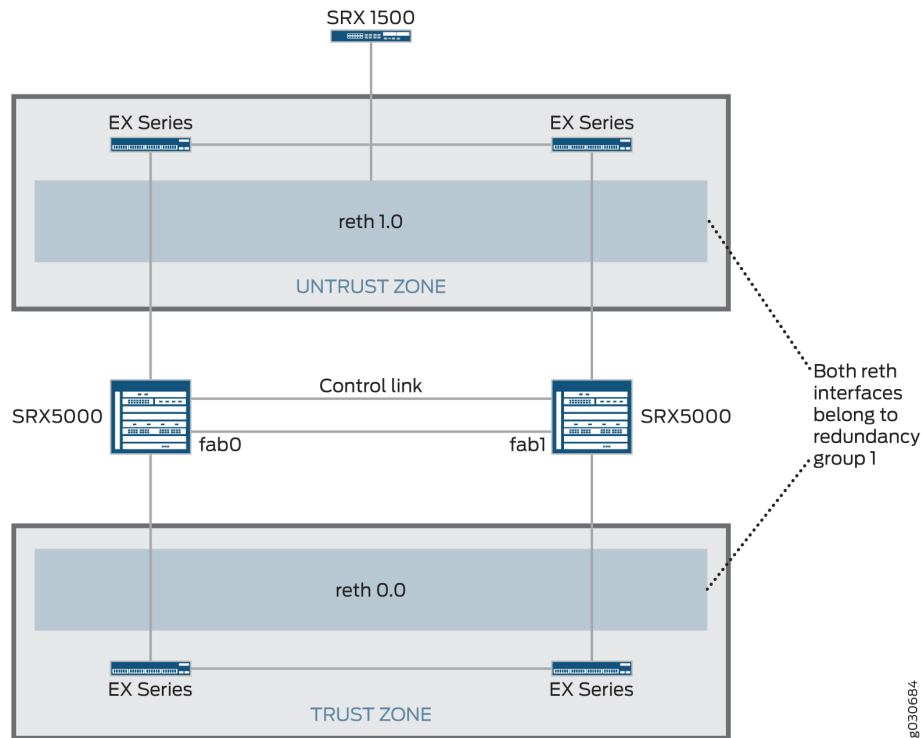
[Understanding Active/Passive Chassis Cluster Deployment](#) | 341

[Example: Configuring an Active/Passive Chassis Cluster Pair \(SRX1500 or SRX1600\)](#) | 361

## Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel

In this case, a single device in the cluster terminates in an IPsec tunnel and is used to process all traffic while the other device is used only in the event of a failure (see [Figure 45 on page 381](#)). When a failure occurs, the backup device becomes primary and controls all forwarding.

Figure 45: Active/Passive Chassis Cluster with IPsec Tunnel Scenario (SRX Series Firewalls)



An active/passive *chassis cluster* can be achieved by using redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. If any of the interfaces in an active group in a node fails, the group is declared inactive and all the interfaces in the group fail over to the other node.

This configuration provides a way for a site-to-site IPsec tunnel to terminate in an active/passive cluster where a redundant Ethernet interface is used as the tunnel endpoint. In the event of a failure, the redundant Ethernet interface in the backup SRX Series Firewall becomes active, forcing the tunnel to change endpoints to terminate in the new active SRX Series Firewall. Because tunnel keys and session information are synchronized between the members of the chassis cluster, a failover does not require the tunnel to be renegotiated and all established sessions are maintained.

In case of RGO (routing engine) failure, the routing protocols need to re-establish on the new Primary node. If VPN monitoring or Dead-peer-detection is configured, and its timer expires before the routing reconverges on new RGO Primary, then VPN tunnel will be brought down and renegotiated.

Dynamic tunnels cannot load-balance across different SPCs.

## SEE ALSO

[IPsec VPN Overview](#)

## Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel

### IN THIS SECTION

- [Requirements | 382](#)
- [Overview | 383](#)
- [Configuration | 389](#)
- [Verification | 399](#)

This example shows how to configure active/passive chassis clustering with an IPsec tunnel for SRX Series Firewalls.

### Requirements

Before you begin:

- Get two SRX5000 models with identical hardware configurations, one SRX1500 or SRX1600 device, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:
  - On node 0:

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster ID is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 1 through 255. Setting a cluster ID to 0 is equivalent to disabling a cluster.

Cluster ID greater than 15 can only be set when the fabric and control link interfaces are connected back-to-back.

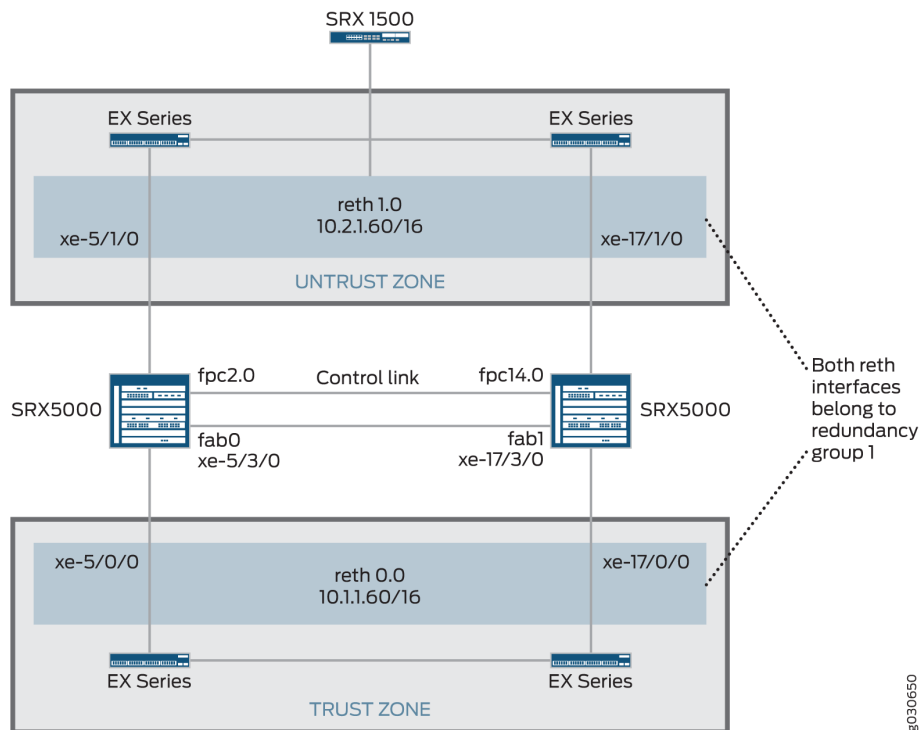
- Get two SRX5000 models with identical hardware configurations, one SRX1500 edge router, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line.

From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device. Member-specific configurations (such as the IP address of the management port of each member) are entered using configuration groups.

## Overview

In this example, a single device in the cluster terminates in an IPsec tunnel and is used to process all traffic, and the other device is used only in the event of a failure. (See [Figure 46 on page 383.](#)) When a failure occurs, the backup device becomes primary and controls all forwarding.

**Figure 46: Active/Passive Chassis Cluster with IPsec Tunnel Topology (SRX Series Firewalls)**



In this example, you configure group (applying the configuration with the `apply-groups` command) and chassis cluster information. Then you configure IKE, IPsec, static route, security zone, and security policy parameters. See [Table 25 on page 384](#) through [Table 31 on page 389](#).

**Table 25: Group and Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: SRX5800-1</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 172.19.100.50/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: SRX5800-2</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 172.19.100.51/24</li> </ul> </li> </ul>

**Table 26: Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: xe-5/3/0
	fab1	Interface: xe-17/3/0
Number of redundant Ethernet interfaces	-	2
Heartbeat interval	-	1000
Heartbeat threshold	-	3

Table 26: Chassis Cluster Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
Redundancy group	0	<ul style="list-style-type: none"> <li>Priority:               <ul style="list-style-type: none"> <li>Node 0: 254</li> <li>Node 1: 1</li> </ul> </li> </ul>
	1	<ul style="list-style-type: none"> <li>Priority:               <ul style="list-style-type: none"> <li>Node 0: 254</li> <li>Node 1: 1</li> </ul> </li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>xe-5/0/0</li> <li>xe-5/1/0</li> <li>xe-17/0/0</li> <li>xe-17/1/0</li> </ul>
Interfaces	xe-5/1/0	Redundant parent: reth1
	xe-5/1/0	Redundant parent: reth1
	xe-5/0/0	Redundant parent: reth0
	xe-17/0/0	Redundant parent: reth0
	reth0	Redundancy group: 1
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.1.1.60/16</li> </ul>

Table 26: Chassis Cluster Configuration Parameters *(Continued)*

Feature	Name	Configuration Parameters
	reth1	Redundancy group: 1
		<ul style="list-style-type: none"> <li>• Multipoint</li> <li>• Unit 0</li> <li>• 10.10.1.1/30</li> </ul>
	st0	
		<ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 10.10.1.1/30</li> </ul>

Table 27: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	-
Policy	preShared	<ul style="list-style-type: none"> <li>• Mode: main</li> <li>• Proposal reference: proposal-set standard</li> <li>• IKE Phase 1 policy authentication method: pre-shared-key ascii-text</li> </ul>



Table 27: IKE Configuration Parameters (Continued)

Feature	Name	Configuration Parameters
Gateway	SRX1500-1	<ul style="list-style-type: none"> <li>• IKE policy reference: perShared</li> <li>• External interface: reth0.0</li> <li>• Gateway address: 10.1.1.90</li> </ul> <p><b>NOTE:</b> In SRX chassis clustering, only reth and lo0 interfaces are supported for the IKE external interface configuration. Other interface types can be configured, but IPsec VPN might not work. If a lo0 logical interface is used as an IKE gateway external interface, it cannot be configured with RGO.</p>

Table 28: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	–
Policy	std	–
VPN	SRX1500-1	<ul style="list-style-type: none"> <li>• IKE gateway reference: SRX1500-1</li> <li>• IPsec policy reference: std</li> <li>• Bind to interface: st0.0</li> <li>• VPN monitoring: vpn-monitor optimized</li> <li>• Tunnels established: establish-tunnels immediately</li> </ul> <p><b>NOTE:</b> The manual VPN name and the site-to-site gateway name cannot be the same.</p>

**NOTE:** A secure tunnel interface (st0) from st0.16000 to st0.16385 is reserved for Multinode High Availability and for HA control link encryption in Chassis Cluster. These interfaces are not user configurable interfaces. You can only use interfaces from st0.0 to st0.15999.

**Table 29: Static Route Configuration Parameters**

Name	Configuration Parameters
0.0.0.0/0	Next hop: 10.2.1.1
10.3.0.0/16	Next hop: 10.10.1.2

**Table 30: Security Zone Configuration Parameters**

Name	Configuration Parameters
trust	<ul style="list-style-type: none"> <li>• All system services are allowed.</li> <li>• All protocols are allowed.</li> <li>• The reth0.0 interface is bound to this zone.</li> </ul>
untrust	<ul style="list-style-type: none"> <li>• All system services are allowed.</li> <li>• All protocols are allowed.</li> <li>• The reth1.0 interface is bound to this zone.</li> </ul>
vpn	<ul style="list-style-type: none"> <li>• All system services are allowed.</li> <li>• All protocols are allowed.</li> <li>• The st0.0 interface is bound to this zone.</li> </ul>

**Table 31: Security Policy Configuration Parameters**

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>• Match criteria: <ul style="list-style-type: none"> <li>• source-address any</li> <li>• destination-address any</li> <li>• application any</li> </ul> </li> <li>• Action: permit</li> </ul>
This security policy permits traffic from the trust zone to the vpn zone.	vpn-any	<ul style="list-style-type: none"> <li>• Match criteria: <ul style="list-style-type: none"> <li>• source-address any</li> <li>• destination-address any</li> <li>• application any</li> </ul> </li> <li>• Action: permit</li> </ul>

## Configuration

### IN THIS SECTION

- [Procedure | 390](#)

## 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
set chassis cluster control-ports fpc 2 port 0
set chassis cluster control-ports fpc 14 port 0
set groups node0 system host-name SRX5800-1
set groups node0 interfaces fxp0 unit 0 family inet address 172.19.100.50/24
set groups node1 system host-name SRX5800-2
set groups node1 interfaces fxp0 unit 0 family inet address 172.19.100.51/24
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces xe-5/3/0
set interfaces fab1 fabric-options member-interfaces xe-17/3/0
set chassis cluster reth-count 2
set chassis cluster heartbeat-interval 1000
set chassis cluster heartbeat-threshold 3
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 interface-monitor xe-5/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-5/1/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-17/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-17/1/0 weight 255
set interfaces xe-5/1/0 gigether-options redundant-parent reth1
set interfaces xe-17/1/0 gigether-options redundant-parent reth1
set interfaces xe-5/0/0 gigether-options redundant-parent reth0
set interfaces xe-17/0/0 gigether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.60/16
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.1.60/16
set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
set security ike policy preShared mode main
```

```

set security ike policy preShared proposal-set standard
set security ike policy preShared pre-shared-key ascii-text "$ABC123"## Encrypted password
set security ike gateway SRX1500-1 ike-policy preShared
set security ike gateway SRX1500-1 address 10.1.1.90
set security ike gateway SRX1500-1 external-interface reth0.0
set security ipsec policy std proposal-set standard
set security ipsec vpn SRX1500-1 bind-interface st0.0
set security ipsec vpn SRX1500-1 vpn-monitor optimized
set security ipsec vpn SRX1500-1 ike gateway SRX1500-1
set security ipsec vpn SRX1500-1 ike ipsec-policy std
set security ipsec vpn SRX1500-1 establish-tunnels immediately
set routing-options static route 0.0.0.0/0 next-hop 10.2.1.1
set routing-options static route 10.3.0.0/16 next-hop 10.10.1.2
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 reth1.0
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 reth0.0
set security zones security-zone vpn host-inbound-traffic system-services all 144
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.0
set security policies from-zone trust to-zone untrust policy ANY match source-address any
set security policies from-zone trust to-zone untrust policy ANY match destination-address any
set security policies from-zone trust to-zone untrust policy ANY match application any
set security policies from-zone trust to-zone vpn policy vpn-any then permit

```

## Step-by-Step Procedure

To configure an active/passive chassis cluster pair with an IPsec tunnel:

1. Configure control ports.

```

{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 2 port 0
user@host# set chassis cluster control-ports fpc 14 port 0

```

2. Configure the management interface.

```

{primary:node0}[edit]
user@host# set groups node0 system host-name SRX5800-1

```

```

user@host# set groups node0 interfaces fxp0 unit 0 family inet address 172.19.100.50/24
user@host# set groups node1 system host-name SRX5800-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 172.19.100.51/24
user@host# set apply-groups "${node}"

```

### 3. Configure the fabric interface.

```

{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces xe-5/3/0
user@host# set interfaces fab1 fabric-options member-interfaces xe-17/3/0

```

### 4. Configure redundancy groups.

```

{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
user@host# set chassis cluster heartbeat-interval 1000
user@host# set chassis cluster heartbeat-threshold 3
user@host# set chassis cluster node 0
user@host# set chassis cluster node 1
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 254
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 preempt
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/0/0 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/1/0 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/0/0 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/1/0 weight 255

```

### 5. Configure redundant Ethernet interfaces.

```

{primary:node0}[edit]
user@host# set interfaces xe-5/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-17/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-5/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-17/0/0 gigether-options redundant-parent reth0
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 10.1.1.60/16

```

```

user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 10.2.1.60/16

```

## 6. Configure IPsec parameters.

```

{primary:node0}[edit]
user@host# set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
user@host# set security ike policy preShared mode main
user@host# set security ike policy preShared proposal-set standard
user@host# set security ike policy preShared pre-shared-key ascii-text "$ABC123"## Encrypted
password
user@host# set security ike gateway SRX1500-1 ike-policy preShared
user@host# set security ike gateway SRX1500-1 address 10.1.1.90
user@host# set security ike gateway SRX1500-1 external-interface reth0.0
user@host# set security ipsec policy std proposal-set standard
user@host# set security ipsec vpn SRX1500-1 bind-interface st0.0
user@host# set security ipsec vpn SRX1500-1 vpn-monitor optimized
user@host# set security ipsec vpn SRX1500-1 ike gateway SRX1500-1
user@host# set security ipsec vpn SRX1500-1 ike ipsec-policy std
user@host# set security ipsec vpn SRX1500-1 establish-tunnels immediately

```

## 7. Configure static routes.

```

{primary:node0}[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.2.1.1
user@host# set routing-options static route 10.3.0.0/16 next-hop 10.10.1.2

```

## 8. Configure security zones.

```

{primary:node0}[edit]
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 reth1.0
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 reth0.0
user@host# set security zones security-zone vpn host-inbound-traffic system-services all
user@host# set security zones security-zone vpn host-inbound-traffic protocols all
user@host# set security zones security-zone vpn interfaces st0.0

```

## 9. Configure security policies.

```
{primary:node0}[edit]
user@host# set security policies from-zone trust to-zone untrust policy ANY match source-
address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match destination-
address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match application
any
user@host# set security policies from-zone trust to-zone vpn policy vpn-any then permit
```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX58001;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 172.19.100.50/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name SRX58002;
    }
  }
}
```



```
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 172.19.100.51/24;
                }
            }
        }
    }
}
apply-groups "${node}";
system {
    root-authentication {
        encrypted-password "$ABC123";
    }
}
chassis {
    cluster {
        reth-count 2;
        heartbeat-interval 1000;
        heartbeat-threshold 3;
        control-ports {
            fpc 2 port 0;
            fpc 14 port 0;
        }
        redundancy-group 0 {
            node 0 priority 254;
            node 1 priority 1;
        }
        redundancy-group 1 {
            node 0 priority 254;
            node 1 priority 1;
            preempt;
            interface-monitor {
                xe-6/0/0 weight 255;
                xe-6/1/0 weight 255;
                xe-18/0/0 weight 255;
                xe-18/1/0 weight 255;
            }
        }
    }
}
}
```

```
interfaces {
  xe-5/0/0 {
    gigether-options {
      redundant-parent reth0;
    }
  }
  xe-5/1/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-17/0/0 {
    gigether-options {
      redundant-parent reth0;
    }
  }
  xe-17/1/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  fab0 {
    fabric-options {
      member-interfaces {
        xe-5/3/0;
      }
    }
  }
  fab1 {
    fabric-options {
      member-interfaces {
        xe-17/3/0;
      }
    }
  }
  reth0 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet {
        address 10.1.1.60/16;
      }
    }
  }
}
```

```
    }
  }
  reth1 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet {
        address 10.2.1.60/16;
      }
    }
  }
}
st0 {
  unit 0 {
    multipoint;
    family inet {
      address 5.4.3.2/32;
    }
  }
}
}
routing-options {
  static {
    route 0.0.0.0/0 {
      next-hop 10.2.1.1;
    }
    route 10.3.0.0/16 {
      next-hop 10.10.1.2;
    }
  }
}
security {
  zones {
    security-zone trust {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
    }
    interfaces {
      reth0.0;
    }
  }
}
```

```
security-zone untrust
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  protocols {
    all;
  }
  interfaces {
    reth1.0;
  }
}

security-zone vpn {
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  protocols {
    all;
  }
  interfaces {
    st0.0;
  }
}

}
}
policies {
  from-zone trust to-zone untrust {
    policy ANY {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
  }
  from-zone trust to-zone vpn {
```

```
policy vpn {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit;
  }
}
}
```

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

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Status | 399](#)
- [Verifying Chassis Cluster Interfaces | 400](#)
- [Verifying Chassis Cluster Statistics | 401](#)
- [Verifying Chassis Cluster Control Plane Statistics | 402](#)
- [Verifying Chassis Cluster Data Plane Statistics | 402](#)
- [Verifying Chassis Cluster Redundancy Group Status | 403](#)
- [Troubleshooting with Logs | 404](#)

Confirm that the configuration is working properly.

### Verifying Chassis Cluster Status

#### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

## Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              1          primary  no       no
  node1              254       secondary no       no

Redundancy group: 1 , Failover count: 1
  node0              1          primary  yes      no
  node1              254       secondary yes      no
```

## Verifying Chassis Cluster Interfaces

### Purpose

Verify the chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

Redundant-ethernet Information:
  Name      Status    Redundancy-group
  reth0     Up        1
  reth1     Up        1

Interface Monitoring:
  Interface    Weight    Status    Redundancy-group
  xe-5/0/0     255      Up        1
  xe-5/1/0     255      Up        1
```

xe-17/0/0	255	Up	1
xe-17/1/0	255	Up	1

## Verifying Chassis Cluster Statistics

### Purpose

Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

### Action

From operational mode, enter the `show chassis cluster statistics` command.

```
{primary:node0}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
Services Synchronized:
  Service name                RTOs sent  RTOs received
  Translation context          0           0
  Incoming NAT                 0           0
  Resource manager             6           0
  Session create               161         0
  Session close                148         0
  Session change               0           0
  Gate create                   0           0
  Session ageout refresh requests 0           0
  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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

## Verifying Chassis Cluster Control Plane Statistics

### Purpose

Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

### Action

From operational mode, enter the `show chassis cluster control-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

## Verifying Chassis Cluster Data Plane Statistics

### Purpose

Verify information about the number of RTOs sent and received for services.



## Action

From operational mode, enter the `show chassis cluster data-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics
```

Services Synchronized:

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

## Verifying Chassis Cluster Redundancy Group Status

### Purpose

Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

## Action

From operational mode, enter the `chassis cluster status redundancy-group` command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority  Status  Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
node0           0        primary yes      no
node1           254     secondary yes      no
```

## Troubleshooting with Logs

### Purpose

Use these logs to identify any chassis cluster issues. You must run these logs on both nodes.

### Action

From operational mode, enter these `show` commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

## Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web)

1. Enable clusters. See Step 1 in ["Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel" on page 382](#).
2. Configure the management interface. See Step 2 in ["Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel" on page 382](#).

3. Configure the fabric interface. See Step 3 in ["Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel"](#) on page 382.
4. Configure the redundancy groups.
  - Select Configure>System Properties>Chassis Cluster.
  - Enter the following information, and then click Apply:
    - a. Redundant ether-Interfaces Count: 2
    - b. Heartbeat Interval: 1000
    - c. Heartbeat Threshold: 3
    - d. Nodes: 0
    - e. Group Number: 0
    - f. Priorities: 254
  - Enter the following information, and then click Apply:
    - a. Nodes: 0
    - b. Group Number: 1
    - c. Priorities: 254
  - Enter the following information, and then click Apply:
    - a. Nodes: 1
    - b. Group Number: 0
    - c. Priorities: 1
  - Enter the following information, and then click Apply:
    - a. Nodes: 1
    - b. Group Number: 1
    - c. Priorities: 1
    - d. Preempt: Select the check box.
    - e. Interface Monitor—Interface: xe-5/0/0
    - f. Interface Monitor—Weight: 255

- g. Interface Monitor—Interface: xe-5/1/0
- h. Interface Monitor—Weight: 255
- i. Interface Monitor—Interface: xe-17/0/0
- j. Interface Monitor—Weight: 255
- k. Interface Monitor—Interface: xe-17/1/0
- l. Interface Monitor—Weight: 255

5. Configure the redundant Ethernet interfaces.

- Select Configure>System Properties>Chassis Cluster.
- Select xe-5/1/0.
- Enter reth1 in the Redundant Parent box.
- Click Apply.
- Select xe-17/1/0.
- Enter reth1 in the Redundant Parent box.
- Click Apply.
- Select xe-5/0/0.
- Enter reth0 in the Redundant Parent box.
- Click Apply.
- Select xe-17/0/0.
- Enter reth0 in the Redundant Parent box.
- Click Apply.
- See Step 5 in ["Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel" on page 382.](#)

6. Configure the IPsec configuration. See Step 6 in ["Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel" on page 382.](#)

7. Configure the static routes .

- Select Configure>Routing>Static Routing.

- Click Add.
  - Enter the following information, and then click Apply:
    - a. Static Route Address: 0.0.0.0/0
    - b. Next-Hop Addresses: 10.2.1.1
  - Enter the following information, and then click Apply:
    - a. Static Route Address: 10.3.0.0/16
    - b. Next-Hop Addresses: 10.10.1.2
8. Configure the security zones. See Step 8 in "[Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel](#)" on page 382.
  9. Configure the security policies. See Step 9 in "[Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel](#)" on page 382.
  10. Click OK to check your configuration and save it as a candidate configuration, then click Commit Options>Commit.

#### RELATED DOCUMENTATION

[SRX Series Chassis Cluster Configuration Overview](#) | 14

[Configuring Chassis Clustering on SRX Series Devices](#) | 120

## Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster

#### IN THIS SECTION

- [Requirements](#) | 408
- [Overview](#) | 408
- [Configuration](#) | 411

This example shows how to set up basic active/passive full mesh chassis clustering on a high-end SRX Series device.

## Requirements

This example uses the following hardware and software components:

- Two Juniper Networks SRX5800 Services Gateways with identical hardware configurations running Junos OS Release 9.6 or later.
- Two Juniper Networks MX480 3D Universal Edge Routers running Junos OS Release 9.6 or later.
- Two Juniper Networks EX9214 Ethernet Switches running Junos OS Release 9.6 or later.

**NOTE:** This configuration example has been tested using the software release listed and is assumed to work on all later releases.

Before you begin:

- Physically connect the two SRX Services Gateways (back-to-back for the fabric and control ports).

## Overview

This example shows how to set up basic active/passive full mesh chassis clustering on a pair of high-end SRX Series devices. Full mesh active/passive clustering allows you to set up an environment that does not have a single point of failure, not only on the SRX Series devices but also on the surrounding network devices. The main difference in the full mesh deployment described in this example and the basic active/passive deployment described in [Configuring an Active/Passive Chassis Cluster Deployment](#) is that additional design elements must be considered to accommodate recovery of possible failure scenarios.

Full mesh chassis clustering requires you to configure reth interfaces for each node and ensure that they are connected together by one or more switches. In this scenario, shown in [Figure 47 on page 410](#), there are four reth interfaces (reth0, reth1, reth2, and reth3). A reth interface bundles the two physical interfaces (one from each node) together. A reth interface is part of a redundancy group. Only the member that is on the primary node (active) for the redundancy group is active. The member on the secondary (passive) node is completely inactive, that is, it does not send or receive any traffic.

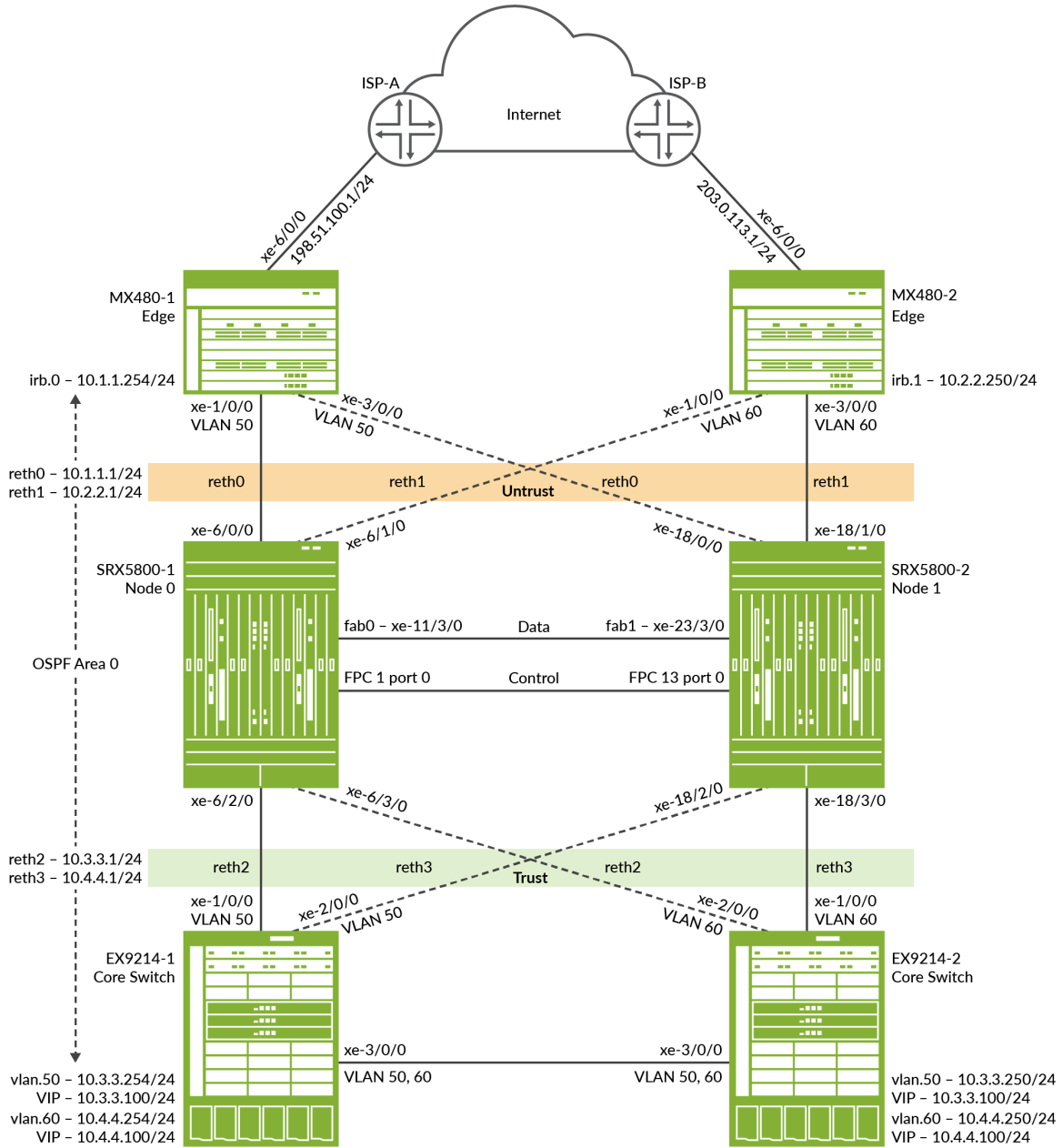
Each reth interface can have one or more logical or subinterfaces (for example, reth 0.0, reth0.1, and so forth). Each must use a different VLAN tag.

The full mesh active/passive chassis cluster consists of two devices:

- One device actively provides routing, firewall, NAT, VPN, and security services, along with maintaining control of the chassis cluster.
- The other device passively maintains its state for cluster failover capabilities should the active device become inactive.

[Figure 47 on page 410](#) shows the topology used in this example.

Figure 47: Full Mesh Active/Passive Chassis Clustering Topology on a Pair of High-End SRX Series Devices



Note: Control plane is active on Node 0 and  
 Data Plane has active reths on both nodes.

— Primary Connection  
 - - - Secondary Connection



## Configuration

### IN THIS SECTION

- [Verification | 423](#)

To configure this example, perform the following procedures:

### Configuring the Control Ports

#### Step-by-Step Procedure

Select FPC 1/13, because the central point (CP) is always on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the central point (for this example, use the SPC in slot 1). You must enter the operational mode commands on both devices.

**NOTE:** Control port configuration is required for SRX5600 and SRX5800 devices.

To configure the control port for each device, and commit the configuration:

1. Configure the control port for the SRX5800-1 (node 0) and commit the configuration.

```
user@SRX5800-1# set chassis cluster control-ports fpc 1 port 0
user@SRX5800-1# set chassis cluster control-ports fpc 13 port 0
user@SRX5800-1# commit and-quit
```

2. Configure the control port for the SRX5800-2 (node 1) and commit the configuration.

```
user@SRX5800-2# set chassis cluster control-ports fpc 1 port 0
user@SRX5800-2# set chassis cluster control-ports fpc 13 port 0
user@SRX5800-2# commit and-quit
```

## Enabling Cluster Mode

### Step-by-Step Procedure

Set the two devices to cluster mode. A reboot is required to enter into cluster mode after the cluster ID and node ID are set. You can cause the system to boot automatically by including the `reboot` parameter in the CLI. You must enter the operational mode commands on both devices. When the system boots, both the nodes come up as a cluster.

**NOTE:** Since there is only a single cluster on the segments, this example uses cluster ID 1 with Device SRX5800-1 as node 0 and Device SRX5800-2 as node 1.

To set the two devices in cluster mode:

1. Enable cluster mode on the SRX5800-1 (node 0).

```
user@SRX5800-1> set chassis cluster cluster-id 1 node 0 reboot
```

2. Enable cluster mode on the SRX5800-2 (node 1).

```
user@SRX5800-2> set chassis cluster cluster-id 1 node 1 reboot
```

**NOTE:** If you have multiple SRX device clusters on a single broadcast domain, make sure that you assign different cluster IDs to each cluster to avoid a MAC address conflict.

The cluster ID is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 1 through 15. Setting a cluster ID to 0 is equivalent to disabling a cluster.

Now the devices are a pair. From this point forward, configuration of the cluster is synchronized between the node members, and the two separate devices function as one device.

## Configuring Cluster Mode

### Step-by-Step Procedure

**NOTE:** In cluster mode, the cluster is synchronized between the nodes when you execute a `commit` command. All commands are applied to both nodes regardless of which device the command is configured on.

To configure a chassis cluster on a high-end SRX Series device:

1. Configure the fabric (data) ports of the cluster that are used to pass real-time objects (RTOs) in active/passive mode. Define two fabric interfaces, one on each chassis, to connect together.

```
user@SRX5800-1# set interfaces fab0 fabric-options member-interfaces xe-11/3/0
user@SRX5800-1# set interfaces fab1 fabric-options member-interfaces xe-23/3/0
```

2. Because the SRX Services Gateway chassis cluster configuration is contained within a single common configuration, use the Junos OS node-specific configuration method called groups to assign some elements of the configuration to a specific member only.

The `set apply-groups ${node}` command uses the node variable to define how the groups are applied to the nodes. Each node recognizes its number and accepts the configuration accordingly. You must also configure out-of-band management on the `fxp0` interface of the SRX5800 Services Gateway using separate IP addresses for the individual control planes of the cluster.

**NOTE:** Configuring the backup router destination address as `x.x.x.0/0` is not allowed.

```
user@SRX5800-1# set groups node0 system host-name SRX5800-1
user@SRX5800-1# set groups node0 system backup-router 10.52.63.254
user@SRX5800-1# set groups node0 system backup-router destination 10.0.0.0/8
user@SRX5800-1# set groups node0 interfaces fxp0 unit 0 family inet address 10.52.43.57/19
user@SRX5800-1# set groups node1 system host-name SRX5800-2
user@SRX5800-1# set groups node1 system backup-router 10.52.63.254
user@SRX5800-1# set groups node1 system backup-router destination 10.0.0.0/8
user@SRX5800-1# set groups node1 interfaces fxp0 unit 0 family inet address 10.52.52.27/19
user@SRX5800-1# set apply-groups "${node}"
```

3. Configure redundancy groups for chassis clustering. Each node has interfaces in a redundancy group where interfaces are active in active redundancy groups (multiple active interfaces can exist in one redundancy group).

Redundancy group 0 controls the control plane and redundancy group 1+ controls the data plane and includes the data plane ports. For any active/passive mode cluster, only redundancy groups 0 and 1 need to be configured. Use four reth interfaces, all of which are members of redundancy group 1. Besides redundancy groups, you must also define:

- Redundant Ethernet Interface count—Configure how many redundant Ethernet interfaces (reth) can possibly be configured so that the system can allocate the appropriate resources for it.
- Priority for control plane and data plane—Define which device has priority (for chassis cluster, high priority is preferred) for the control plane, and which device is preferred to be active for the data plane.

**NOTE:** In active/passive or active/active mode, the control plane (redundancy group 0) can be active on a chassis different from the data plane (redundancy group 1+ and groups) chassis. However, for this example, we recommend having both the control and data plane active on the same chassis member. When traffic passes through the fabric link to go to another member node, latency is introduced.

```
user@SRX5800-1# set chassis cluster reth-count 4
user@SRX5800-1# set chassis cluster redundancy-group 0 node 0 priority 129
user@SRX5800-1# set chassis cluster redundancy-group 0 node 1 priority 128
user@SRX5800-1# set chassis cluster redundancy-group 1 node 0 priority 129
user@SRX5800-1# set chassis cluster redundancy-group 1 node 1 priority 128
```

4. Configure the data interfaces on the platform so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly.

Seamless transition to a new active node occurs with data plane failover. In case of control plane failover, all the daemons are restarted on the new node. Because of this, enabling graceful restart for relevant routing protocols is strongly recommended to avoid losing neighborships with peers. This promotes a seamless transition to the new node without any packet loss.

Define the following items:

- Membership information of the member interfaces to the reth interface.

```

user@SRX5800-1# set interfaces xe-6/0/0 gigether-options redundant-parent reth0
user@SRX5800-1# set interfaces xe-6/1/0 gigether-options redundant-parent reth1
user@SRX5800-1# set interfaces xe-6/2/0 gigether-options redundant-parent reth2
user@SRX5800-1# set interfaces xe-6/3/0 gigether-options redundant-parent reth3
user@SRX5800-1# set interfaces xe-18/0/0 gigether-options redundant-parent reth0
user@SRX5800-1# set interfaces xe-18/1/0 gigether-options redundant-parent reth1
user@SRX5800-1# set interfaces xe-18/2/0 gigether-options redundant-parent reth2
user@SRX5800-1# set interfaces xe-18/3/0 gigether-options redundant-parent reth3

```

- Which redundancy group the reth interface is a member of. For this active/passive example, it is always 1.

```

user@SRX5800-1# set interfaces reth0 redundant-ether-options redundancy-group 1
user@SRX5800-1# set interfaces reth1 redundant-ether-options redundancy-group 1
user@SRX5800-1# set interfaces reth2 redundant-ether-options redundancy-group 1
user@SRX5800-1# set interfaces reth3 redundant-ether-options redundancy-group 1

```

- The reth interface information such as the IP address of the interface.

```

user@SRX5800-1# set interfaces reth0 unit 0 family inet address 10.1.1.1/24
user@SRX5800-1# set interfaces reth1 unit 0 family inet address 10.2.2.1/24
user@SRX5800-1# set interfaces reth2 unit 0 family inet address 10.3.3.1/24
user@SRX5800-1# set interfaces reth3 unit 0 family inet address 10.4.4.1/24

```

## 5. Configure the chassis cluster behavior in case of a failure.

Each interface is configured with a weight value that is deducted from the redundancy group threshold of 255 upon a link loss. The failover threshold is hard coded at 255 and cannot be changed. You can alter an interface link's weight to determine the impact on the chassis failover.

When a redundancy group threshold reaches 0, that redundancy group fails over to the secondary node.

Enter the following commands on the SRX5800-1:

```

user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/2/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/3/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/2/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/3/0 weight 255
user@SRX5800-1# set chassis cluster control-link-recovery
user@SRX5800-1# set chassis cluster redundancy-group 1 preempt

```

This step completes the chassis cluster configuration part of the active/passive mode example for the SRX5800. The rest of this procedure describes how to configure the zone, virtual router, routing, the EX9214, and the MX480 to complete the deployment scenario.

## Configuring Zones, Security Policy, and Protocols

### Step-by-Step Procedure

Configure zones and add the appropriate reth interfaces, and configure OSPF.

To configure zones and OSPF:

1. Configure two zones and add the appropriate reth interfaces .

```

user@SRX5800-1# set security zones security-zone Untrust interfaces reth0.0
user@SRX5800-1# set security zones security-zone Untrust interfaces reth1.0
user@SRX5800-1# set security zones security-zone Trust interfaces reth2.0
user@SRX5800-1# set security zones security-zone Trust interfaces reth3.0

```

2. Permit the appropriate protocols and services to reach interfaces in the Trust and Untrust zones.

```

user@SRX5800-1# set security zones security-zone Trust host-inbound-traffic protocols ospf
user@SRX5800-1# set security zones security-zone Trust host-inbound-traffic system-services
all
user@SRX5800-1# set security zones security-zone Untrust host-inbound-traffic system-services

```

```
ping
```

```
user@SRX5800-1# set security zones security-zone Untrust host-inbound-traffic protocols ospf
```

3. Configure a security policy to permit traffic from the trust zone to the untrust zone.

```
user@SRX5800-1# set security policies from-zone Trust to-zone Untrust policy allow match
source-address any
```

```
user@SRX5800-1# set security policies from-zone Trust to-zone Untrust policy allow match
destination-address any
```

```
user@SRX5800-1# set security policies from-zone Trust to-zone Untrust policy allow match
application any
```

```
user@SRX5800-1# set security policies from-zone Trust to-zone Untrust policy allow then permit
```

```
user@SRX5800-1# set security policies from-zone Untrust to-zone Trust policy allow match
source-address any
```

```
user@SRX5800-1# set security policies from-zone Untrust to-zone Trust policy allow match
destination-address any
```

```
user@SRX5800-1# set security policies from-zone Untrust to-zone Trust policy allow match
application any
```

```
user@SRX5800-1# set security policies from-zone Untrust to-zone Trust policy allow then permit
```

4. Configure OSPF.

```
user@SRX5800-1# set protocols ospf area 0.0.0.0 interface reth0.0
```

```
user@SRX5800-1# set protocols ospf area 0.0.0.0 interface reth1.0
```

```
user@SRX5800-1# set protocols ospf area 0.0.0.0 interface reth2.0
```

```
user@SRX5800-1# set protocols ospf area 0.0.0.0 interface reth3.0
```

## Configuring the EX9214-1

### Step-by-Step Procedure

For the EX9214 Ethernet switches, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive full mesh example for the SRX5800 Services Gateway; most notably the VLANs, routing, and interface configuration.

To configure the EX9214-1:

### 1. Configure the interfaces.

```
user@EX9214-1# set interfaces xe-1/0/0 unit 0 family ethernet-switching interface-mode access
user@EX9214-1# set interfaces xe-1/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH2
user@EX9214-1# set interfaces xe-2/0/0 unit 0 family ethernet-switching interface-mode access
user@EX9214-1# set interfaces xe-2/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH2
user@EX9214-1# set interfaces xe-3/0/0 unit 0 family ethernet-switching interface-mode trunk
user@EX9214-1# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH2
user@EX9214-1# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH3
```

### 2. Configure VRRP between the two EX switches.

```
user@EX9214-1# set interfaces irb unit 50 family inet address 10.3.3.254/24 vrrp-group 1
virtual-address 10.3.3.100
user@EX9214-1# set interfaces irb unit 50 family inet address 10.3.3.254/24 vrrp-group 1
priority 200
user@EX9214-1# set interfaces irb unit 50 family inet address 10.3.3.254/24 vrrp-group 1
accept-data
user@EX9214-1# set interfaces irb unit 60 family inet address 10.4.4.254/24 vrrp-group 2
virtual-address 10.4.4.100
user@EX9214-1# set interfaces irb unit 60 family inet address 10.4.4.254/24 vrrp-group 2
priority 100
user@EX9214-1# set interfaces irb unit 60 family inet address 10.4.4.254/24 vrrp-group 2
accept-data
```

### 3. Configure the VLANs.

```
user@EX9214-1# set vlans SRX5800-RETH2 vlan-id 50
user@EX9214-1# set vlans SRX5800-RETH2 l3-interface irb.50
user@EX9214-1# set vlans SRX5800-RETH3 vlan-id 60
user@EX9214-1# set vlans SRX5800-RETH3 l3-interface irb.60
```



#### 4. Configure the protocols.

```
user@EX9214-1# set protocols ospf area 0.0.0.0 interface irb.50
user@EX9214-1# set protocols ospf area 0.0.0.0 interface irb.60
user@EX9214-1# set protocols rstp interface all
```

## Configuring the EX9214-2

### Step-by-Step Procedure

To configure the EX9214-2:

#### 1. Configure the interfaces.

```
user@EX9214-2# set interfaces xe-1/0/0 unit 0 family ethernet-switching interface-mode access
user@EX9214-2# set interfaces xe-1/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH3
user@EX9214-2# set interfaces xe-2/0/0 unit 0 family ethernet-switching interface-mode access
user@EX9214-2# set interfaces xe-2/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH3
user@EX9214-2# set interfaces xe-3/0/0 unit 0 family ethernet-switching interface-mode trunk
user@EX9214-2# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH2
user@EX9214-2# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-
RETH3
```

#### 2. Configure VRRP between the two EX switches.

```
user@EX9214-2# set interfaces irb unit 50 family inet address 10.3.3.250/24 vrrp-group 1
virtual-address 10.3.3.100
user@EX9214-2# set interfaces irb unit 50 family inet address 10.3.3.250/24 vrrp-group 1
priority 100
user@EX9214-2# set interfaces irb unit 50 family inet address 10.3.3.250/24 vrrp-group 1
accept-data
user@EX9214-2# set interfaces irb unit 60 family inet address 10.4.4.250/24 vrrp-group 2
virtual-address 10.4.4.100
user@EX9214-2# set interfaces irb unit 60 family inet address 10.4.4.250/24 vrrp-group 2
priority 200
```

```
user@EX9214-2# set interfaces irb unit 60 family inet address 10.4.4.250/24 vrrp-group 2
accept-data
```

### 3. Configure the VLANs.

```
user@EX9214-2# set vlans SRX5800-RETH2 vlan-id 50
user@EX9214-2# set vlans SRX5800-RETH2 l3-interface irb.50
user@EX9214-2# set vlans SRX5800-RETH3 vlan-id 60
user@EX9214-2# set vlans SRX5800-RETH3 l3-interface irb.60
```

### 4. Configure the protocols.

```
user@EX9214-2# set protocols ospf area 0.0.0.0 interface irb.50
user@EX9214-2# set protocols ospf area 0.0.0.0 interface irb.60
user@EX9214-2# set protocols rstp interface all
```

## Configure MX480-1

### Step-by-Step Procedure

For the MX480 Edge Routers, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably you must use an IRB interface within a virtual switch instance on the switch.

To configure the MX480-1:

#### 1. Configure the downstream interfaces.

```
user@MX480-1# set interfaces xe-1/0/0 encapsulation ethernet-bridge
user@MX480-1# set interfaces xe-1/0/0 unit 0 family bridge
user@MX480-1# set interfaces xe-3/0/0 encapsulation ethernet-bridge
user@MX480-1# set interfaces xe-3/0/0 unit 0 family bridge
```

#### 2. Configure the upstream interface.

```
user@MX480-1# set interfaces xe-6/0/0 unit 0 family inet address 198.51.100.1/24
```

### 3. Configure the IRB interface.

```
user@MX480-1# set interfaces irb unit 0 family inet address 10.1.1.254/24
```

### 4. Configure a static route and graceful restart.

```
user@MX480-1# set routing-options static route 0.0.0.0/0 next-hop 198.51.100.254
user@MX480-1# set policy-options policy-statement def-route term 1 from protocol static
user@MX480-1# set policy-options policy-statement def-route term 1 from route-filter
0.0.0.0/0 exact
user@MX480-1# set policy-options policy-statement def-route term 1 then accept
```

### 5. Configure the bridge domain.

```
user@MX480-1# set bridge-domains BD-50 vlan-id 50
user@MX480-1# set bridge-domains BD-50 domain-type bridge
user@MX480-1# set bridge-domains BD-50 interface xe-1/0/0
user@MX480-1# set bridge-domains BD-50 interface xe-3/0/0
user@MX480-1# set bridge-domains BD-50 routing-interface irb.0
```

### 6. Configure OSPF.

```
user@MX480-1# set protocols ospf area 0.0.0.0 interface irb.0
user@MX480-1# set protocols ospf area 0.0.0.0 interface xe-6/0/0.0 passive
user@MX480-1# set protocols ospf export def-route
```

## Configuring the MX480-2

### Step-by-Step Procedure

To configure the MX480-2:

1. Configure the downstream interfaces.

```
user@MX480-2# set interfaces xe-1/0/0 encapsulation ethernet-bridge
user@MX480-2# set interfaces xe-1/0/0 unit 0 family bridge
user@MX480-2# set interfaces xe-3/0/0 encapsulation ethernet-bridge
user@MX480-2# set interfaces xe-3/0/0 unit 0 family bridge
```

2. Configure the upstream interface.

```
user@MX480-2# set interfaces xe-6/0/0 unit 0 family inet address 203.0.113.1/24
```

3. Configure the IRB interface.

```
user@MX480-2# set interfaces irb unit 0 family inet address 10.2.2.250/24
```

4. Configure a static route and graceful restart.

```
user@MX480-2# set routing-options static route 0.0.0.0/0 next-hop 203.0.113.254
user@MX480-2# set policy-options policy-statement def-route term 1 from protocol static
user@MX480-2# set policy-options policy-statement def-route term 1 from route-filter
0.0.0.0/0 exact
user@MX480-2# set policy-options policy-statement def-route term 1 then accept
```

5. Configure the bridge domain.

```
user@MX480-2# set bridge-domains BD-60 vlan-id 60
user@MX480-2# set bridge-domains BD-60 domain-type bridge
user@MX480-2# set bridge-domains BD-60 interface xe-1/0/0
user@MX480-2# set bridge-domains BD-60 interface xe-3/0/0
user@MX480-2# set bridge-domains BD-60 routing-interface irb.0
```

6. Configure OSPF.

```
user@MX480-2# set protocols ospf area 0.0.0.0 interface irb.0
```

```
user@MX480-2# set protocols ospf area 0.0.0.0 interface xe-6/0/0.0 passive
user@MX480-2# set protocols ospf export def-route
```

## Configuring Miscellaneous Settings

### Step-by-Step Procedure

This full mesh chassis clustering example for the SRX5800 does not describe in detail miscellaneous configurations such as how to configure NAT, security policies, or VPNs. They are essentially the same as they would be for standalone configurations.

However, if you are performing proxy ARP in chassis cluster configurations, you must apply the proxy ARP configurations to the reth interfaces rather than the member interfaces because the reth interfaces hold the logical configurations.

You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

### Verification

#### IN THIS SECTION

- [Verify Chassis Cluster Status | 424](#)
- [Verify Chassis Cluster Interfaces | 425](#)
- [Verify Chassis Cluster Statistics | 426](#)
- [Verify Chassis Cluster Control Plane Statistics | 428](#)
- [Verifying Chassis Cluster Data Plane Statistics | 429](#)
- [Verifying Chassis Cluster Redundancy Group Status | 431](#)
- [Verify Connection on EX Device | 432](#)
- [Troubleshoot with Logs | 432](#)

To confirm that the configuration is working properly, perform these tasks:

## Verify Chassis Cluster Status

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@SRX5800-1> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring    RE Relinquish monitoring
  IS IRQ storm

Cluster ID: 1
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 129   primary          no    no    None
node1 128   secondary        no    no    None

Redundancy group: 1 , Failover count: 3
node0 129   primary          yes   no    None
node1 128   secondary        yes   no    None
```

### Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

## Verify Chassis Cluster Interfaces

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@SRX5800-1> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      em0       Up                Disabled     Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status                Security
                (Physical/Monitored)
  fab0    xe-11/3/0       Up / Up              Disabled
  fab0
  fab1    xe-23/3/0       Up / Up              Disabled
  fab1

Redundant-ethernet Information:
  Name      Status  Redundancy-group
  reth0     Up      1
  reth1     Up      1
  reth2     Up      1
  reth3     Up      1

Redundant-pseudo-interface Information:
  Name      Status  Redundancy-group
  lo0       Up      0

Interface Monitoring:
  Interface  Weight  Status                Redundancy-group
                (Physical/Monitored)
```

xe-18/3/0	255	Up / Up	1
xe-18/1/0	255	Up / Up	1
xe-6/3/0	255	Up / Up	1
xe-6/1/0	255	Up / Up	1
xe-18/2/0	255	Up / Up	1
xe-6/2/0	255	Up / Up	1
xe-18/0/0	255	Up / Up	1
xe-6/0/0	255	Up / Up	1

## Meaning

The sample output shows each interface's status, weight value, and the redundancy group to which that interface belongs.

## Verify Chassis Cluster Statistics

### Purpose

Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of real-time objects (RTOs) sent and received for services.

### Action

From operational mode, enter the `show chassis cluster statistics` command.

```
{primary:node0}
user@SRX5800-1> show chassis cluster statistics
```

```
Control link statistics:
```

```
Control link 0:
```

```
Heartbeat packets sent: 1191183
```

```
Heartbeat packets received: 1191154
```

```
Heartbeat packet errors: 0
```

```
Fabric link statistics:
```

```
Child link 0
```

```
Probes sent: 2387707
```

```
Probes received: 2387679
```

```
Child link 1
```

```
Probes sent: 0
```

```
Probes received: 0
```



## 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	251	7
IPv6 session create	0	0
IPv4/6 session RT0 ACK	0	0
Session close	230	4
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	1
IPv6 session ageout refresh requests	0	0
Session ageout refresh replies	1	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	0	0

## Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

## Verify Chassis Cluster Control Plane Statistics

### Purpose

Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

### Action

From operational mode, enter the `show chassis cluster control-plane statistics` command.

```
{primary:node0}
user@SRX5800-1> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 1191222
```

```

Heartbeat packets received: 1191193
Heartbeat packet errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 2387785
    Probes received: 2387757
  Child link 1
    Probes sent: 0
    Probes received: 0

```

## Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

## Verifying Chassis Cluster Data Plane Statistics

### Purpose

Verify information about the number of real-time objects (RTOs) sent and received for services.

### Action

From operational mode, enter the `show chassis cluster data-plane statistics` command.

```

{primary:node0}
user@SRX5800-1> show chassis cluster 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	251	7
IPv6 session create	0	0
IPv4/6 session RTO ACK	0	0
Session close	230	4
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	1
IPv6 session ageout refresh requests	0	0
Session ageout refresh replies	1	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	0	0

## Meaning

The sample output shows the RTOs sent and received for various services.

## Verifying Chassis Cluster Redundancy Group Status

### Purpose

Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

### Action

From operational mode, enter the `chassis cluster status redundancy-group` command.

```
{primary:node0}
user@SRX5800-1> show chassis cluster status redundancy-group 1
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring            SM Schedule monitoring
  CF Config Sync monitoring    RE Relinquish monitoring
  IS IRQ storm

Cluster ID: 1
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 1 , Failover count: 3
node0 129   primary      yes   no   None
node1 128   secondary   yes   no   None
```

## Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

## Verify Connection on EX Device

### Purpose

Verify the connection from EX device.

### Action

From operational mode, enter these ping 192.168.1.1 count 2 and traceroute 192.168.1.1 commands.

```
user@@EX1-R3> ping 192.168.1.1 count 2
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=62 time=3.964 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=62 time=20.603 ms

--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.964/12.284/20.603/8.320 ms
```

```
user@@EX1-R3> traceroute 192.168.1.1
traceroute to 192.168.1.1 (192.168.1.1), 30 hops max, 52 byte packets
 1 10.3.3.1 (10.3.3.1) 2.287 ms 1.700 ms 1.978 ms
 2 10.1.1.254 (10.1.1.254) 3.164 ms 2.750 ms 2.537 ms
 3 192.168.1.1 (192.168.1.1) 4.660 ms 4.144 ms 4.473 ms
```

## Troubleshoot with Logs

### Purpose

Look at the system logs to identify any chassis cluster issues. You should look at the system log files on both nodes.

## Action

From operational mode, enter these `show log` commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

## RELATED DOCUMENTATION

No Link Title

No Link Title

# Example: Configuring an Active/Active Layer 3 Cluster Deployment

## IN THIS SECTION

- [Requirements | 433](#)
- [Overview | 434](#)
- [Configuration | 435](#)

This example shows how to set up basic active/active chassis clustering on high-end SRX Series devices.

## Requirements

This example uses the following hardware and software components:

- Two Juniper Networks SRX5800 Services Gateways with identical hardware configurations running Junos OS Release 9.6 or later.
- Two Juniper Networks EX9214 Ethernet Switches running Junos OS Release 9.6 or later.
  - Any EX Series switch can be used here.

Before you begin:

- Physically connect the two SRX Services Gateways (back-to-back for the fabric and control ports).

**NOTE:** This configuration example has been tested using the software release listed and is assumed to work on all later releases.

## Overview

A chassis cluster consists of two SRX Series devices with identical hardware. Active/active clustering on SRX Series devices is supported for those environments that want to maintain traffic on both chassis cluster members whenever possible. In an active/active deployment, only the data plane is in active/active mode; the control plane is in active/passive mode. This allows one control plane to control both chassis members as a single logical device, allowing the control plane to fail over to the other device in case of failure. Having only the data plane in active/active mode allows the data plane to failover independently of the control plane.

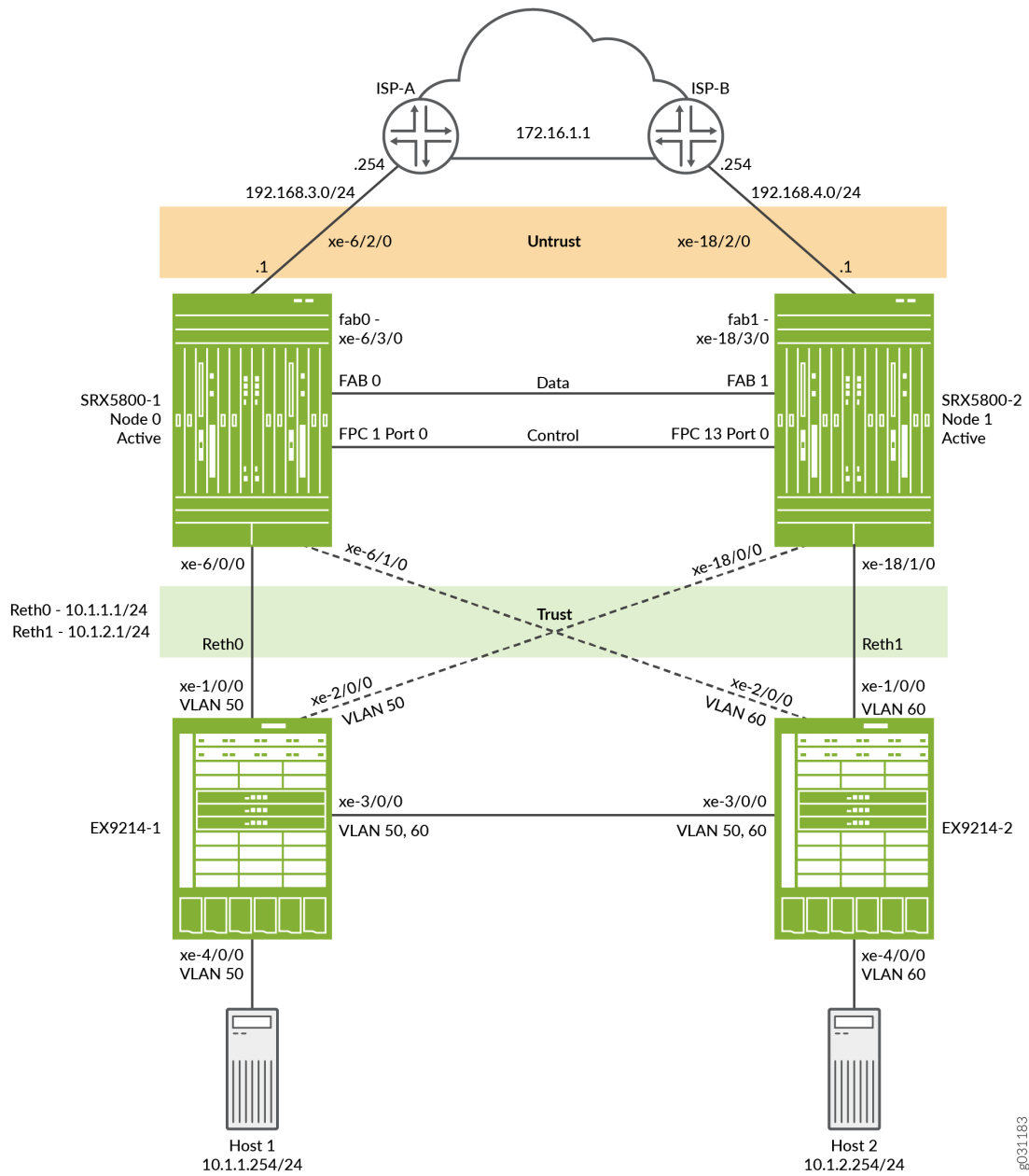
Active/active configuration also allows ingress interfaces to be on one cluster device and the egress interfaces to be on the other. When the ingress and egress interfaces are set up on different devices, the data traffic must pass through the data fabric to the other cluster device and out the egress interface.

This active/active chassis clustering example requires you to configure two redundant Ethernet (reth) interfaces—reth0 and reth1—for each node and ensure they are connected together by one or more switches. A reth interface bundles the two physical interfaces (one from each node) together. A reth interface is assigned to a redundancy group.

[Figure 48 on page 435](#) shows the topology used in this example.



Figure 48: Active/Active Layer 3 Chassis Clustering Topology on a Pair of High-End SRX Series Devices



## Configuration

### IN THIS SECTION

- Verification | 443

To configure this example, perform the following procedures:

## Configuring the Control Ports

### Step-by-Step Procedure

Configure the control port for each device.

Select FPC 1 and FPC 13, because the central point (CP) is always on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the central point (this example uses the SPC in slot 1).

**NOTE:** Control port configuration is required only for SRX5600 and SRX5800 devices.

1. Configure the control ports and commit the configuration:

```
user@host# set chassis cluster control-ports fpc 1 port 0
user@host# set chassis cluster control-ports fpc 13 port 0
user@host# commit and-quit
```

## Enabling Cluster Mode

### Step-by-Step Procedure

Assign a cluster ID and node ID to each device.

Set the two devices to cluster mode by adding a cluster ID and node ID on each and rebooting. You can configure the system to boot automatically by including the `reboot` parameter in the `set` command.

**NOTE:** Since there is only a single cluster on the segments, this example uses cluster ID 1 with Device SRX5800-1 as node 0 and Device SRX5800-2 as node 1.

To set the two devices in cluster mode:

1. Enable cluster mode on SRX5800-1 (node 0).

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

2. Enable cluster mode on SRX5800-2 (node 1).

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

**NOTE:** If you have multiple SRX device clusters on a single broadcast domain, make sure that you assign different cluster IDs to each cluster to avoid MAC address conflicts.

When the system reboots, the nodes come up as a cluster. From this point forward, configuration of the cluster is synchronized between the node members, and the two separate devices function as one device.

## Configuring Cluster Parameters

### Step-by-Step Procedure

**NOTE:** In cluster mode, all commands and configuration are applied to both nodes.

To configure chassis cluster settings:

1. Configure a fabric (data) port on each device to enable traffic to pass from one device to the other for cases where traffic arrives on an ingress interface on one node but leaves on another node.

**NOTE:** A 10-Gigabit Ethernet connection is recommended for active/active deployments.

```
user@host# set interfaces fab0 fabric-options member-interfaces xe-6/3/0  
user@host# set interfaces fab1 fabric-options member-interfaces xe-18/3/0
```

2. Configure each device's fxp0 interface for out-of-band management. Assign a separate IP address for each device (control plane) of the cluster.

Because the SRX Services Gateway chassis cluster configuration is contained within a single common configuration, to assign some elements of the configuration to a specific member only, use the Junos OS node-specific configuration method called groups. The `set apply-groups ${node}` command uses the

node variable to define how the groups are applied to the nodes. Each node recognizes its number and accepts the configuration accordingly.

```
user@host# set groups node0 system host-name SRX5800-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.52.43.57/19
user@host# set groups node1 system host-name SRX5800-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 10.52.52.27/19
user@host# set apply-groups ${node}
```

### 3. Configure redundancy groups for chassis clustering.

Each node has interfaces in a redundancy group. Redundancy group 0 controls the control plane, it defines which node will be the primary. Redundancy group 1+ controls the data plane and includes the data plane ports. This active/active clustering mode example uses 2 reth interfaces with redundancy groups 0, 1, and 2.

As part of redundancy group configuration, you must also define the priority for control plane and data plane—which device is preferred for the control plane, and which device is preferred for the data plane. (For chassis clustering, higher priority is preferred.)

**NOTE:** The control plane (redundancy group 0) and data plane (redundancy group 1+) can each be active on a different chassis. However, for this example, we recommend having both the control and data plane active on the same chassis member. Redundancy group 0 (RG0) and redundancy group 1 (RG1) default to the active state on node 0, whereas, redundancy group 2 (RG2) defaults to active state on node 1.

```
user@host# set chassis cluster redundancy-group 0 node 0 priority 129
user@host# set chassis cluster redundancy-group 0 node 1 priority 128
user@host# set chassis cluster redundancy-group 1 node 0 priority 129
user@host# set chassis cluster redundancy-group 1 node 1 priority 128
user@host# set chassis cluster redundancy-group 2 node 0 priority 128
user@host# set chassis cluster redundancy-group 2 node 1 priority 129
```

### 4. Configure the data interfaces on the platform so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly.

Define the following items:

- The maximum number of reth interfaces for the cluster, so that the system can allocate the appropriate resources for them.

```
user@host# set chassis cluster reth-count 2
```

- The reth interface information such as the IP address of the interface.

```
user@host# set interfaces reth0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces reth1 unit 0 family inet address 10.1.2.1/24
```

- Membership information of the member interfaces to reth interfaces.

```
user@host# set interfaces xe-6/0/0 gigger-options redundant-parent reth0
user@host# set interfaces xe-6/1/0 gigger-options redundant-parent reth1
user@host# set interfaces xe-18/0/0 gigger-options redundant-parent reth0
user@host# set interfaces xe-18/1/0 gigger-options redundant-parent reth1
```

- The mapping of reth interfaces to redundancy groups.

```
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 redundant-ether-options redundancy-group 2
```

## 5. Configure the behavior in case of a failure.

Each interface is configured with a weight value that is deducted from the redundancy group threshold of 255 upon a link loss. When a redundancy group threshold reaches 0, that redundancy group fails over to the secondary node.

**NOTE:** If the control-link-recovery feature is not enabled, a manual reboot is required to bring the secondary node back into sync with the primary node.

```
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
user@host# set chassis cluster redundancy-group 2 interface-monitor xe-6/1/0 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
```

```
user@host# set chassis cluster redundancy-group 2 interface-monitor xe-18/1/0 weight 255
user@host# set chassis cluster control-link-recovery
```

**NOTE:** Individual VLANs on an interface are not monitored. Only interfaces as a whole are monitored.

This step completes the chassis cluster configuration.

6. Configure other interfaces that do not belong to the reth interfaces. These are the upstream interfaces towards the ISPs.

```
user@host# set interface xe-6/2/0 unit 0 family inet address 192.168.3.1/24
user@host# set interface xe-18/2/0 unit 0 family inet address 192.168.4.1/24
```

The following sections describe how to configure zones, security policies, NAT, routing, and the EX8208 Core Switches to complete the deployment scenario.

## Configuring Zones, Policies, NAT, and Routes

### Step-by-Step Procedure

Configure and connect the reth interfaces to the appropriate zones and define a security policy that permits outbound traffic. Also, for this example we will use a default route and NAT to enable end hosts to reach the Internet.

To configure zones, policies, NAT, and routes:

1. Assign the interfaces to the appropriate zones.

```
user@host# set security zones security-zone trust interfaces reth0.0
user@host# set security zones security-zone trust interfaces reth1.0
user@host# set security zones security-zone untrust interfaces xe-6/2/0.0
user@host# set security zones security-zone untrust interfaces xe-18/2/0.0
```

2. Configure a policy to allow traffic from the hosts in the trust zone to the Internet.

```
user@host# set security policies from-zone trust to-zone untrust policy allow match source-address any
```

```

user@host# set security policies from-zone trust to-zone untrust policy allow match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy allow match
application any
user@host# set security policies from-zone trust to-zone untrust policy allow then permit

```

### 3. Configure source NAT for outbound traffic.

```

user@host# set security nat source rule-set internet from zone trust
user@host# set security nat source rule-set internet to zone untrust
user@host# set security nat source rule-set internet rule rule1 match source-address
10.1.0.0/16
user@host# set security nat source rule-set internet rule rule1 then source-nat interface

```

### 4. Define a default static route to enable hosts to reach the Internet.

```

user@host# set routing-options static route 0.0.0.0/0 next-hop 192.168.3.254
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 192.168.4.254
preference 7
user@host# set routing-options static route 10.0.0.0/8 next-hop 10.52.63.254

```

### 5. Configure OSPF.

```

user@host# set protocols ospf area 0.0.0.0 interface reth0.0
user@host# set protocols ospf area 0.0.0.0 interface reth1.0
user@host# set protocols ospf area 0.0.0.0 interface reth2.0
user@host# set protocols ospf area 0.0.0.0 interface reth3.0

```

## Configuring the EX9214-1

### Step-by-Step Procedure

For the EX9214, the following commands provide configuration only as it pertains to this active/active example for the SRX5800, most notably the VLANs, routing, and interface configuration.

### 1. Configure the interfaces.

```
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH0
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH0
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH1
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH0
user@host# set interfaces ge-4/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set interfaces ge-4/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH0
```

**NOTE:** The end host is sending untagged traffic.

### 2. Configure the VLANs.

```
user@host# set vlans SRX5800-RETH0 vlan-id 50
user@host# set vlans SRX5800-RETH1 vlan-id 60
```

### 3. Enable RSTP.

```
user@host# set protocols rstp interface all
```

**NOTE:** In this example, RSTP is not strictly required as there is no Layer 2 loop. However, a typical environment would likely have more switches, which would require the protocol to be enabled.

## Configuring the EX9214-2

### Step-by-Step Procedure

To configure the EX9214-2:



### 1. Configure the interfaces.

```
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH1
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH1
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching interface-mode trunk
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH0
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH1
user@host# set interfaces ge-4/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set interfaces ge-4/0/0 unit 0 family ethernet-switching vlan members SRX5800-RETH1
```

**NOTE:** The end host is sending untagged traffic.

### 2. Configure the VLANs.

```
user@host# set vlans SRX5800-RETH0 vlan-id 50
user@host# set vlans SRX5800-RETH1 vlan-id 60
```

### 3. Enable RSTP.

```
user@host# set protocols rstp interface all
```

**NOTE:** In this example, RSTP is not strictly required as there is no Layer 2 loop. However, a typical environment would likely have more switches, which would require the protocol to be enabled.

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Status | 444](#)
- [Verifying Chassis Cluster Interfaces | 445](#)

- [Verifying Chassis Cluster Statistics | 446](#)
- [Verifying Chassis Cluster Control Plane Statistics | 448](#)
- [Verifying Chassis Cluster Data Plane Statistics | 448](#)
- [Verifying Chassis Cluster Redundancy Group Status | 449](#)
- [Troubleshooting with Logs | 450](#)

Confirm that the configuration is working properly.

## Verifying Chassis Cluster Status

### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host>show chassis cluster status

Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring            SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 1
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 129   primary   no    no    None
node1 128   secondary no    no    None
```

```

Redundancy group: 1 , Failover count: 1
node0 129      primary      no      no      None
node1 128      secondary   no      no      None

Redundancy group: 2 , Failover count: 0
node0 128      secondary   no      no      None
node1 129      primary     no      no      None

```

## Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

## Verifying Chassis Cluster Interfaces

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```

{primary:node0}
user@host>show chassis cluster interfaces

Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
  0      fxp1      Up                Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
          (Physical/Monitored)
fab0     xe-6/3/0        Up / Up
fab0
fab1     xe-18/3/0       Up / Up
fab1

```

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	2

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-6/0/0	255	Up	1
ge-18/0/0	255	Up	1
ge-6/1/0	255	Up	2
ge-18/1/0	255	Up	2

## Meaning

The sample output provides status information about the control and fabric links. It also shows each reth interface's status, weight value, and redundancy group.

## Verifying Chassis Cluster Statistics

### Purpose

Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of real-time objects (RTOs) sent and received for services.

### Action

From operational mode, enter the `show chassis cluster statistics` command.

```
{primary:node0}
user@host> show chassis cluster statistics
```

Control link statistics:

Control link 0:

Heartbeat packets sent: 258689

```

Heartbeat packets received: 258684
Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
Services Synchronized:
Service name                RTOs sent   RTOs received
Translation context         0           0
Incoming NAT                0           0
Resource manager           6           0
Session create             161         0
Session close              148         0
Session change             0           0
Gate create                 0           0
Session ageout refresh requests 0           0
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
RPC ALG                    0           0
RTSP ALG                   0           0
RAS ALG                    0           0
MAC address learning       0           0
GPRS GTP                   0           0

```

## Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

## Verifying Chassis Cluster Control Plane Statistics

### Purpose

Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

### Action

From operational mode, enter the `show chassis cluster control-plane statistics` command.

```
{primary:node0}
user@host>show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

### Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

## Verifying Chassis Cluster Data Plane Statistics

### Purpose

Verify information about the number of real-time objects (RTOs) sent and received for services.

## Action

From operational mode, enter the `show chassis cluster data-plane statistics` command.

```
{primary:node0}
user@host>show chassis cluster data-plane statistics

Services Synchronized:
  Service name           RTOs sent   RTOs received
  Translation context    0           0
  Incoming NAT           0           0
  Resource manager       6           0
  Session create         161        0
  Session close          148        0
  Session change         0           0
  Gate create            0           0
  Session ageout refresh requests 0           0
  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
  RPC ALG                0           0
  RTSP ALG               0           0
  RAS ALG                0           0
  MAC address learning   0           0
  GPRS GTP               0           0
```

## Meaning

The sample output shows the number of RTOs sent and received for the various services.

## Verifying Chassis Cluster Redundancy Group Status

### Purpose

Verify the status of a redundancy group.

## Action

From operational mode, enter the `chassis cluster status redundancy-group 1` command.

```
{primary:node0}
user@host>show chassis cluster status redundancy-group 1

Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring     IP IP monitoring
  LB Loopback monitoring      MB Mbuf monitoring
  NH Nexthop monitoring      NP NPC monitoring
  SP SPU monitoring          SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 1
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 1 , Failover count: 1
node0 129   primary    no    no    None
node1 128   secondary no    no    None
```

## Meaning

The sample output shows that redundancy group 1 is functioning normally, with no preemptions, manual fail overs, or other failures.

## Troubleshooting with Logs

### Purpose

Look at the system log files to identify any chassis cluster issues. You should look at the system log files on both nodes.



## Action

From operational mode, enter these `show log` commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

## Results

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

```
groups {
  node0 {
    system {
      host-name SRX5800-1;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 10.3.5.1/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name SRX5800-2;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 10.3.5.2/24;
          }
        }
      }
    }
  }
}
```



```
xe-6/1/0 {
    gigheter-options {
        redundant-parent reth1;
    }
}
xe-6/2/0 {
    unit 0 {
        family inet {
            address 192.168.3.1/24;
        }
    }
}
xe-18/0/0 {
    gigheter-options {
        redundant-parent reth0;
    }
}
xe-18/1/0 {
    gigheter-options {
        redundant-parent reth1;
    }
}
xe-18/2/0 {
    unit 0 {
        family inet {
            address 192.168.4.1/24;
        }
    }
}
fab0 {
    fabric-options {
        member-interfaces {
            xe-6/3/0;
        }
    }
}
fab1 {
    fabric-options {
        member-interfaces {
            xe-18/3/0;
        }
    }
}
```

```
reth0 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
reth1 {
  redundant-ether-options {
    redundancy-group 2;
  }
  unit 0 {
    family inet {
      address 10.1.2.1/24;
    }
  }
}
}
routing-options {
  static {
    route 0.0.0.0/0 {
      next-hop 192.168.3.254;
      qualified-next-hop 192.168.4.254 {
        preference 7;
      }
    }
  }
}
security {
  nat {
    source {
      rule-set internet {
        from zone trust;
        to zone untrust;
        rule rule1 {
          match {
            source-address 10.1.0.0/16;
          }
          then {
            source-nat {
```

```
        interface;  
    }  
}  
}  
}  
}  
}  
}  
}  
policies {  
    from-zone trust to-zone untrust {  
        policy allow {  
            match {  
                source-address any;  
                destination-address any;  
                application any;  
            }  
            then {  
                permit;  
            }  
        }  
    }  
}  
}  
zones {  
    security-zone trust {  
        interfaces {  
            reth0.0;  
            reth1.0;  
        }  
    }  
    security-zone untrust {  
        interfaces {  
            xe-6/2/0.0;  
            xe-18/2/0.0;  
        }  
    }  
}  
}
```

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

## RELATED DOCUMENTATION

No Link Title

No Link Title

No Link Title

# Multicast Routing and Asymmetric Routing on Chassis Cluster

## IN THIS SECTION

- [Understanding Multicast Routing on a Chassis Cluster | 456](#)
- [Understanding Asymmetric Routing on a Chassis Cluster | 458](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair | 460](#)

Multicast routing support in a *chassis cluster* allows different multicast protocols to send traffic across interfaces to multiple recipients. Asymmetric routing is the situation where packets from source host to destination host but follow a different path than packets from destination host to source host. For more information, see the following topics:

## Understanding Multicast Routing on a Chassis Cluster

### IN THIS SECTION

- [Understanding PIM Data Forwarding | 457](#)
- [Understanding Multicast and PIM Session Synchronization | 457](#)

Multicast routing support across nodes in a *chassis cluster* allows multicast protocols, such as Protocol Independent Multicast (PIM) versions 1 and 2, Internet Group Management Protocol (IGMP), Session Announcement Protocol (SAP), and Distance Vector Multicast Routing Protocol (DVMRP), to send traffic across interfaces in the cluster. Note, however, that the multicast protocols should not be enabled on the chassis management interface (*fxp0*) or on the fabric interfaces (*fab0* and *fab1*). Multicast sessions are

synched across the cluster and maintained during redundant group failovers. During failover, as with other types of traffic, there might be some multicast packet loss.

Multicast data forwarding in a chassis cluster uses the incoming interface to determine whether or not the session remains active. Packets are forwarded to the peer node if a leaf session's outgoing interface is on the peer instead of on the incoming interface's node. Multicast routing on a chassis cluster supports tunnels for both incoming and outgoing interfaces.

Multicast traffic has an upstream (toward source) and downstream (toward subscribers) direction in traffic flows. The devices replicate (fanout) a single multicast packet to multiple networks that contain subscribers. In the chassis cluster environment, multicast packet fanouts can be active on either nodes.

If the incoming interface is active on the current node and backup on the peer node, then the session is active on the current node and backup on the peer node.

Multicast configuration on a chassis cluster is the same as multicast configuration on a standalone device. See the [Multicast Protocols User Guide](#) for more information.

## Understanding PIM Data Forwarding

Protocol Independent Multicast (PIM) is used between devices to track the multicast packets to be forwarded to each other.

A PIM session encapsulates multicast data into a PIM unicast packet. A PIM session creates the following sessions:

- Control session
- Data session

The data session saves the control session ID. The control session and the data session are closed independently. The incoming interface is used to determine whether the PIM session is active or not. If the outgoing interface is active on the peer node, packets are transferred to the peer node for transmission.

## Understanding Multicast and PIM Session Synchronization

Synchronizing multicast and PIM sessions helps to prevent packet loss due to failover because the sessions do not need to be set up again when there is a failover.

In PIM sessions, the control session is synchronized to the backup node, and then the data session is synchronized.

In multicast sessions, the template session is synchronized to the peer node, then all the leaf sessions are synchronized, and finally the template session is synchronized again.

**SEE ALSO**[Chassis Cluster Overview | 2](#)

## Understanding Asymmetric Routing on a Chassis Cluster

**IN THIS SECTION**

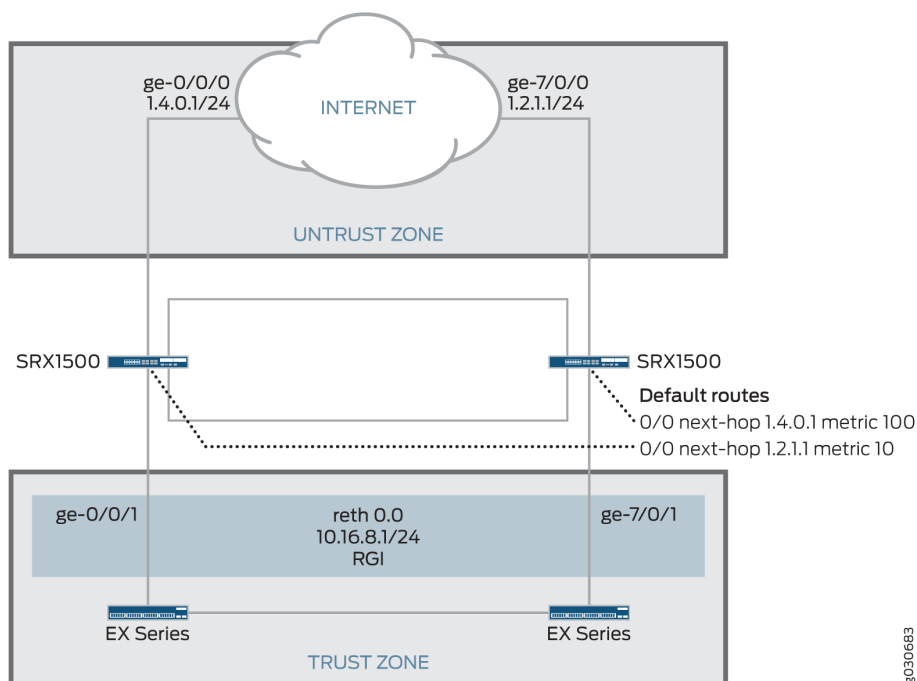
- [Understanding Failures in the Trust Zone Redundant Ethernet Interface | 459](#)
- [Understanding Failures in the Untrust Zone Interfaces | 460](#)

You can use SRX Series Firewalls in chassis clusters asymmetric routing scenarios (see [Figure 49 on page 459](#)). Traffic received by a node is matched against that node's session table. The result of this lookup determines whether or not that the node processes the packet or forwards it to the other node over the fabric link. Sessions are anchored on the egress node for the first packet that created the session. If traffic is received on the node in which the session is not anchored, those packets are forwarded over the fabric link to the node where the session is anchored.

The anchor node for the session can change if there are changes in routing during the session.



Figure 49: Asymmetric Routing Chassis Cluster Scenario



In this scenario, two Internet connections are used, with one being preferred. The connection to the trust zone is done by using a redundant Ethernet interface to provide LAN redundancy for the devices in the trust zone. This scenario describes two failover cases in which sessions originate in the trust zone with a destination of the Internet (untrust zone).

### Understanding Failures in the Trust Zone Redundant Ethernet Interface

Under normal operating conditions, traffic flows from the trust zone interface ge-0/0/1, belonging to reth0.0, to the Internet. Because the primary Internet connection is on node 0, sessions are created in node 0 and synced to node 1. However, sessions are only active on node 0.

A failure in interface ge-0/0/1 triggers a failover of the redundancy group, causing interface ge-7/0/1 in node 1 to become active. After the failover, traffic arrives at node 1. After session lookup, the traffic is sent to node 0 because the session is active on this node. Node 0 then processes the traffic and forwards it to the Internet. The return traffic follows a similar process. The traffic arrives at node 0 and gets processed for security purposes—for example, antispam scanning, antivirus scanning, and application of security policies—on node 0 because the session is anchored to node 0. The packet is then sent to node 1 through the fabric interface for egress processing and eventual transmission out of node 1 through interface ge-7/0/1.

## Understanding Failures in the Untrust Zone Interfaces

In this case, sessions are migrated from node to node. Under normal operating conditions, traffic is processed by only node 0. A failure of interface ge-0/0/0 on node 0 causes a change in the routing table, so that it now points to interface ge-7/0/0 in node 1. After the failure, sessions in node 0 become inactive, and the passive sessions in node 1 become active. Traffic arriving from the trust zone is still received on interface ge-0/0/1, but is forwarded to node 1 for processing. After traffic is processed in node 1, it is forwarded to the Internet through interface ge-7/0/0.

In this chassis cluster configuration, redundancy group 1 is used to control the redundant Ethernet interface connected to the trust zone. As configured in this scenario, redundancy group 1 fails over only if interface ge-0/0/1 or ge-7/0/1 fails, but not if the interfaces connected to the Internet fail. Optionally, the configuration could be modified to permit redundancy group 1 to monitor all interfaces connected to the Internet and fail over if an Internet link were to fail. So, for example, the configuration can allow redundancy group 1 to monitor ge-0/0/0 and make ge-7/0/1 active for reth0 if the ge-0/0/0 Internet link fails. (This option is not described in the following configuration examples.)

### SEE ALSO

[Chassis Cluster Overview | 2](#)

## Example: Configuring an Asymmetric Chassis Cluster Pair

### IN THIS SECTION

- [Requirements | 461](#)
- [Overview | 461](#)
- [Configuration | 465](#)
- [Verification | 472](#)

This example shows how to configure a chassis cluster to allow asymmetric routing. Configuring asymmetric routing for a chassis cluster allows traffic received on either device to be processed seamlessly.

## Requirements

Before you begin:

1. Physically connect a pair of devices together, ensuring that they are the same models. This example uses a pair of SRX1500 or SRX1600 devices.
  - a. To create the fabric link, connect a Gigabit Ethernet interface on one device to another Gigabit Ethernet interface on the other device.
  - b. To create the control link, connect the control port of the two SRX1500 devices.
2. Connect to one of the devices using the console port. (This is the node that forms the cluster.)
  - a. Set the cluster ID and node number.

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

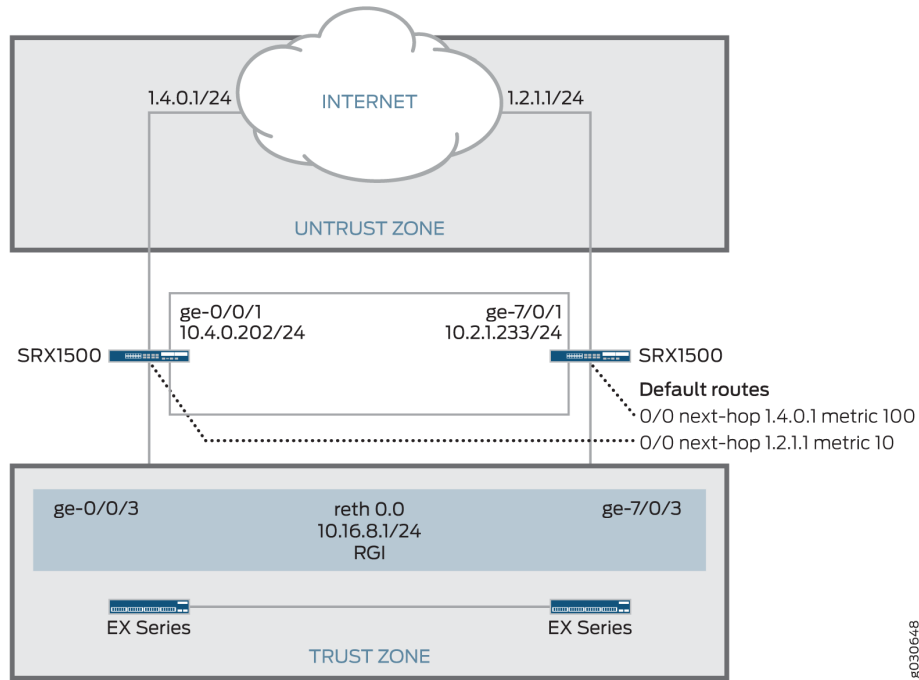
3. Connect to the other device using the console port.
  - a. Set the cluster ID and node number.

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

## Overview

In this example, a chassis cluster provides asymmetric routing. As illustrated in [Figure 50 on page 462](#), two Internet connections are used, with one being preferred. The connection to the trust zone is provided by a redundant Ethernet interface to provide LAN redundancy for the devices in the trust zone.

Figure 50: Asymmetric Routing Chassis Cluster Topology



In this example, you configure group (applying the configuration with the `apply-groups` command) and chassis cluster information. Then you configure security zones and security policies. See [Table 32 on page 462](#) through [Table 35 on page 465](#).

Table 32: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: srxseries-1</li> <li>• Interface: fxp0                             <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.100.50/24</li> </ul> </li> </ul>

**Table 32: Group and Chassis Cluster Configuration Parameters (Continued)**

Feature	Name	Configuration Parameters
	node1	<ul style="list-style-type: none"> <li>• Hostname: srxseries-2</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.100.51/24</li> </ul> </li> </ul>

**Table 33: Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: ge-0/0/7
	fab1	Interface: ge-7/0/7
Heartbeat interval	-	1000
Heartbeat threshold	-	3
Redundancy group	1	<ul style="list-style-type: none"> <li>• Priority:               <ul style="list-style-type: none"> <li>• Node 0: 100</li> <li>• Node 1: 1</li> </ul> </li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>• ge-0/0/3</li> <li>• ge-7/0/3</li> </ul>
Number of redundant Ethernet interfaces	-	1

**Table 33: Chassis Cluster Configuration Parameters (Continued)**

Feature	Name	Configuration Parameters
Interfaces	ge-0/0/1	<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.4.0.202/24</li> </ul>
	ge-7/0/1	<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.2.1.233/24</li> </ul>
	ge-0/0/3	<ul style="list-style-type: none"> <li>Redundant parent: reth0</li> </ul>
	ge-7/0/3	<ul style="list-style-type: none"> <li>Redundant parent: reth0</li> </ul>
	reth0	<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.16.8.1/24</li> </ul>

**Table 34: Security Zone Configuration Parameters**

Name	Configuration Parameters
trust	The reth0.0 interface is bound to this zone.
untrust	The ge-0/0/1 and ge-7/0/1 interfaces are bound to this zone.

**Table 35: Security Policy Configuration Parameters**

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>• Match criteria: <ul style="list-style-type: none"> <li>• source-address any</li> <li>• destination-address any</li> <li>• application any</li> </ul> </li> <li>• Action: permit</li> </ul>

## Configuration

### IN THIS SECTION

- [Procedure | 465](#)

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
{primary:node0}[edit]
set groups node0 system host-name srxseries-1
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.100.50/24
set groups node1 system host-name srxseries-2
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.100.51/24
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces ge-0/0/7
set interfaces fab1 fabric-options member-interfaces ge-7/0/7
set chassis cluster reth-count 1
set chassis cluster heartbeat-interval 1000
```

```

set chassis cluster heartbeat-threshold 3
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-7/0/3 weight 255
set interfaces ge-0/0/1 unit 0 family inet address 1.4.0.202/24
set interfaces ge-0/0/3 gigether-options redundant-parent reth0
set interfaces ge-7/0/1 unit 0 family inet address 10.2.1.233/24
set interfaces ge-7/0/3 gigether-options redundant-parent reth0
set interfaces reth0 unit 0 family inet address 10.16.8.1/24
set routing-options static route 0.0.0.0/0 qualified-next-hop 10.4.0.1 metric 10
set routing-options static route 0.0.0.0/0 qualified-next-hop 10.2.1.1 metric 100
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-7/0/1.0
set security zones security-zone trust interfaces reth0.0
set security policies from-zone trust to-zone untrust policy ANY match source-address any
set security policies from-zone trust to-zone untrust policy ANY match destination-address any
set security policies from-zone trust to-zone untrust policy ANY match application any
set security policies from-zone trust to-zone untrust policy ANY then permit

```

## Step-by-Step Procedure

To configure an asymmetric chassis cluster pair:

1. Configure the management interface.

```

{primary:node0}[edit]
user@host# set groups node0 system host-name srxseries-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 192.168.100.50/24
user@host# set groups node1 system host-name srxseries-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 192.168.100.51/24
user@host# set apply-groups "${node}"

```

2. Configure the fabric interface.

```

{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/7
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/7

```



3. Configure the number of redundant Ethernet interfaces.

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 1
```

4. Configure the redundancy groups.

```
{primary:node0}[edit]
user@host# set chassis cluster heartbeat-interval 1000
user@host# set chassis cluster heartbeat-threshold 3
user@host# set chassis cluster node 0
user@host# set chassis cluster node 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/3 weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/3 weight 255
```

5. Configure the redundant Ethernet interfaces.

```
{primary:node0}[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 1.4.0.202/24
user@host# set interfaces ge-0/0/3 gigether-options redundant-parent reth0
user@host# set interfaces ge-7/0/1 unit 0 family inet address 10.2.1.233/24
user@host# set interfaces ge-7/0/3 gigether-options redundant-parent reth0
user@host# set interfaces reth0 unit 0 family inet address 10.16.8.1/24
```

6. Configure the static routes (one to each ISP, with preferred route through ge-0/0/1).

```
{primary:node0}[edit]
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 10.4.0.1 metric 10
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 10.2.1.1 metric 100
```

7. Configure the security zones.

```
{primary:node0}[edit]
user@host# set security zones security-zone untrust interfaces ge-0/0/1.0
```

```

user@host# set security zones security-zone untrust interfaces ge-7/0/1.0
user@host# set security zones security-zone trust interfaces reth0.0

```

## 8. Configure the security policies.

```

{primary:node0}[edit]
user@host# set security policies from-zone trust to-zone untrust policy ANY match source-
address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match destination-
address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match application
any
user@host# set security policies from-zone trust to-zone untrust policy ANY then permit

```

## Results

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

For brevity, this `show` command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

user@host> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name srxseries-1;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.168.100.50/24;
          }
        }
      }
    }
  }
}
node1 {

```

```

system {
    host-name srxseries-2;
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 192.168.100.51/24;
                }
            }
        }
    }
}
}
apply-groups "${node}";
chassis {
    cluster {
        reth-count 1;
        heartbeat-interval 1000;
        heartbeat-threshold 3;
        redundancy-group 1 {
            node 0 priority 100;
            node 1 priority 1;
            interface-monitor {
                ge-0/0/3 weight 255;
                ge-7/0/3 weight 255;
            }
        }
    }
}
}
interfaces {
    ge-0/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-7/0/3 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {

```

```
        address 10.4.0.202/24;
    }
}
ge-7/0/1 {
    unit 0 {
        family inet {
            address 10.2.1.233/24;
        }
    }
}
fab0 {
    fabric-options {
        member-interfaces {
            ge-0/0/7;
        }
    }
}
fab1 {
    fabric-options {
        member-interfaces {
            ge-7/0/7;
        }
    }
}
reth0 {
    gigheter-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 10.16.8.1/24;
        }
    }
}
...
routing-options {
    static {
        route 0.0.0.0/0 {
            next-hop 10.4.0.1;
            metric 10;
        }
    }
}
```

```
    }  
  }  
  routing-options {  
    static {  
      route 0.0.0.0/0 {  
        next-hop 10.2.1.1;  
        metric 100;  
      }  
    }  
  }  
  security {  
    zones {  
      security-zone untrust {  
        interfaces {  
          ge-0/0/1.0;  
          ge-7/0/1.0;  
        }  
      }  
      security-zone trust {  
        interfaces {  
          reth0.0;  
        }  
      }  
    }  
    policies {  
      from-zone trust to-zone untrust {  
        policy ANY {  
          match {  
            source-address any;  
            destination-address any;  
            application any;  
          }  
          then {  
            permit;  
          }  
        }  
      }  
    }  
  }  
}
```

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

## Verification

### IN THIS SECTION

- [Verifying Chassis Cluster Status | 472](#)
- [Verifying Chassis Cluster Interfaces | 473](#)
- [Verifying Chassis Cluster Statistics | 473](#)
- [Verifying Chassis Cluster Control Plane Statistics | 474](#)
- [Verifying Chassis Cluster Data Plane Statistics | 475](#)
- [Verifying Chassis Cluster Redundancy Group Status | 476](#)
- [Troubleshooting with Logs | 476](#)

Confirm that the configuration is working properly.

### Verifying Chassis Cluster Status

#### Purpose

Verify the chassis cluster status, failover status, and redundancy group information.

#### Action

From operational mode, enter the `show chassis cluster status` command.

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

Redundancy group: 1 , Failover count: 1
node0                100       primary  no       no
node1                1        secondary no       no
```

## Verifying Chassis Cluster Interfaces

### Purpose

Verify information about chassis cluster interfaces.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

Redundant-ethernet Information:
  Name      Status   Redundancy-group
  reth0     Up       1

Interface Monitoring:
  Interface   Weight   Status   Redundancy-group
  ge-0/0/3    255     Up       1
  ge-7/0/3    255     Up       1
```

## Verifying Chassis Cluster Statistics

### Purpose

Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

### Action

From operational mode, enter the `show chassis cluster statistics` command.

```
{primary:node0}
user@host> show chassis cluster statistics

Control link statistics:
  Control link 0:
```

```

Heartbeat packets sent: 228
Heartbeat packets received: 2370
Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 2272
    Probes received: 597
Services Synchronized:
Service name                RTOs sent   RTOs received
Translation context          0           0
Incoming NAT                 0           0
Resource manager            6           0
Session create              160         0
Session close               147         0
Session change              0           0
Gate create                 0           0
Session ageout refresh requests 0           0
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
RPC ALG                     0           0
RTSP ALG                    0           0
RAS ALG                     0           0
MAC address learning        0           0
GPRS GTP                    0           0

```

## Verifying Chassis Cluster Control Plane Statistics

### Purpose

Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).



## Action

From operational mode, enter the `show chassis cluster control-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics
```

```
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

## Verifying Chassis Cluster Data Plane Statistics

### Purpose

Verify information about the number of RTOs sent and received for services.

### Action

From operational mode, enter the `show chassis cluster data-plane statistics` command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics
```

Services Synchronized:		
Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	160	0
Session close	147	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0

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
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

## Verifying Chassis Cluster Redundancy Group Status

### Purpose

Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

### Action

From operational mode, enter the chassis cluster status redundancy-group command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
  Node           Priority  Status  Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
node0           100      primary no        no
node1           1        secondary no        no
```

## Troubleshooting with Logs

### Purpose

Use these logs to identify any chassis cluster issues. You must run these logs on both nodes.

## Action

From operational mode, enter these `show` commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

## RELATED DOCUMENTATION

[Chassis Cluster Overview | 2](#)

[Configuring Chassis Clustering on SRX Series Devices | 120](#)

# Ethernet Switching on Chassis Cluster

## IN THIS SECTION

- [Layer 2 Ethernet Switching Capability in a Chassis Cluster Mode | 478](#)
- [Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device | 479](#)

You can configure a chassis cluster to act as a Layer 2 Ethernet switch. For more information, see the following topics:

## Layer 2 Ethernet Switching Capability in a Chassis Cluster Mode

### IN THIS SECTION

- [Understanding Layer 2 Ethernet Switching Capability in a Chassis Cluster on SRX Series Devices | 478](#)
- [Understanding Chassis Cluster Failover and New Primary Election | 478](#)
- [Benefits of Ethernet Switching on Chassis Cluster | 479](#)

### Understanding Layer 2 Ethernet Switching Capability in a Chassis Cluster on SRX Series Devices

Ethernet ports support various Layer 2 features such as spanning-tree protocols (STPs), IEEE 802.1x, Link Layer Discovery Protocol (LLDP), and Multiple VLAN Registration Protocol (MVRP). With the extension of Layer 2 switching capability to devices in a chassis cluster, you can use Ethernet switching features on both nodes of a chassis cluster.

To ensure that Layer 2 switching works seamlessly across chassis cluster nodes, a dedicated physical link connecting the nodes is required. This type of link is called a *switching fabric interface*. Its purpose is to carry Layer 2 traffic between nodes.

- Configuring a LAG with `family ethernet-switching` is not supported.
- Configuring a Reth with `family ethernet-switching` is not supported. This is only supported in Transparent mode.
- If a switching fabric interface (swfab) is not configured on both nodes, and if you try to configure Ethernet switching related features on the nodes, then the behavior of the nodes might be unpredictable.

### Understanding Chassis Cluster Failover and New Primary Election

When chassis cluster failover occurs, a new primary node is elected and the Ethernet switching process (eswd) runs in a different node. During failover, the chassis control subsystem is restarted. Also during failover, traffic outage occurs until the PICs are up and the VLAN entries are reprogrammed. After failover, all Layer 2 protocols reconverge because Layer 2 protocol states are not maintained in the secondary node.

The Q-in-Q feature in chassis cluster mode is not supported because of chip limitation for swfab interface configuration in Broadcom chipsets.

## Benefits of Ethernet Switching on Chassis Cluster

- Enables Ethernet switching functionality on both nodes of a chassis cluster and provides the option to configure the Ethernet ports on either node for family Ethernet switching.
- Enables configuring a Layer 2 VLAN domain with member ports from both nodes and the Layer 2 switching protocols on both devices.

### SEE ALSO

[Ethernet Switching and Layer 2 Transparent Mode Overview](#)

*Understanding Mixed Mode (Transparent and Route Mode) on Security Devices*

## Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device

### IN THIS SECTION

- [Requirements | 479](#)
- [Overview | 480](#)
- [Configuration | 480](#)

This example shows how to configure switching fabric interfaces to enable switching in chassis cluster mode.

### Requirements

- The physical link used as the switch fabric member must be directly connected to the device.
- Switching fabric interfaces must be configured on ports that support switching features. See [Ethernet Ports Switching Overview for Security Devices](#) for information about the ports on which switching features are supported.

The physical link used as the switch fabric member must be directly connected to the device. Switching supported ports must be used for switching fabric interfaces. See [Ethernet Ports Switching Overview for Security Devices](#) for switching supported ports.

Before you begin, See ["Example: Configuring the Chassis Cluster Fabric Interfaces"](#) on page 65.

## Overview

In this example, pseudointerfaces swfab0 and swfab1 are created for Layer 2 fabric functionality. You also configure dedicated Ethernet ports on each node to be associated with the swfab interfaces.

## Configuration

### IN THIS SECTION

- [Verification | 481](#)

## Procedure

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces swfab0 fabric-options member-interfaces ge-0/0/3
set interfaces swfab1 fabric-options member-interfaces ge-9/0/3
```

### Step-by-Step Procedure

To configure swfab interfaces:

1. Configure swfab0 and swfab1 and associate these switch fabric interfaces to enable switching across the nodes. Note that swfab0 corresponds to node 0 and swfab1 corresponds to node 1.

```
{primary:node0} [edit]
user@host# set interfaces swfab0 fabric-options member-interfaces ge-0/0/3
user@host# set interfaces swfab1 fabric-options member-interfaces ge-9/0/3
```

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

```
{primary:node0} [edit]  
user@host# commit
```

## Results

From configuration mode, confirm your configuration by entering the `show interfaces swfab0` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct the configuration.

```
[edit]  
user@host# show interfaces swfab0  
fabric-options{  
  member-interfaces {  
    ge-0/0/3;  
  }  
}
```

## Verification

### IN THIS SECTION

- [Verifying Switching Fabric Ports | 481](#)

To confirm that the configuration is working properly, perform these tasks:

### *Verifying Switching Fabric Ports*

#### **Purpose**

Verify that you are able to configure multiple ports as members of switching fabric ports.

## Action

From configuration mode, enter the **show interfaces swfab0** command to view the configured interfaces for each port.

```
user@host# show interfaces swfab0
fabric-options{
  member-interfaces {
    ge-0/0/3;
  }
}
```

From operational mode, enter the **show chassis cluster ethernet-switching interfaces** command to view the appropriate member interfaces.

```
user@host> show chassis cluster ethernet-switching interfaces
swfab0:

  Name           Status
  ge-0/0/3       up
swfab1:

  Name           Status
  ge-9/0/3       up
```

## SEE ALSO

| [SRX Series Chassis Cluster Configuration Overview](#) | 14

## RELATED DOCUMENTATION

| [Configuring Chassis Clustering on SRX Series Devices](#) | 120



# Media Access Control Security (MACsec) on Chassis Cluster

## IN THIS SECTION

- [Understanding Media Access Control Security \(MACsec\) | 483](#)
- [Configure Media Access Control Security \(MACsec\) | 486](#)

Media Access Control Security (MACsec) is an industry-standard security technology that provides secure communication for all traffic on Ethernet links. For more information, see the following topics:

## Understanding Media Access Control Security (MACsec)

### IN THIS SECTION

- [How MACsec Works | 484](#)
- [Understanding Connectivity Associations and Secure Channels | 485](#)
- [Understanding Static Connectivity Association Key Security Mode | 485](#)
- [MACsec Considerations | 486](#)

Media Access Control Security (MACsec) is an industry-standard security technology that provides secure communication for all traffic on Ethernet links. MACsec provides point-to-point security on Ethernet links between directly connected nodes and is capable of identifying and preventing most security threats, including denial of service, intrusion, man-in-the-middle, masquerading, passive wiretapping, and playback attacks.

MACsec allows you to secure an Ethernet link for almost all traffic, including frames from the Link Layer Discovery Protocol (LLDP), Link Aggregation Control Protocol (LACP), Dynamic Host Configuration Protocol (DHCP), Address Resolution Protocol (ARP), and other protocols that are not typically secured on an Ethernet link because of limitations with other security solutions. MACsec can be used in

combination with other security protocols such as IP Security (IPsec) and Secure Sockets Layer (SSL) to provide end-to-end network security.

Starting in Junos OS Release 15.1X49-D60, Media Access Control Security(MACsec) is supported on control and fabric ports of SRX340 and SRX345 devices in chassis cluster mode.

Starting in Junos OS Release 20.1R1, MACsec is supported on control ports, fabric ports and revenue ports of SRX380 devices in chassis cluster mode to secure the traffic. MACsec is supported on 16X1Gigabit Ethernet ports (ge-0/0/0 to ge-0/0/15) and 4X10Gigabit Ethernet ports (xe-0/0/16 to xe-0/0/19).

Starting in Junos OS Release 17.4R1, MACsec is supported on HA control and fabric ports of SRX4600 devices in chassis cluster mode.

For SRX1600, SRX2300, and SRX4300 devices, MACsec is supported on dual control ports in chassis cluster mode.

This topic contains the following sections:

## How MACsec Works

To determine if a feature is supported by a specific platform or Junos OS release, see [Feature Explorer](#).

MACsec provides industry-standard security through the use of secured point-to-point Ethernet links. The point-to-point links are secured after matching security keys. When you enable MACsec using static connectivity association key (CAK) security mode, user-configured pre-shared keys are exchanged and verified between the interfaces at each end of the point-to-point Ethernet link.

Once MACsec is enabled on a point-to-point Ethernet link, all traffic traversing the link is MACsec-secured through the use of data integrity checks and, if configured, encryption.

The data integrity checks verify the integrity of the data. MACsec appends an 8-byte header and a 16-byte tail to all Ethernet frames traversing the MACsec-secured point-to-point Ethernet link, and the header and tail are checked by the receiving interface to ensure that the data was not compromised while traversing the link. If the data integrity check detects anything irregular about the traffic, the traffic is dropped.

MACsec can also be used to encrypt all traffic on the Ethernet link. The encryption used by MACsec ensures that the data in the Ethernet frame cannot be viewed by anybody monitoring traffic on the link.

Encryption is enabled for all traffic entering or leaving the interface when MACsec is enabled using static CAK security mode, by default.

MACsec is configured on point-to-point Ethernet links between MACsec-capable interfaces. If you want to enable MACsec on multiple Ethernet links, you must configure MACsec individually on each point-to-point Ethernet link.

## Understanding Connectivity Associations and Secure Channels

MACsec is configured in connectivity associations. MACsec is enabled when a connectivity association is assigned to an interface.

When you enable MACsec using static CAK or dynamic security mode, you have to create and configure a connectivity association. Two secure channels—one secure channel for inbound traffic and another secure channel for outbound traffic—are automatically created. The automatically-created secure channels do not have any user-configurable parameters; all configuration is done in the connectivity association outside of the secure channels.

## Understanding Static Connectivity Association Key Security Mode

When you enable MACsec using static connectivity association key (CAK) security mode, two security keys—a connectivity association key (CAK) that secures control plane traffic and a randomly-generated secure association key (SAK) that secures data plane traffic—are used to secure the point-to-point Ethernet link. Both keys are regularly exchanged between both devices on each end of the point-to-point Ethernet link to ensure link security.

You initially establish a MACsec-secured link using a pre-shared key when you are using static CAK security mode to enable MACsec. A pre-shared key includes a connectivity association name (CKN) and its own connectivity association key (CAK). The CKN and CAK are configured by the user in the connectivity association and must match on both ends of the link to initially enable MACsec.

Once matching pre-shared keys are successfully exchanged, the MACsec Key Agreement (MKA) protocol is enabled. The MKA protocol is responsible for maintaining MACsec on the link, and decides which switch on the point-to-point link becomes the key server. The key server then creates an SAK that is shared with the switch at the other end of the point-to-point link only, and that SAK is used to secure all data traffic traversing the link. The key server will continue to periodically create and share a randomly-created SAK over the point-to-point link for as long as MACsec is enabled.

You enable MACsec using static CAK security mode by configuring a connectivity association on both ends of the link. All configuration is done within the connectivity association but outside of the secure channel. Two secure channels—one for inbound traffic and one for outbound traffic—are automatically created when using static CAK security mode. The automatically-created secure channels do not have any user-configurable parameters that cannot already be configured in the connectivity association.

We recommend enabling MACsec using static CAK security mode. Static CAK security mode ensures security by frequently refreshing to a new random security key and by only sharing the security key between the two devices on the MACsec-secured point-to-point link. Additionally, some optional MACsec features—replay protection, SCI tagging, and the ability to exclude traffic from MACsec—are only available when you enable MACsec using static CAK security mode.

Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, for SRX Series Firewalls supporting MACsec on HA control and fabric links, if the command `restart 802.1x-protocol-daemon` is run on

the primary node, the chassis cluster control and fabric links will flap causing the cluster nodes to enter into split brain mode.

## MACsec Considerations

All types of Spanning Tree Protocol frames cannot currently be encrypted using MACsec.

The connectivity association can be defined anywhere, either global or node specific or any other configuration group as long as it is visible to the MACsec interface configuration.

For MACsec configurations, identical configurations must exist on both the ends. That is, each node should contain the same configuration as the other node. If the other node is not configured or improperly configured with MACsec on the other side, the port is disabled and stops forwarding the traffic.

SRX340, SRX345, and SRX380 devices support MACsec for host-to-host or switch-to-host connections.

SRX4600 devices currently do not support MACsec for host-to-host connections. Macsec is supported only on dedicated fab ports and is not supported if any other traffic port is used as fab.

On SRX340, SRX345, and SRX380 devices, fabric interfaces must be configured such that the Media Access Control Security (MACsec) configurations are local to the nodes. Otherwise, the fabric link will not be reachable.

## Configure Media Access Control Security (MACsec)

### IN THIS SECTION

- [Configuration Considerations When Configuring MACsec on Chassis Cluster Setup | 487](#)
- [Configure MACsec Using Static Connectivity Association Key Security Mode | 488](#)
- [Configure Static CAK on the Chassis Cluster Control Port | 494](#)
- [Configure Static CAK on the Chassis Cluster Fabric Port | 495](#)
- [Configure Static CAK on the Control Port for SRX1600, SRX2300, and SRX4300 Devices | 495](#)
- [Configure Static CAK on the Control Port for SRX4600 | 499](#)
- [Verify MACSEC Configuration | 500](#)

This topic shows how to configure MACsec on control and fabric ports of supported SRX Series Firewall in chassis cluster to secure point-to-point Ethernet links between the peer devices in a cluster. Each point-to-point Ethernet link that you want to secure using MACsec must be configured independently. You can enable MACsec encryption on device-to-device links using static connectivity association key (CAK) security mode.

The configuration steps for both processes are provided in this document.

## Configuration Considerations When Configuring MACsec on Chassis Cluster Setup

Before you begin, follow these steps to configure MACsec on control ports:

1. If the chassis cluster is already up, disable it by using the `set chassis cluster disable` command and reboot both nodes.
2. Configure MACsec on the control port with its attributes as described in the following sections "[Configure Static CAK on the Chassis Cluster Control Port](#)" on page 494. Both nodes must be configured independently with identical configurations.
3. Enable the chassis cluster by using `set chassis cluster cluster-id id` on both of the nodes. Reboot both nodes.

Control port states affect the integrity of a chassis cluster. Consider the following when configuring MACsec on control ports:

- Any new MACsec chassis cluster port configurations or modifications to existing MACsec chassis cluster port configurations will require the chassis cluster to be disabled and displays a warning message `Modifying cluster control port CA will break chassis cluster`. Once disabled, you can apply the preceding configurations and enable the chassis cluster.
- By default, chassis clusters synchronize all configurations. Correspondingly, you must monitor that synchronization does not lead to loss of any MACsec configurations. Otherwise, the chassis cluster will break. For example, for nonsymmetric, node-specific MACsec configurations, identical configurations should exist on both ends. That is, each node should contain the same configuration as the other node.

The ineligible timer is 300 seconds when MACsec on the chassis cluster control port is enabled on SRX340, SRX345, and SRX380 devices.

If both control link fail, Junos OS changes the operating state of the secondary node to ineligible for a 180 seconds. When MACsec is enabled on the control port, the ineligibility duration is 200 seconds for SRX4600 devices.

The initial hold timer is extended from 30 seconds to 120 seconds in chassis clusters on SRX340, SRX345, and SRX380 devices.

For any change in the MACsec configurations of control ports, the steps mentioned above must be repeated.

Consider the following when configuring MACsec on fabric ports:

Configuring MACsec leads to link state changes that can affect traffic capability of the link. When you configure fabric ports, keep the effective link state in mind. Incorrect MACsec configuration on both ends of the fabric links can move the link to an ineligible state. Note the following key points about configuring fabric links:

- Both ends of the links must be configured simultaneously when the chassis cluster is formed.
- Incorrect configuration can lead to fabric failures and errors in fabric recovery logic.

Because of potential link failure scenarios, we recommend that fabric links be configured during formation of the chassis cluster.

### Configure MACsec Using Static Connectivity Association Key Security Mode

You can enable MACsec encryption by using static connectivity association key (CAK) security mode on a point-to-point Ethernet link connecting devices. This procedure shows you how to configure MACsec using static CAK security mode.

For SRX340 and SRX345 devices, ge-0/0/0 is a fabric port and ge-0/0/1 is a control port for the chassis cluster and assigned as cluster-control-port 0.

For configuring MACsec on cluster-control-port and cluster-data-port on SRX380 devices, the node must be in standalone node. The MACsec configuration is applied on both the nodes and the nodes are rebooted to chassis cluster mode.

For SRX380 devices, ge-0/0/0 is a fabric port and ge-0/0/15 is a control port for the chassis cluster..

For SRX4600 devices, dedicated control and fabric ports are available. MACsec on control link can be configured on dedicated control ports (control port 0 [em0] and port 1 [em1]). Macsec on fabric links can be configured only on dedicated fabric ports port 2 and port 3 of fpc0 pic0 (e.g. xe-0/0/2 and xe-0/0/3), similarly on port-2 and port-3 of fpc7 pic0.

For SRX1600 devices, dedicated dual control ports em0/em1 and dual fabric ports are available.

For SRX2300 devices, dual control ports em0/em1 and dual fabric ports are available.

Both SRX1600, SRX2300, and SRX4300 devices support dual control ports with MACsec configuration.

MACsec on dual control link is configured on control port 0 [em0] and control port 1 [em1]. MACsec configured on revenue interfaces is used for forming fabric links. Fabric links are configured on fabric ports (mge-0/0/1 and mge-7/0/1).

To configure MACsec by using static CAK security mode to secure a device-to-device Ethernet link:

1. Create a connectivity association. You can skip this step if you are configuring an existing connectivity association.

```
[edit security macsec]
user@host# set connectivity-association connectivity-association-name
```

For instance, to create a connectivity association named ca1, enter:

```
[edit security macsec]
user@host# set connectivity-association ca1
```

2. Configure the MACsec security mode as static-cak for the connectivity association.

```
[edit security macsec]
user@host# set connectivity-association connectivity-association-name security-mode static-cak
```

For instance, to configure the MACsec security mode to static-cak on connectivity association ca1:

```
[edit security macsec]
user@host# set connectivity-association connectivity-association-name security-mode static-cak
```

3. Create the preshared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

```
[edit security macsec]
user@host# set connectivity-association connectivity-association-name pre-shared-key ckn hexadecimal-number
user@host# set connectivity-association connectivity-association-name pre-shared-key cak hexadecimal-number
```

A preshared key is exchanged between directly-connected links to establish a MACsec-secure link. The pre-shared-key includes the CKN and the CAK. The CKN is a 64-digit hexadecimal number and the CAK is a 64-digit hexadecimal number. The CKN and the CAK must match on both ends of a link to create a MACsec-secured link.

To maximize security, we recommend configuring all 64 digits of a CKN and all 64 digits of a CAK.

After the preshared keys are successfully exchanged and verified by both ends of the link, the MACsec Key Agreement (MKA) protocol is enabled and manages the secure link. The MKA protocol then elects one of the two directly-connected devices as the key server. The key server then shares a random security with the other device over the MACsec-secure point-to-point link. The key server will continue to periodically create and share a random security key with the other device over the MACsec-secured point-to-point link as long as MACsec is enabled.

To configure a CKN of 11c1c1c11xxx012xx5xx8ef284aa23ff6729xx2e4xxx66e91fe34ba2cd9fe311 and CAK of 228xx255aa23xx6729xx664xxx66e91f on connectivity association ca1:

```
[edit security macsec]
user@host# set connectivity-association ca1 pre-shared-key ckn
11c1c1c11xxx012xx5xx8ef284aa23ff6729xx2e4xxx66e91fe34ba2cd9fe311
user@host# set connectivity-association ca1 pre-shared-key cak
228xx255aa23xx6729xx664xxx66e91f
```

MACsec is not enabled until a connectivity association is attached to an interface. See the final step of this procedure to attach a connectivity association to an interface.

4. (Optional) Set the MKA key server priority.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set mka key-server-priority priority-number
```

Specifies the key server priority used by the MKA protocol to select the key server. The device with the lower *priority-number* is selected as the key server.

The default *priority-number* is 16.

If the key-server-priority is identical on both sides of the point-to-point link, the MKA protocol selects the interface with the lower MAC address as the key server. Therefore, if this statement is not configured in the connectivity associations at each end of a MACsec-secured point-to-point link, the interface with the lower MAC address becomes the key server.

To change the key server priority to 0 to increase the likelihood that the current device is selected as the key server when MACsec is enabled on the interface using connectivity association ca1:

```
[edit security macsec connectivity-association ca1]
user@host# set mka key-server-priority 0
```



To change the key server priority to 255 to decrease the likelihood that the current device is selected as the key server in connectivity association ca1:

```
[edit security macsec connectivity-association ca1]
user@host# set mka key-server-priority 255
```

5. (Optional) Set the MKA transmit interval.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set mka transmit-interval interval
```

The MKA transmit interval setting sets the frequency for how often the MKA protocol data unit (PDU) is sent to the directly connected device to maintain MACsec connectivity on the link. A lower *interval* increases bandwidth overhead on the link; a higher *interval* optimizes MKA protocol communication.

The default *interval* is 2000 milliseconds. We recommend increasing the interval to 6000 ms in high-traffic load environments. The transmit interval settings must be identical on both ends of the link when MACsec using static CAK security mode is enabled.

For SRX340, SRX345, and SRX4600 devices, the default MKA transmit interval is 10000 ms on HA links.

For instance, if you wanted to increase the MKA transmit interval to 6000 milliseconds when connectivity association ca1 is attached to an interface:

```
[edit security macsec connectivity-association ca1]
user@host# set mka transmit-interval 6000
```

6. (Optional) Disable MACsec encryption.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set no-encryption
```

Encryption is enabled for all traffic entering or leaving the interface when MACsec is enabled using static CAK security mode, by default.

When encryption is disabled, traffic is forwarded across the Ethernet link in clear text. You are able to view unencrypted data in the Ethernet frame traversing the link when you are monitoring it. The MACsec header is still applied to the frame, however, and all MACsec data integrity checks are run on both ends of the link to ensure the traffic sent or received on the link has not been tampered with and does not represent a security threat.

7. (Optional) Set an offset for all packets traversing the link.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set offset (0 | 30 | 50)
```

For instance, if you wanted to set the offset to 30 in the connectivity association named ca1:

```
[edit security macsec connectivity-association ca1]
user@host# set offset 30
```

The default offset is 0. All traffic in the connectivity association is encrypted when encryption is enabled and an offset is not set.

When the offset is set to 30, the IPv4 header and the TCP/UDP header are unencrypted while encrypting the rest of the traffic. When the offset is set to 50, the IPv6 header and the TCP/UDP header are unencrypted while encrypting the rest of the traffic.

You would typically forward traffic with the first 30 or 50 octets unencrypted if a feature needed to see the data in the octets to perform a function, but you otherwise prefer to encrypt the remaining data in the frames traversing the link. Load balancing features, in particular, typically need to see the IP and TCP/UDP headers in the first 30 or 50 octets to properly load balance traffic.

8. (Optional) Enable replay protection.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set replay-protect replay-window-size number-of-packets
```

When MACsec is enabled on a link, an ID number is assigned to each packet on the MACsec-secured link.

When replay protection is enabled, the receiving interface checks the ID number of all packets that have traversed the MACsec-secured link. If a packet arrives out of sequence and the difference between the packet numbers exceeds the replay protection window size, the packet is dropped by the receiving interface. For instance, if the replay protection window size is set to five and a packet assigned the ID of 1006 arrives on the receiving link immediately after the packet assigned the ID of 1000, the packet that is assigned the ID of 1006 is dropped because it falls outside the parameters of the replay protection window.

Replay protection is especially useful for fighting man-in-the-middle attacks. A packet that is replayed by a man-in-the-middle attacker on the Ethernet link will arrive on the receiving link out of sequence, so replay protection helps ensure the replayed packet is dropped instead of forwarded through the network.

Replay protection should not be enabled in cases where packets are expected to arrive out of order.

You can require that all packets arrive in order by setting the replay window size to 0.

To enable replay protection with a window size of five on connectivity association ca1:

```
[edit security macsec connectivity-association ca1]
user@host# set replay-protect replay-window-size 5
```

9. (Optional) Exclude a protocol from MACsec.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set exclude-protocol protocol-name
```

For instance, if you did not want Link Level Discovery Protocol (LLDP) to be secured using MACsec:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set exclude-protocol lldp
```

When this option is enabled, MACsec is disabled for all packets of the specified protocol—in this case, LLDP—that are sent or received on the link.

10. Assign the connectivity association to a chassis cluster control interface.

```
[edit security macsec]
user@host# set cluster-control-port port-no connectivity-association connectivity-association-name
```

Assigning the connectivity association to an interface is the final configuration step for enabling MACsec on an interface.

For instance, to assign connectivity association ca1 to interface ge-0/0/1 (For SRX340/SRX345):

For instance, to assign connectivity association ca1 to interface ge-0/0/0 (For SRX380):

```
[edit security macsec]
user@host# set cluster-control-port interfaces ge-0/0/1 connectivity-association ca1
```

11. Assign a connectivity association for enabling MACsec on a chassis cluster fabric interface.

```
[edit security macsec]
user@host# set cluster-data-port port-number connectivity-association connectivity-association-name
[edit security macsec]
user@host# set cluster-data-port interfaces ge-5/0/2 connectivity-association ca1
```

MACsec using static CAK security mode is not enabled until a connectivity association on the opposite end of the link is also configured, and contains preshared keys that match on both ends of the link.

### Configure Static CAK on the Chassis Cluster Control Port

To establish a CA over a chassis cluster control link on two SRX345 devices.

1. Configure the MACsec security mode as static-cak for the connectivity association:

```
[edit security macsec]
user@host# set connectivity-association ca1 security-mode static-cak
```

2. Create the preshared key by configuring the connectivity association key name (CKN).

```
[edit security macsec]
user@host# set connectivity-association ca1 pre-shared-key ckn
0123456789abcdefABCDEF0123456789
```

The CKN must be an even-length string up to 64 hexadecimal characters (0-9, a-f, A-F).

3. Create the pre-shared key by configuring the connectivity association key (CAK).

```
[edit security macsec]
user@host# set connectivity-association ca1 pre-shared-key cak
0123456789abcdefABCDEF0123456789
```

The CAK must contain 64 hexadecimal characters (0-9, a-f, A-F).

4. Specify chassis cluster control ports for the connectivity association.

```
[edit security macsec]
user@host# set cluster-control-port 0 connectivity-association ca1
```

## Configure Static CAK on the Chassis Cluster Fabric Port

To establish a connectivity association over a chassis cluster fabric link on two SRX345 devices:

1. Configure the MACsec security mode as static-cak for the connectivity association.

```
[edit security macsec]
user@host# set connectivity-association ca2 security-mode static-cak
```

2. Create the preshared key by configuring the connectivity association key name (CKN).

```
[edit security macsec]
user@host# set connectivity-association ca2 pre-shared-key ckn
0123456789abcdefABCDEF0123456789
```

The CKN must be an even-length string up to 64 hexadecimal characters (0-9, a-f, A-F).

3. Create the preshared key by configuring the connectivity association key (CAK).

```
[edit security macsec]
user@host# set connectivity-association ca2 pre-shared-key cak
0123456789abcdefABCDEFabcdefabcdef
```

The CAK must contain 64 hexadecimal characters (0-9, a-f, A-F).

4. Specify a chassis cluster fabric ports to a connectivity association.

```
[edit security macsec]
user@host# set cluster-data-port ge-0/0/2 connectivity-association ca2
user@host# set cluster-data-port ge-5/0/2 connectivity-association ca2
```

## Configure Static CAK on the Control Port for SRX1600, SRX2300, and SRX4300 Devices

To configure a connectivity association over a chassis cluster control link on two SRX1600 devices or two SRX2300 devices or SRX4300 devices.

1. Configure the MACsec security mode as static-cak for the connectivity association.

```
[edit]
user@host# set security macsec connectivity-association ca1 security-mode static-cak
```

2. Create the preshared key by configuring the connectivity association key name (CKN).

```
[edit]
user@host# set security macsec connectivity-association ca1 pre-shared-key ckn
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

The CKN must be an even-length string up to 64 hexadecimal characters (0-9, a-f, A-F).

3. Create the preshared key by configuring the connectivity association key (CAK).

```
[edit]
user@host# set security macsec connectivity-association ca1 pre-shared-key cak
ABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEF.
```

The CAK must contain 64 hexadecimal characters (0-9, a-f, A-F).

4. Specify a chassis cluster control port for the connectivity association.

```
[edit]
user@host# set security macsec cluster-control-port 0 connectivity-association ca1
user@host# set security macsec cluster-control-port 1 connectivity-association ca1
```

To view the status of the active MACsec connections, run the **show security macsec connections** command.

```
user@host> show security macsec connections
Interface name: em0
  CA name: ca1
  Cipher suite: GCM-AES-128   Encryption: on
  Key server offset: 0       Include SCI: no
  Replay protect: off       Replay window: 0
  Outbound secure channels
    SC Id: 02:00:00:01:01:04/1
    Outgoing packet number: 1914287
    Secure associations
      AN: 0 Status: inuse Create time: 07:33:26
  Inbound secure channels
    SC Id: 02:00:00:02:01:04/1
    Secure associations
      AN: 0 Status: inuse Create time: 07:33:26
```

```

Interface name: em1
  CA name: ca1
  Cipher suite: GCM-AES-128   Encryption: on
  Key server offset: 0       Include SCI: no
  Replay protect: off        Replay window: 0
  Outbound secure channels
    SC Id: 02:00:01:01:01:04/1
    Outgoing packet number: 108885
    Secure associations
      AN: 0 Status: inuse Create time: 07:33:26
  Inbound secure channels
    SC Id: 02:00:01:02:01:04/1
    Secure associations
      AN: 0 Status: inuse Create time: 07:33:26

```

To view the MACsec key agreement session information, run the **show security mka sessions** command.

```

user@host> show security mka sessions
Interface name: em0
  Interface State: Secured - Primary
  Member identifier: 7A3FC14B77F5296124A8D22A
  CAK name: 12345678
  CAK type: primary
  Security mode: static
  MKA suspended: 0(s)
  Transmit interval: 10000(ms)
  SAK rekey interval: 0(s)
  Preceding Key: enabled
  Bounded Delay: disabled
  Outbound SCI: 02:00:00:01:01:04/1
  Message number: 2713      Key number: 1
  MKA ICV Indicator: enabled
  Key server: yes           Key server priority: 16
  Latest SAK AN: 0         Latest SAK KI: 7A3FC14B77F5296124A8D22A/1
  MKA Suspend For: disabled MKA Suspend On Request: disabled
  Previous SAK AN: 0       Previous SAK KI: 00000000000000000000/0
  Peer list
    1. Member identifier: 6A9B3CC75376160D74AAA1E7 (live)
      Message number: 2711      Hold time: 57000 (ms)
      SCI: 02:00:00:02:01:04/1  Uptime: 07:31:39
      Lowest acceptable PN: 1674733

```

```

Interface name: em1
  Interface State: Secured - Primary
  Member identifier: 989CB809BF3759C9EAC10F5A
  CAK name: 12345678
  CAK type: primary
  Security mode: static
  MKA suspended: 0(s)
  Transmit interval: 10000(ms)
  SAK rekey interval: 0(s)
  Preceding Key: enabled
  Bounded Delay: disabled
  Outbound SCI: 02:00:01:01:01:04/1
  Message number: 2713      Key number: 1
  MKA ICV Indicator: enabled
  Key server: yes          Key server priority: 16
  Latest SAK AN: 0        Latest SAK KI: 989CB809BF3759C9EAC10F5A/1
  MKA Suspend For: disabled MKA Suspend On Request: disabled
  Previous SAK AN: 0      Previous SAK KI: 0000000000000000000000/0
Peer list
  1. Member identifier: 16015BCD3844F12DFA89AB7F (live)
     Message number: 2711      Hold time: 57000 (ms)
     SCI: 02:00:01:02:01:04/1  Uptime: 07:31:39
     Lowest acceptable PN: 111017

```

To view the security status of control and fabric ports. MACsec is enabled for both control port 0 and control port 1, run the **show chassis cluster interfaces** command.

```

user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      em0        Up                 Disabled     Enabled
  1      em1        Up                 Disabled     Enabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status                Security
          (Physical/Monitored)
  fab0    et-0/3/0        Up / Up               Disabled

```



```

fab0    et-0/3/1      Up    / Up      Disabled
fab1    et-7/3/0      Up    / Up      Disabled
fab1    et-7/3/1      Up    / Up      Disabled

```

Redundant-pseudo-interface Information:

```

Name      Status      Redundancy-group
lo0       Up          0

```

## Configure Static CAK on the Control Port for SRX4600

Use this procedure to establish a CA over a chassis cluster control link on two SRX4600 devices.

1. Configure the MACsec security mode as static-cak for the connectivity association:

```

[edit]
user@host# set security macsec connectivity-association ca1 security-mode static-cak

```

2. Create the preshared key by configuring the connectivity association key name (CKN).

```

[edit]
user@host# set security macsec connectivity-association ca1 pre-shared-key ckn
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

```

The CKN must be an even-length string up to 64 hexadecimal characters (0-9, a-f, A-F).

3. Create the preshared key by configuring the connectivity association key (CAK).

```

[edit]
user@host# set security macsec connectivity-association ca1 pre-shared-key cak
ABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEF.

```

The CAK must contain 64 hexadecimal characters (0-9, a-f, A-F).

4. Specify a chassis cluster control port for the connectivity association.

```

[edit]
user@host# set security macsec cluster-control-port 0 connectivity-association ca1
user@host# set security macsec cluster-control-port 1 connectivity-association ca1

```

## Verify MACSEC Configuration

### IN THIS SECTION

- [Display the Status of Active MACsec Connections on the Device | 500](#)
- [Display MACsec Key Agreement \(MKA\) Session Information | 501](#)
- [Verify the MACsec-Secured Traffic Is Traversing Through the Interface | 502](#)
- [Verify Chassis Cluster Ports Are Secured with MACsec Configuration | 504](#)

To confirm that the configuration provided in "[Configure Static CAK on the Control Port for SRX4600](#)" on page 499 is working properly, perform these tasks:

### Display the Status of Active MACsec Connections on the Device

#### IN THIS SECTION

- [Purpose | 500](#)
- [Action | 500](#)
- [Meaning | 501](#)

#### *Purpose*

Verify that MACsec is operational on the chassis cluster setup.

#### *Action*

From the operational mode, enter the `show security macsec connections interface interface-name` command on one or both of the nodes of chassis cluster setup.

```
{primary:node0}[edit]
user@host# show security macsec connections
```

```
Interface name: em0
CA name: ca1
```

```

Cipher suite: GCM-AES-128   Encryption: on
Key server offset: 0       Include SCI: no
Replay protect: off       Replay window: 0
  Outbound secure channels
    SC Id: 02:00:00:01:01:04/1
    Outgoing packet number: 1
    Secure associations
      AN: 3 Status: inuse Create time: 00:01:43
  Inbound secure channels
    SC Id: 02:00:00:02:01:04/1
    Secure associations
      AN: 3 Status: inuse Create time: 00:01:43

```

### *Meaning*

The `Interface name` and `CA name` outputs show that the MACsec connectivity association is operational on the interface `em0`. The output does not appear when the connectivity association is not operational on the interface.

### Display MACsec Key Agreement (MKA) Session Information

#### IN THIS SECTION

- Purpose | 501
- Action | 502
- Meaning | 502

### *Purpose*

Display MACsec Key Agreement (MKA) session information for all interfaces.

**Action**

From the operational mode, enter the `show security mka sessions` command.

```

user@host> show security mka sessions

Interface name: em0
  Member identifier: B51CXXX2678A7F5F6C12345
  CAK name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  Transmit interval: 10000(ms)
  Outbound SCI: 02:00:00:01:01:04/1
  Message number: 270      Key number: 8
  Key server: yes        Key server priority: 16
  Latest SAK AN: 3        Latest SAK KI: B51C8XXX2678A7A5B6C54321/8
  Previous SAK AN: 0      Previous SAK KI: 00000000000000000000/0
  Peer list
  1. Member identifier: 0413427B38817XXXXF054321 (live)
     Message number: 8 Hold time: 59000 (ms)
     SCI: 02:00:00:02:01:04/1
     Lowest acceptable PN: 0

```

**Meaning**

The outputs show the status of MKA sessions.

**Verify the MACsec-Secured Traffic Is Traversing Through the Interface****IN THIS SECTION**

- Purpose | 502
- Action | 503
- Meaning | 503

**Purpose**

Verify that traffic traversing through the interface is MACsec-secured.

### ***Action***

From the operational mode, enter the `show security macsec statistics interface em0 detail` command.

```
user@host> show security macsec statistics interface em0 detail
```

```
Interface name: em0
Secure Channel transmitted
  Encrypted packets: 2397305
  Encrypted bytes: 129922480
  Protected packets: 0
  Protected bytes: 0
Secure Association transmitted
  Encrypted packets: 2397305
  Protected packets: 0
Secure Channel received
  Accepted packets: 2395850
  Validated bytes: 0
  Decrypted bytes: 131715088
Secure Association received
  Accepted packets: 2395850
  Validated bytes: 0
  Decrypted bytes: 0
```

### ***Meaning***

The `Encrypted packets` line under the `Secure Channel transmitted` field are the values incremented each time a packet is sent from the interface that is secured and encrypted by MACsec.

The `Accepted packets` line under the `Secure Association received` field are the values incremented each time a packet that has passed the MACsec integrity check is received on the interface. The `Decrypted bytes` line under the `Secure Association received` output is incremented each time an encrypted packet is received and decrypted.

## Verify Chassis Cluster Ports Are Secured with MACsec Configuration

### IN THIS SECTION

- Purpose | 504
- Action | 504
- Meaning | 505

### *Purpose*

Verify that MACsec is configured on chassis cluster ports.

### *Action*

From operational mode, enter the `show chassis cluster interfaces` command.

```
user@host> show chassis cluster interfaces

Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      em0        Up                 Disabled     Enabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status                               Security
          (Physical/Monitored)
  fab0    xe-1/1/6         Up / Up                             Enabled
  fab0
  fab1    xe-8/1/6         Up / Up                             Enabled
  fab1

Redundant-ethernet Information:
  Name      Status  Redundancy-group
  reth0     Up      1
```

```

reth1      Up      2
reth2      Down    Not configured
reth3      Down    Not configured
reth4      Down    Not configured
reth5      Down    Not configured
reth6      Down    Not configured
reth7      Down    Not configured

```

Redundant-pseudo-interface Information:

```

Name      Status  Redundancy-group
lo0       Up      0

```

### Meaning

The Security line under the Control interfaces output for em0 interface shown as Secured means that the traffic sent from the em0 interface is secured and encrypted by MACsec.

You can also use the `show chassis cluster status` command to display the current status of the chassis cluster.

## RELATED DOCUMENTATION

[Understanding Media Access Control Security \(MACsec\) | 483](#)

`macsec`

### 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
17.4R1	Starting in Junos OS Release 17.4R1, MACsec is supported on HA control and fabric ports of SRX4600 devices in chassis cluster mode.
15.1X49-D60	Starting in Junos OS Release 15.1X49-D60, Media Access Control Security(MACsec) is supported on control and fabric ports of SRX340 and SRX345 devices in chassis cluster mode.

## RELATED DOCUMENTATION

[SRX Series Chassis Cluster Configuration Overview | 14](#)

## Understanding Sctp Behavior in Chassis Cluster

In a *chassis cluster* configuration mode, the Sctp configuration and the established Sctp association is synced with the peer device. The Sctp module supports both active-active and active-passive modes.

The established Sctp association sends a creation or deletion message to the peer whenever an association is created or deleted on the active device. The secondary device adds or deletes an association respectively upon receiving the message from the established Sctp association. Sctp module then registers the corresponding callback function to receive and handle this message. There is no continuous timer sync between the two associations.

Sctp module will register a cold start sync function when a secondary device joins the cluster or reboots. The Sctp cold start function is called to sync all Sctp associations with the peer devices at the same time.

After the switchover, the established Sctp associations will remain functioning, but the associations in the progress of establishment will be lost and the establishment procedure needs to be re-initiated. It is also possible that the associations in the progress of teardown miss the ack message and leaves unestablished Sctp associations in the firewall. These associations will be cleaned up when the timer expires (5 hours by default) due to no activity in the association.

- You should configure all policies for your required Sctp sessions.  
For example, suppose you have endpoints A and B. Endpoint A has one Sctp association with x number of IPs (IP\_a1, IP\_a2, IP\_a3...IP\_ax). Endpoint B has one Sctp association with y number of IPs (IP\_b1, IP\_b2, IP\_b3...IP\_by.) The policy on the security device should permit all possible x\*y paths in both directions.
- When an Sctp association is removed, the related Sctp sessions still exist and time out by themselves.



## Example: Encrypting Messages Between Two Nodes in a Chassis Cluster

This example provides you a procedure to enable encryption on security devices.

This procedure provides you step on how you can optionally configure the control-link to encrypt messages between two nodes in a chassis cluster. This configuration will ensure secure login by using configured internal IPsec security association (SA).

When the internal IPsec SA is configured, IPsec-based rlogin and remote command (rcmd) are enforced so that attackers cannot gain privileged access or observe traffic containing administrator commands and outputs.

You do not need to configure the internal IPsec SA on both nodes because the nodes are synchronized when the configuration is committed.

1. To enable control link encryption in chassis cluster, run the following commands:

The only supported encryption algorithm is 3des-cbc and the key must be exactly 24 bytes long, otherwise the configuration will result in commit failure.

```
edit security ipsec internal security-association
root@srx-8# show | display set
set security ipsec internal security-association manual encryption algorithm 3des-cbc
set security ipsec internal security-association manual encryption ike-ha-link-encryption
enable
set security ipsec internal security-association manual encryption key ascii-text "$9$8gPx-
b4aU.PQs2PQFnpu8X7dsgGUHPT3.Pu1EhvMwYgJjq3n9CpBFnt0REeKZGDj.fu01hcr"
```

2. Commit the configuration.

```
{primary:node0} [edit] root@srx-8# commit
warning: changes needs reboot to take effect
```

```
warning: changes needs reboot to take effect
node0: commit complete
node1: commit complete
```

After the settings have been configured correctly and committed, a reboot would be required for the feature to take effect.

3. View the configuration of control link encryption before reboot and after reboot.

**Before reboot, the status of this feature is disabled.**

```
show security internal-security-association
```

```
node0:
-----
Internal SA Status : Disabled
HA link encryption for IKE internal message status: Disabled

node1:
-----
Internal SA Status : Disabled
HA link encryption for IKE internal message status: Disabled
```

**After reboot, to ensure that the encryption is active**

```
show security internal-security-association
```

```
{primary:node0}
root@srx-8> show security internal-security-association

node0:
-----
Internal SA Status : Enabled
HA link encryption for IKE internal message status: Enabled

node1:
-----
Internal SA Status : Enabled
HA link encryption for IKE internal message status: Enabled
```

## RELATED DOCUMENTATION

| [internal \(Security IPsec\)](#)

# 5

CHAPTER

## Upgrading or Disabling a Chassis Cluster

---

[Upgrading Individual Devices in a Chassis Cluster Separately | 510](#)

[Upgrading Devices in a Chassis Cluster Using ICU | 510](#)

[Upgrading a Chassis Cluster Using In-Service Software Upgrade | 516](#)

[Disabling a Chassis Cluster | 538](#)

---

# Upgrading Individual Devices in a Chassis Cluster Separately

Devices in a chassis cluster can be upgraded separately one at a time; some models allow one device after the other to be upgraded using failover and an in-service software upgrade (ISSU) to reduce the operational impact of the upgrade.

To upgrade each device in a chassis cluster separately:

During this type of chassis cluster upgrade, a service disruption of about 3 to 5 minutes occurs.

1. Load the new image file on node 0.
2. Perform the image upgrade without rebooting the node by entering:

```
user@host> request system software add image_name
```

3. Load the new image file on node 1.
4. Repeat Step 2.
5. Reboot both nodes simultaneously.

## RELATED DOCUMENTATION

[Upgrading Devices in a Chassis Cluster Using ICU | 510](#)

[Upgrading a Chassis Cluster Using In-Service Software Upgrade | 516](#)

# Upgrading Devices in a Chassis Cluster Using ICU

## IN THIS SECTION

- [Upgrading Both Devices in a Chassis Cluster Using ICU | 511](#)
- [Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster | 512](#)
- [Upgrading ICU Using a Build Available on an FTP Server | 513](#)
- [Terminating an Upgrade in a Chassis Cluster During an ICU | 514](#)

The chassis cluster ICU feature allows both devices in a cluster to be upgraded from supported Junos OS versions using a single command. For more information, see the following topics:

## Upgrading Both Devices in a Chassis Cluster Using ICU

Before you begin, note the following:

- ICU is available with the no-sync option only for SRX300, SRX320, SRX340, SRX345, and SRX380 devices.
- Before starting ICU, you should ensure that sufficient disk space is available. See ["Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster" on page 512](#) and ["Upgrading ICU Using a Build Available on an FTP Server" on page 513](#).
- For SRX300, SRX320, SRX340, SRX345 and SRX380 devices, this feature cannot be used to downgrade to a build earlier than Junos OS 11.2 R2.

For SRX1500 devices, this feature cannot be used to downgrade to a build earlier than Junos OS 15.1X49-D50.

SRX Series Firewalls in a chassis cluster can be upgraded with a minimal service disruption using In-Band Cluster Upgrade (ICU). The chassis cluster ICU feature allows both devices in a cluster to be upgraded from supported Junos OS versions using a single command. You can enable this feature by executing the request `system software in-service-upgrade image_name` command on the primary node. This command upgrades the Junos OS and reboots both the secondary node and the primary node in turn. During the ICU process, traffic outage is minimal; however, cold synchronization is not provided between the two nodes.

For SRX300, SRX320, SRX340, SRX345, and SRX380 devices, the devices in a chassis cluster can be upgraded with a minimal service disruption of approximately 30 seconds using ICU with the no-sync option. The chassis cluster ICU feature allows both devices in a cluster to be upgraded from supported Junos OS versions.

You must use the in-band cluster upgrade (ICU) commands on SRX1500 device to upgrade following Junos OS Releases:

- Junos OS Release 15.1X49-D50 to Junos OS Release 15.1X49-D100
- Junos OS Release 15.1X49-D60 to Junos OS Release 15.1X49-D110
- Junos OS Release 15.1X49-D50 to Junos OS Release 15.1X49-D120

You must use the in-band cluster upgrade (ICU) commands on SRX1600 device to upgrade from Junos OS Release 23.3R1 to later release.

You can use the in-band cluster upgrade (ICU) commands on SRX2300,SRX4100 and SRX4200 devices to upgrade following Junos OS Releases:

- Junos OS Release 15.1X49-D65 to Junos OS Release 15.1X49-D70
- Junos OS Release 15.1X49-D70 to Junos OS Release 15.1X49-D80.

For SRX300, SRX320, SRX340, SRX345 and SRX380 devices, the impact on traffic is as follows:

- Drop in traffic (30 seconds approximately)
- Loss of security flow sessions

The upgrade is initiated with the Junos OS build locally available on the primary node of the device or on an FTP server.

- The primary node, RG0, changes to the secondary node after an ICU upgrade.
- During ICU, the chassis cluster redundancy groups are failed over to the primary node to change the cluster to active/passive mode.
- ICU states can be checked from the syslog or with the console/console logs.
- ICU requires that both nodes be running a dual-root partitioning scheme with one exception being the SRX1500 and SRX1600. ICU will not continue if it fails to detect dual-root partitioning on either of the nodes. Requirement of the dual-root partitioning is applicable only for SRX300, SRX320, SRX340, SRX345, and SRX380 devices.

Dual-root partitioning is not supported on SRX1500 and SRX1600 devices. SRX1500 and SRX1600 use solid-state drive (SSD) as secondary storage.

## Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster

Ensure that sufficient disk space is available for the Junos OS package in the `/var/tmp` location in the secondary node of the cluster.

To upgrade ICU using a build locally available on the primary node of a cluster:

1. Copy the Junos OS package build to the primary node at any location, or mount a network file server folder containing the Junos OS build.
2. Start ICU by entering the following command:

```
user@host> request system software in-service-upgrade image_name no-sync (for SRX300, SRX320, SRX340, SRX345, and SRX380 devices)
```

```
user@host> request system software in-service-upgrade image_name (for SRX1500 devices prior to Junos OS Release 15.1X49-D70)
```

```
user@host> request vmhost software in-service-upgrade image_name (for SRX1600 and SRX2300 devices)
```

## Upgrading ICU Using a Build Available on an FTP Server

Ensure that sufficient disk space is available for the Junos OS package in the `/var/tmp` location in both the primary and the secondary nodes of the cluster.

To upgrade ICU using a build available on an FTP server:

1. Place the Junos OS build on an FTP server.
2. (SRX300, SRX320, SRX340, SRX345 and SRX380 only) Start ICU by entering the following command:

```
user@root> request system software in-service-upgrade <ftp url for junos image> no-sync
```

### Sample Command

```
user@root> request system software in-service-upgrade ftp://<user>:<password>@<server>:/<path> no-sync
```

This command upgrades the Junos OS and reboots both nodes in turn.

3. (SRX1500 only prior to Junos OS Release 15.1X49-D70) Start ICU by entering the following command:

```
user@root> request system software in-service-upgrade <ftp url for junos image>
```

### Sample Command

```
user@root> request system software in-service-upgrade ftp://<user>:<password>@<server>:/<path>
```

This command upgrades the Junos OS and reboots both nodes in turn.

For SRX1600 and SRX2300 devices, start ICU by entering the following command:

```
user@root> request vmhost software in-service-upgrade ftp://<user>:<password>@<server>:/<path>
```

The upgrade process displays the following warning message to reboot the system:

**WARNING:** A reboot is required to load this software correctly. Use the `request system reboot` command when software installation is complete.

This warning message can be ignored because the ICU process automatically reboots both the nodes.

## Terminating an Upgrade in a Chassis Cluster During an ICU

You can terminate an ICU at any time by issuing the following command on the primary node:

```
request system software abort in-service-upgrade
```

Issuing an `abort` command during or after the secondary node reboots puts the cluster in an inconsistent state. The secondary node boots up running the new Junos OS build, while the primary continues to run the older Junos OS build.

To recover from the chassis cluster inconsistent state, perform the following actions sequentially on the secondary node:

1. Issue an `abort` command:

```
request system software abort in-service-upgrade
```

2. Roll back the Junos OS build by entering the following command:

```
request system software rollback node < node-id >
```

3. Reboot the secondary node immediately by using the following command:

```
request system reboot
```

You must execute the above steps sequentially to complete the recovery process and avoid cluster instability.

[Table 36 on page 514](#) lists the options and their descriptions for the `request system software in-service-upgrade` command.

**Table 36: request system software in-service-upgrade Output Fields**

Options	Description
no-sync	Disables the flow state from syncing up when the old secondary node has booted with a new Junos OS image.  This option is not available on SRX1500 and SRX1600 devices.



**Table 36: request system software in-service-upgrade Output Fields (Continued)**

Options	Description
no-tcp-syn-check	Creates a window wherein the TCP SYN check for the incoming packets will be disabled. The default value for the window is 7200 seconds (2 hours).  This option is not available on SRX1500 and SRX1600 devices.
no-validate	Disables the validation of the configuration at the time of the installation. The system behavior is similar to software add.
unlink	Removes the package from the local media after installation.

- During ICU, if a termination command is executed, ICU will terminate only after the current operation finishes. This is required to avoid any inconsistency with the devices.  
  
For example, if formatting and upgrade of a node is in progress, ICU terminates after this operation finishes.
- After a termination, ICU will try to roll back the build on the nodes if the upgrading nodes step was completed.

## RELATED DOCUMENTATION

[Upgrading Individual Devices in a Chassis Cluster Separately | 510](#)

[Upgrading a Chassis Cluster Using In-Service Software Upgrade | 516](#)

[Disabling a Chassis Cluster | 538](#)

# Upgrading a Chassis Cluster Using In-Service Software Upgrade

## IN THIS SECTION

- [Understanding ISSU for a Chassis Cluster | 516](#)
- [ISSU System Requirements | 519](#)
- [Upgrading Both Devices in a Chassis Cluster Using ISSU | 521](#)
- [Rolling Back Devices in a Chassis Cluster After an ISSU | 523](#)
- [Enabling an Automatic Chassis Cluster Node Failback After an ISSU | 523](#)
- [Log Error Messages used for Troubleshooting ISSU-Related Problems | 524](#)
- [Managing Chassis Cluster ISSU-Related Problems | 533](#)

In-service software upgrade (ISSU) enables a software upgrade from one Junos OS version to a later Junos OS version with minimal downtime. For more information, see the following topics:

## Understanding ISSU for a Chassis Cluster

In-service software upgrade (ISSU) enables a software upgrade from one Junos OS version to a later Junos OS version with little or no downtime. ISSU is performed when the devices are operating in chassis cluster mode only.

The *chassis cluster* ISSU feature enables both devices in a cluster to be upgraded from supported Junos OS versions with a minimal disruption in traffic and without a disruption in service.

Starting with Junos OS Release 15.1X49-D80, SRX4100 and SRX4200 devices support ISSU.

Starting with Junos OS Release 15.1X49-D70, SRX1500 devices support ISSU.

Starting with Junos OS Release 23.4R1, SRX1600 and SRX2300 devices support ISSU.

Starting with Junos OS Release 24.2R1, SRX4300 devices support ISSU.

- On SRX1500, SRX4100, and SRX4200 devices, ISSU is not supported for upgrading to 17.4 releases from previous Junos OS releases. ISSU is supported for upgrading from Junos OS 17.4 to successive 17.4 releases.
- On SRX5400, SRX5600 and SRX5800 devices, ISSU is not supported for upgrading to 17.3 and higher releases from earlier Junos OS releases. ISSU is supported for upgrading from Junos OS 17.3 to Junos 17.4 releases.
- SRX300 Series devices and vSRX Virtual Firewall do not support ISSU.

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

ISSU has the following limitations:

- ISSU is available only for Junos OS Release 10.4R4 or later.
- ISSU does not support software downgrades.
- If you upgrade from a Junos OS version that supports only IPv4 to a version that supports both IPv4 and IPv6, the IPv4 traffic continue to work during the upgrade process. If you upgrade from a Junos OS version that supports both IPv4 and IPv6 to a version that supports both IPv4 and IPv6, both the IPv4 and IPv6 traffic continue to work during the upgrade process. Junos OS Release 10.2 and later releases support flow-based processing for IPv6 traffic.
- During an ISSU, you cannot bring any PICs online. You cannot perform operations such as commit, restart, or halt.
- During an ISSU, operations like fabric monitoring, control link recovery, and RGX preempt are suspended.
- During an ISSU, you cannot commit any configurations.

For details about ISSU support status, see knowledge base article [KB17946](#).

The following process occurs during an ISSU for devices in a chassis cluster. The sequences given below are applicable when RG-0 is node 0 (primary node). Note that you must initiate an ISSU from RG-0 primary. If you initiate the upgrade on node 1 (RG-0 secondary), an error message is displayed.

1. At the beginning of a chassis cluster ISSU, the system automatically fails over all RG-1+ redundancy groups that are not primary on the node from which the ISSU is initiated. This action ensures that all the redundancy groups are active on only the RG-0 primary node.

The automatic failover of all RG-1+ redundancy groups is available from Junos OS release 12.1 or later. If you are using Junos OS release 11.4 or earlier, before starting the ISSU, ensure that all the redundancy groups are all active on only the RG-0 primary node.

After the system fails over all RG-1+ redundancy groups, it sets the manual failover bit and changes all RG-1+ primary node priorities to 255, regardless of whether the redundancy group failed over to the RG-0 primary node.

2. The primary node (node 0) validates the device configuration to ensure that it can be committed using the new software version. Checks are made for disk space availability for the `/var` file system on both nodes, unsupported configurations, and unsupported Physical Interface Cards (PICs).

If the disk space available on either of the Routing Engines is insufficient, the ISSU process fails and returns an error message. However, unsupported PICs do not prevent the ISSU. The software issues a warning to indicate that these PICs will restart during the upgrade. Similarly, an unsupported protocol configuration does not prevent the ISSU. However, the software issues a warning that packet loss might occur for the protocol during the upgrade.

3. When the validation succeeds, the kernel state synchronization daemon (`ksyncd`) synchronizes the kernel on the secondary node (node 1) with the node 0.
4. Node 1 is upgraded with the new software image. Before being upgraded, the node 1 gets the configuration file from node 0 and validates the configuration to ensure that it can be committed using the new software version. After being upgraded, it is resynchronized with node 0.
5. The chassis cluster process (`chassisd`) on the node 0 prepares other software processes for the ISSU. When all the processes are ready, `chassisd` sends a message to the PICs installed in the device.
6. The Packet Forwarding Engine on each Flexible PIC Concentrator (FPC) saves its state and downloads the new software image from node 1. Next, each Packet Forwarding Engine sends a message (`unified-ISSU ready`) to the `chassisd`.
7. After receiving the message (`unified-ISSU ready`) from a Packet Forwarding Engine, the `chassisd` sends a 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 node 1 running the new software. The `chassisd` is also reestablished with node 0.
8. After all Packet Forwarding Engines have sent a `ready` message using the `chassisd` on node 0, other software processes are prepared for a node switchover. The system is ready for a switchover at this point.
9. Node switchover occurs and node 1 becomes the new primary node (hitherto secondary node 1).
10. The new secondary node (hitherto primary node 0) is now upgraded to the new software image.

When both nodes are successfully upgraded, the ISSU is complete.

When upgrading a version cluster that does not support encryption to a version that supports encryption, upgrade the first node to the new version. Without the encryption configured and enabled, two nodes with different versions can still communicate with each other and service is not broken. After upgrading the first node, upgrade the second node to the new version. Users can decide whether to turn on the encryption feature after completing the upgrade. Encryption must be deactivated before downgrading to a version that does not support encryption. This ensures that communication between an encryption-enabled version node and a downgraded node does not break, because both are no longer encrypted.

**NOTE:** The policies in the Routing Engine and Packet Forwarding Engine must be in sync for the configuration to be committed. When the policy configurations are modified and the policies are out of sync, the system displays an error message.

As a workaround, you must use the *request security policies resync* command to synchronize the configuration of security policies in the Routing Engine and Packet Forwarding Engine, in case if you notice security policies are out of sync after an upgrade.

## ISSU System Requirements

You can use ISSU to upgrade from an ISSU-capable software release to a later release.

To perform an ISSU, your device must be running a Junos OS release that supports ISSU for the specific platform. See [Table 37 on page 519](#) for platform support.

**Table 37: ISSU Platform Support**

Device	Junos OS Release
SRX5800	10.4R4 or later
SRX5600	10.4R4 or later
SRX5400	12.1X46-D20 or later
SRX1500	15.1X49-D70 or later

**Table 37: ISSU Platform Support (Continued)**

Device	Junos OS Release
SRX1600	23.4R1 or later
SRX2300	23.4R1 or later
SRX4100	15.1X49-D80 or later
SRX4200	15.1X49-D80 or later
SRX4300	24.2R1 or later
SRX4600	17.4R1 or later

For additional details on ISSU support and limitations, see [ISSU/ICU Upgrade Limitations on SRX Series Devices](#).

Note the following limitations related to an ISSU:

- The ISSU process is terminated if the Junos OS version specified for installation is a version earlier than the one currently running on the device.
- The ISSU process is terminated if the specified upgrade conflicts with the current configuration, the components supported, and so forth.
- ISSU does not support the extension application packages developed using the Junos OS SDK.
- ISSU does not support version downgrading on all supported SRX Series Firewalls.
- ISSU occasionally fails under heavy CPU load.

To downgrade from an ISSU-capable release to an earlier release (ISSU-capable or not), use the `request system software add` command. Unlike an upgrade using the ISSU process, a downgrade using the `request system software add` command might cause network disruptions and loss of data.

We strongly recommend that you perform ISSU under the following conditions:

- When both the primary and secondary nodes are healthy
- During system maintenance period
- During the lowest possible traffic period

- When the Routing Engine CPU usage is less than 40 percent

In cases where ISSU is not supported or recommended, while still downtime during the system upgrade must be minimized, the minimal downtime procedure can be used, see knowledge base article [KB17947](#).

## Upgrading Both Devices in a Chassis Cluster Using ISSU

Before you begin the ISSU for upgrading both the devices, note the following guidelines:

- Ensure the following ISSU pre-check requirements are met:
  - All redundancy groups priority is greater than 0
  - All redundancy groups are either primary or secondary in state
  - There exists enough (double the image size) space available in the `/var/tmp`
  - Usage of CPU is under 80% within 5 seconds period

If the pre-check requirements are not met, ISSU will terminate at the beginning.

- Back up the software using the `request system snapshot` command on each Routing Engine to back up the system software to the device's hard disk. The `request system snapshot` command is not supported on SRX1500, SRX1600, SRX2300, SRX4100, SRX4200, SRX4300, and SRX4600 platforms.
- If you are using Junos OS Release 11.4 or earlier, before starting the ISSU, set the failover for all redundancy groups so that they are all active on only one node (primary). See "[Initiating a Chassis Cluster Manual Redundancy Group Failover](#)" on page 261.

If you are using Junos OS Release 12.1 or later, Junos OS automatically fails over all RGs to the RGO primary.

- We recommend that you enable graceful restart for routing protocols before you start an ISSU.

On all supported SRX Series Firewalls, the first recommended ISSU *from* release is Junos OS Release 10.4R4.

The chassis cluster ISSU feature enables both devices in a cluster to be upgraded from supported Junos OS versions with a traffic impact similar to that of redundancy group failovers.

Starting with Junos OS Release 15.1X49-D70, SRX1500 devices support ISSU.

Starting with Junos OS Release 15.1X49-D80, SRX4100 and SRX4200 devices support ISSU.

Starting with Junos OS Release 17.4R1, SRX4600 devices support ISSU.

To perform an ISSU from the CLI on Routing Engine2:

1. Download the software package from the Juniper Networks Support website: <https://www.juniper.net/support/downloads/>
2. Copy the package on primary node of the cluster. We recommend that you copy the package to the `/var/tmp` directory, which is a large file system on the hard disk. Note that the node from where you initiate the ISSU must have the software image.  

```
user@host>file copy ftp://username:prompt@ftp.hostname.net/filename /var/tmp/filename
```
3. Verify the current software version running on both nodes by issuing the `show version` command on the primary node.
4. Start the ISSU from the node that is primary for all the redundancy groups by entering the following command:

```
user@host> request system software in-service-upgrade image-name-with-full-path
```

For SRX1500, SRX4100, and SRX4200 devices, you can optionally remove the original image file by including `unlink` in the command.

```
user@host> request system software in-service-upgrade image-name-with-full-path unlink
```

Wait for both nodes to complete the upgrade (After which you are logged out of the device).

5. Wait a few minutes, and then log in to the device again. Verify by using the `show version` command that both devices in the cluster are running the new Junos OS release.
6. Verify that all policies, zones, redundancy groups, and other real-time objects (RTOs) return to their correct states.
7. Make node 0 the primary node again by issuing the `request chassis cluster failover node node-number redundancy-group group-number` command.

If you want redundancy groups to automatically return to node 0 as the primary after an in-service software upgrade (ISSU), you must set the redundancy group priority such that node 0 is primary and enable the `preempt` option. Note that this method works for all redundancy groups except redundancy group 0. You must manually set the failover for redundancy group 0.

To set the redundancy group priority and enable the `preempt` option, see "[Example: Configuring Chassis Cluster Redundancy Groups](#)" on page 87.

To manually set the failover for a redundancy group, see "[Initiating a Chassis Cluster Manual Redundancy Group Failover](#)" on page 261.

During the upgrade, both devices might experience redundancy group failovers, but traffic is not disrupted. Each device validates the package and checks version compatibility before beginning the upgrade. If the system finds that the new package version is not compatible with the currently installed version, the device refuses the upgrade or prompts you to take corrective action. Sometimes a single



feature is not compatible, in which case, the upgrade software prompts you to either terminate the upgrade or turn off the feature before beginning the upgrade.

If you want to operate the SRX Series Firewall back as a standalone device or to remove a node from a chassis cluster, ensure that you have terminated the ISSU procedure on both the nodes (in case ISSU procedure is initiated)

To start ISSU process on SRX5K devices with Routing Engine3 and on SRX1600, SRX2300, and SRX4300 devices:

1. Run the following command to start ISSU:

```
user@host> request vmhost software in-service-upgrade image-name-with-full-path
```

## SEE ALSO

[In-Service Hardware Upgrade for SRX5K-RE-1800X4 and SRX5K-SCBE in a Chassis Cluster](#)

## Rolling Back Devices in a Chassis Cluster After an ISSU

If an ISSU fails to complete and only one device in the cluster is upgraded, you can roll back to the previous configuration on the upgraded device alone by issuing one of the following commands on the upgraded device:

- `request chassis cluster in-service-upgrade abort`
- `request system software rollback node node-id reboot`
- `request system reboot`

## Enabling an Automatic Chassis Cluster Node Failback After an ISSU

If you want redundancy groups to automatically return to node 0 as the primary after the an in-service software upgrade (ISSU), you must set the redundancy group priority such that node 0 is primary and enable the preempt option. Note that this method works for all redundancy groups except redundancy group 0. You must manually set the failover for a redundancy group 0. To set the redundancy group priority and enable the preempt option, see "[Example: Configuring Chassis Cluster Redundancy Groups](#)"

on page 87. To manually set the failover for a redundancy group, see "[Initiating a Chassis Cluster Manual Redundancy Group Failover](#)" on page 261.

To upgrade node 0 and make it available in the chassis cluster, manually reboot node 0. Node 0 does not reboot automatically.

## Log Error Messages used for Troubleshooting ISSU-Related Problems

### IN THIS SECTION

- [Chassisd Process Errors | 524](#)
- [Understanding Common Error Handling for ISSU | 525](#)
- [ISSU Support-Related Errors | 529](#)
- [Initial Validation Checks Failure | 529](#)
- [Installation-Related Errors | 531](#)
- [Redundancy Group Failover Errors | 532](#)
- [Kernel State Synchronization Errors | 533](#)

The following problems might occur during an ISSU upgrade. You can identify the errors by using the details in the logs. For detailed information about specific system log messages, see [System Log Explorer](#).

### Chassisd Process Errors

#### IN THIS SECTION

- [Problem | 525](#)
- [Solution | 525](#)

## Problem

### Description

Errors related to chassisd.

### Solution

Use the error messages to understand the issues related to chassisd.

When ISSU starts, a request is sent to chassisd to check whether there are any problems related to the ISSU from a chassis perspective. If there is a problem, a log message is created.

## Understanding Common Error Handling for ISSU

### IN THIS SECTION

- [Problem | 525](#)
- [Solution | 525](#)

## Problem

### Description

You might encounter some problems in the course of an ISSU. This section provides details on how to handle them.

### Solution

Any errors encountered during an ISSU result in the creation of log messages, and ISSU continues to function without impact to traffic. If reverting to previous versions is required, the event is either logged or the ISSU is halted, so as not to create any mismatched versions on both nodes of the chassis cluster. [Table 38 on page 526](#) provides some of the common error conditions and the workarounds for them. The sample messages used in the [Table 38 on page 526](#) are from the SRX1500 device and are also applicable to all supported SRX Series Firewalls.

**Table 38: ISSU-Related Errors and Solutions**

Error Conditions	Solutions
<p>Attempt to initiate an ISSU when previous instance of an ISSU is already in progress</p>	<p>The following message is displayed:</p> <pre>warning: ISSU in progress</pre> <p>You can abort the current ISSU process, and initiate the ISSU again using the request chassis cluster in-service-upgrade abort command.</p>
<p>Reboot failure on the secondary node</p>	<p>No service downtime occurs, because the primary node continues to provide required services. Detailed console messages are displayed requesting that you manually clear existing ISSU states and restore the chassis cluster.</p> <pre>error: [Oct 6 12:30:16]: Reboot secondary node failed (error-code: 4.1)</pre> <pre>error: [Oct 6 12:30:16]: ISSU Aborted! Backup node maybe in inconsistent state, Please restore backup node</pre> <pre>[Oct 6 12:30:16]: ISSU aborted. But, both nodes are in ISSU window. Please do the following:</pre> <ol style="list-style-type: none"> <li>1. Rollback the node with the newer image using rollback command Note: use the 'node' option in the rollback command otherwise, images on both nodes will be rolled back</li> <li>2. Make sure that both nodes (will) have the same image</li> <li>3. Ensure the node with older image is primary for all RGs</li> <li>4. Abort ISSU on both nodes</li> <li>5. Reboot the rolled back node</li> </ol> <p>Starting with Junos OS Release 17.4R1, the hold timer for the initial reboot of the secondary node during the ISSU process is extended from 15 minutes (900 seconds) to 45 minutes (2700 seconds) in chassis clusters on SRX1500, SRX4100, SRX4200, and SRX4600 devices.</p>

Table 38: ISSU-Related Errors and Solutions (Continued)

Error Conditions	Solutions
<p>Secondary node failed to complete the cold synchronization</p>	<p>The primary node times out if the secondary node fails to complete the cold synchronization. Detailed console messages are displayed that you manually clear existing ISSU states and restore the chassis cluster. No service downtime occurs in this scenario.</p> <pre data-bbox="493 562 1398 632">[Oct 3 14:00:46]: timeout waiting for secondary node node1 to sync(error-code: 6.1) Chassis control process started, pid 36707</pre> <p>error: [Oct 3 14:00:46]: ISSU Aborted! Backup node has been upgraded, Please restore backup node</p> <pre data-bbox="566 741 1295 768">[Oct 3 14:00:46]: ISSU aborted. But, both nodes are in ISSU window.</pre> <p>Please do the following:</p> <ol data-bbox="557 814 1247 1052" style="list-style-type: none"> <li>1. Rollback the node with the newer image using rollback command Note: use the 'node' option in the rollback command otherwise, images on both nodes will be rolled back</li> <li>2. Make sure that both nodes (will) have the same image</li> <li>3. Ensure the node with older image is primary for all RGs</li> <li>4. Abort ISSU on both nodes</li> <li>5. Reboot the rolled back node</li> </ol>

Table 38: ISSU-Related Errors and Solutions (Continued)

Error Conditions	Solutions
Failover of newly upgraded secondary failed	<p>No service downtime occurs, because the primary node continues to provide required services. Detailed console messages are displayed requesting that you manually clear existing ISSU states and restore the chassis cluster.</p> <pre>[Aug 27 15:28:17]: Secondary node0 ready for failover. [Aug 27 15:28:17]: Failing over all redundancy-groups to node0 ISSU: Preparing for Switchover error: remote rg1 priority zero, abort failover. [Aug 27 15:28:17]: failover all RGs to node node0 failed (error-code: 7.1) error: [Aug 27 15:28:17]: ISSU Aborted! [Aug 27 15:28:17]: ISSU aborted. But, both nodes are in ISSU window. Please do the following: 1. Rollback the node with the newer image using rollback command    Note: use the 'node' option in the rollback command        otherwise, images on both nodes will be rolled back 2. Make sure that both nodes (will) have the same image 3. Ensure the node with older image is primary for all RGs 4. Abort ISSU on both nodes 5. Reboot the rolled back node {primary:node1}</pre>
Upgrade failure on primary	<p>No service downtime occurs, because the secondary node fails over as primary and continues to provide required services.</p>
Reboot failure on primary node	<p>Before the reboot of the primary node, devices being out of the ISSU setup, no ISSU-related error messages are displayed. The following reboot error message is displayed if any other failure is detected:</p> <pre>Reboot failure on Primary node Before the reboot of primary node, devices will be out of ISSU setup and no primary node error messages will be displayed. Primary node</pre>

## ISSU Support-Related Errors

### IN THIS SECTION

- [Problem | 529](#)
- [Solution | 529](#)

### Problem

### Description

Installation failure occurs because of unsupported software and unsupported feature configuration.

### Solution

Use the following error messages to understand the compatibility-related problems:

```
WARNING: Current configuration not compatible with /var/tmp/junos-srx5000-11.4X3.2-domestic.tgz
Exiting in-service-upgrade window
Exiting in-service-upgrade window
```

## Initial Validation Checks Failure

### IN THIS SECTION

- [Problem | 529](#)
- [Solution | 530](#)

### Problem

### Description

The initial validation checks fail.

## Solution

The validation checks fail if the image is not present or if the image file is corrupt. The following error messages are displayed when initial validation checks fail when the image is not present and the ISSU is aborted:

### When Image Is Not Present

```

user@host> ...0120914_srx_12q1_major2.2-539764-domestic.tgz reboot
Chassis ISSU Started
Chassis ISSU Started
ISSU: Validating Image
Initiating in-service-upgrade
Initiating in-service-upgrade
Fetching package...
error: File does not exist: /var/tmp/junos-srx1k3k-12.1I20120914_srx_12q1_major2.2-539764-
domestic.tgz
error: Couldn't retrieve package /var/tmp/junos-srx1k3k-12.1I20120914_srx_12q1_major2.2-539764-
domestic.tgz
Exiting in-service-upgrade window
Exiting in-service-upgrade window
Chassis ISSU Aborted
Chassis ISSU Aborted
Chassis ISSU Aborted
ISSU: IDLE
ISSU aborted; exiting ISSU window.

```

### When Image File Is Corrupted

If the image file is corrupted, the following output displays:

```

user@host> ...junos-srx1k3k-11.4X9-domestic.tgz_1 reboot
Chassis ISSU Started
node1:
-----
Chassis ISSU Started
ISSU: Validating Image
Initiating in-service-upgrade

node1:
-----

```



```

Initiating in-service-upgrade
ERROR: Cannot use /var/tmp/junos-srx1k3k-11.4X9-domestic.tgz_1:
gzip: stdin: invalid compressed data--format violated
tar: Child returned status 1
tar: Error exit delayed from previous errors
ERROR: It may have been corrupted during download.
ERROR: Please try again, making sure to use a binary transfer.
Exiting in-service-upgrade window

node1:
-----

Exiting in-service-upgrade window
Chassis ISSU Aborted
Chassis ISSU Aborted

node1:
-----

Chassis ISSU Aborted
ISSU: IDLE
ISSU aborted; exiting ISSU window.

{primary:node0}

```

The primary node validates the device configuration to ensure that it can be committed using the new software version. If anything goes wrong, the ISSU aborts and error messages are displayed.

## Installation-Related Errors

### IN THIS SECTION

- [Problem | 531](#)
- [Solution | 532](#)

### Problem

### Description

The install image file does not exist or the remote site is inaccessible.

## Solution

Use the following error messages to understand the installation-related problems:

```
error: File does not exist: /var/tmp/junos-srx5000-11.4X3.2-domest
error: Couldn't retrieve package /var/tmp/junos-srx5000-11.4X3.2-domest
```

ISSU downloads the install image as specified in the ISSU command as an argument. The image file can be a local file or located at a remote site. If the file does not exist or the remote site is inaccessible, an error is reported.

## Redundancy Group Failover Errors

### IN THIS SECTION

- [Problem | 532](#)
- [Solution | 532](#)

## Problem

### Description

Problem with automatic redundancy group (RG) failure.

### Solution

Use the following error messages to understand the problem:

```
failover all RG 1+ groups to node 0
error: Command failed. None of the redundancy-groups has been failed over.
Some redundancy-groups on node1 are already in manual failover mode.
Please execute 'failover reset all' first..
```

## Kernel State Synchronization Errors

### IN THIS SECTION

- [Problem | 533](#)
- [Solution | 533](#)

### Problem

### Description

Errors related to ksyncd.

### Solution

Use the following error messages to understand the issues related to ksyncd:

```
Failed to get kernel-replication error information from Standby Routing Engine.  
mgd_slave_peer_has_errors() returns error at line 4414 in mgd_package_issu.
```

ISSU checks whether there are any ksyncd errors on the secondary node (node 1) and displays the error message if there are any problems and aborts the upgrade.

## Managing Chassis Cluster ISSU-Related Problems

### IN THIS SECTION

- [Viewing ISSU Progress | 534](#)
- [Stopping ISSU Process if it Halts During an Upgrade | 535](#)
- [Recovering the Node in Case of a Failed ISSU | 536](#)

This topic includes the following sections:

## Viewing ISSU Progress

### IN THIS SECTION

- Problem | 534
- Solution | 534

### Problem

### Description

Rather than wait for an ISSU failure, you can display the progress of the ISSU as it occurs, noting any message indicating that the ISSU was unsuccessful. Providing such messages to JTAC can help with resolving the issue.

### Solution

After starting an ISSU, issue the `show chassis cluster information issu` command. Output similar to the following is displayed indicating the progress of the ISSU for all Services Processing Units (SPUs).

```
Note: Any management session to secondary node will be disconnected.
Shutdown NOW!
[pid 2480]
ISSU: Backup RE Prepare Done
Waiting for node1 to reboot.
Current time: Tue Apr 22 14:37:32 2014
Max. time to complete: 15min 0sec.
Note: For real time ISSU status, open a new management session and run
<show chassis cluster information issu> for detail information
node1 booted up.
Waiting for node1 to become secondary
Current time: Tue Apr 22 14:40:32 2014
Max. time to complete: 60min 0sec.
Note: For real time ISSU status, open a new management session and run
<show chassis cluster information issu> for detail information
node1 became secondary.
Waiting for node1 to be ready for failover
```

```

ISSU: Preparing Daemons
Current time: Tue Apr 22 14:41:27 2014
Max. time to complete: 60min 0sec.
Note: For real time ISSU status, open a new management session and run
<show chassis cluster information issu> for detail information
Secondary node1 ready for failover.
Installing package '/var/tmp/junos-srx5000-12.1I20140421_srx_12q1_x47.0-643920-domestic.tgz' ...
Verified SHA1 checksum of issu-indb.tgz
Verified junos-boot-srx5000-12.1I20140421_srx_12q1_x47.0-643920.tgz signed by
PackageDevelopment_12_1_0
Verified junos-srx5000-12.1I20140421_srx_12q1_x47.0-643920-domestic signed by
PackageDevelopment_12_1_0

```

## Stopping ISSU Process if it Halts During an Upgrade

### IN THIS SECTION

- [Problem | 535](#)
- [Solution | 535](#)

### Problem

### Description

The ISSU process halts in the middle of an upgrade.

### Solution

If the ISSU fails to complete and only one device in the cluster is upgraded, you can roll back to the previous configuration on the upgraded device alone by issuing one of the following commands on the upgraded device:

- `request chassis cluster in-service-upgrade abort` to terminate the ISSU on both nodes.
- `request system software rollback node node-id reboot` to roll back the image.
- `request system reboot` to reboot the rolled back node.

## Recovering the Node in Case of a Failed ISSU

### IN THIS SECTION

- [Problem | 536](#)
- [Solution | 536](#)

### Problem

### Description

The ISSU procedure stops progressing.

### Solution

Open a new session on the primary device and issue the `request chassis cluster in-service-upgrade abort` command.

This step terminates an in-progress ISSU . This command must be issued from a session other than the one on which you issued the `request system in-service-upgrade` command that launched the ISSU. If the node is being upgraded, this command cancels the upgrade. The command is also helpful in recovering the node in case of a failed ISSU.

When an ISSU encounters an unexpected situation that necessitates a termination, the system message provides you with detailed information about when and why the upgrade stopped along with recommendations for the next steps to take.

For example, the following message is issued when a node fails to become RG-0 secondary when it boots up:

```
Rebooting Secondary Node
Shutdown NOW!
[pid 2120]
ISSU: Backup RE Prepare Done
Waiting for node1 to reboot.
node1 booted up.
Waiting for node1 to become secondary
error: wait for node1 to become secondary failed (error-code: 5.1)
ISSU aborted. But, both nodes are in ISSU window.
Please do the following:
```

1. Log on to the upgraded node.
2. Rollback the image using rollback command with node option  
Note: Not using the 'node' option might cause the images on both nodes to be rolled back
3. Make sure that both nodes (will) have the same image
4. Ensure the node with older image is primary for all RGs
5. Abort ISSU on both nodes
6. Reboot the rolled back node  
{primary:node0}

If you attempt to upgrade a device pair running a Junos OS release earlier than Release 9.6, ISSU fails without changing anything on either device in the cluster. Devices running Junos OS releases earlier than Release 9.6 must be upgraded separately using individual device upgrade procedures.

If the secondary device experiences a power-off condition before it boots up using the new image specified when the ISSU was initiated, the newly upgraded device will still be waiting to end the ISSU after power is restored. To end the ISSU process on Routing Engine3 and Routing Engine2 on both the nodes, issue the request chassis cluster in-service-upgrade abort command.

### 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
17.4R1	Starting with Junos OS Release 17.4R1, SRX4600 devices support ISSU.
17.4R1	Starting with Junos OS Release 17.4R1, the hold timer for the initial reboot of the secondary node during the ISSU process is extended from 15 minutes (900 seconds) to 45 minutes (2700 seconds) in chassis clusters on SRX1500, SRX4100, SRX4200, and SRX4600 devices.
15.1X49-D80	Starting with Junos OS Release 15.1X49-D80, SRX4100 and SRX4200 devices support ISSU.
15.1X49-D70	Starting with Junos OS Release 15.1X49-D70, SRX1500 devices support ISSU.

### RELATED DOCUMENTATION

[Upgrading Individual Devices in a Chassis Cluster Separately | 510](#)

[Upgrading Devices in a Chassis Cluster Using ICU | 510](#)

[Disabling a Chassis Cluster | 538](#)

## Disabling a Chassis Cluster

If you want to operate the SRX Series Firewall back as a standalone device or to remove a node from a chassis cluster, you must disable the chassis cluster.

The node could fail to load the configuration if you disable the cluster and the configuration contains groups 'node0', 'node1' defining key resources. These groups (automatically generated in case of a cluster) will exist no more and the resulting configuration could be inconsistent.

In such events, the node will come up partially amnesiac (logins will be remembered).

To disable chassis cluster, enter the following command:

```
{primary:node1}
user@host> set chassis cluster disable reboot
Successfully disabled chassis cluster. Going to reboot now.
```

After the system reboots, the chassis cluster is disabled.

After the chassis cluster is disabled using this CLI command, you do not have a similar CLI option to enable it back.

To restore the chassis cluster, set the cluster-id by entering the following command:

```
{primary:node1}
user@host> set chassis cluster cluster-id cluster-id-number node node-number reboot
```

When setting the cluster, the secondary node PFEMAN process will restart.

You can also use the below CLI commands to disable chassis cluster:

- To disable cluster on node 0:

```
user@host> set chassis cluster cluster-id 0 node 0 reboot
```

- To disable cluster on node 1:

```
user@host> set chassis cluster cluster-id 0 node 1 reboot
```

Setting cluster-id to zero disables clustering on a device.



## RELATED DOCUMENTATION

[Upgrading Individual Devices in a Chassis Cluster Separately | 510](#)

---

[Upgrading Devices in a Chassis Cluster Using ICU | 510](#)

---

*set chassis cluster disable reboot*

---

*set chassis cluster cluster-id node node-number reboot*



CHAPTER

# Troubleshooting

---

[Troubleshooting a Control Link Failure in an SRX Chassis Cluster | 541](#)

[Troubleshooting a Fabric Link Failure in an SRX Chassis Cluster | 543](#)

[Troubleshooting a Redundancy Group that Does Not Fail Over in an SRX Chassis Cluster | 546](#)

[Troubleshooting an SRX Chassis Cluster with One Node in the Primary State and the Other Node in the Disabled State | 551](#)

[Troubleshooting an SRX Chassis Cluster with One Node in the Primary State and the Other Node in the Lost State | 555](#)

[Troubleshooting an SRX Chassis Cluster with One Node in the Hold State and the Other Node in the Lost State | 558](#)

[Troubleshooting Chassis Cluster Management Issues | 562](#)

[Data Collection for Customer Support | 587](#)

---

# Troubleshooting a Control Link Failure in an SRX Chassis Cluster

## IN THIS SECTION

- [Problem | 541](#)
- [Diagnosis | 542](#)

## Problem

### Description

The control link fails to come up in an SRX chassis cluster.

### Environment

SRX chassis cluster

### Symptoms

The chassis cluster is down due to a control link failure. The status of the control link is displayed as down in the output of the `show chassis cluster interfaces` command. Here are sample outputs for an SRX branch device and a high-end SRX device.

```
{primary:node0}  
root@J-SRX-branch> show chassis cluster interfaces  
Control link 0 name: fxp1  
Control link status: Down
```

```
{primary:node0}  
root@J-SRX-highend> show chassis cluster interfaces  
Control link 0 name: em0
```

```
Control link 1 name: em1
Control link status: Down
```

## Diagnosis

1. Are the control link ports connected through a switch?
  - Yes: Remove the switch and connect the control link ports directly. Reboot the secondary node and check whether the control link is up.
    - If the link is up, then there might be an issue in the chassis cluster setup on the Layer 2 switch network. See [SRX Series Gateway Cluster Deployment in Layer 2 Network](#).
    - If the link is down, proceed to Step 2.
  - No: Proceed to Step 2.
2. Are the link LEDs for the control link ports on both the nodes lit green?
  - Yes: Proceed to Step 4.
  - No: The control link cable might be faulty. Proceed to Step 3.
3. Change the cable connecting the control link ports and check the link LED. Is the LED lit green?
  - Yes: This indicates that the original cable was faulty. Reboot both the nodes simultaneously to come out of the bad state. If the control link does not come up after the reboot, proceed to Step 4.
  - No: Open a case with your technical support representative to resolve the issue. Proceed to ["Data Collection for Customer Support" on page 587](#).
4. Is this device an SRX5400, SRX5600, or SRX5800?
  - Yes: Reconfigure the control link on a different Services Processing Card (SPC), connect the cable to the new port, and reboot both the nodes.
    - a. Check whether the control link status is up:
      - If the link is up, the issue is resolved.

There might be a hardware issue with the SPC. Open a case with your technical support representative to resolve the hardware issue. Proceed to ["Data Collection for Customer Support" on page 587](#).
  - If the link is still down, the transceivers might be faulty. Proceed to Step 5.

- No: Verify that the correct ports are connected (see the following table):

Port Type	Port
Management (fxp0)	An Ethernet port on the Routing Engine
HA Control (fxp1 or em0/em1)	<ul style="list-style-type: none"> <li>• Port 0 (fiber only) on SPC—em0</li> <li>• Port 1 (fiber only) on SPC—em1</li> </ul>
Fabric (fab0 and fab1)	Any available ge or xe interface (fiber only)

If the ports are connected correctly, and the link is still down, proceed to Step 5.

5. If the control link port is an SFP or XFP port, change the transceivers on both the nodes. Ensure that you use transceivers provided by Juniper Networks and that the transceivers are of the same type (such as LX or SX). Is the control link up now?

- Yes: The issue is resolved.

The transceiver might be faulty. Open a case with your technical support representative to resolve the issue with the transceivers. Proceed to ["Data Collection for Customer Support" on page 587](#).

- No: Continue to troubleshoot this issue with your technical support representative. Proceed to ["Data Collection for Customer Support" on page 587](#).

## Troubleshooting a Fabric Link Failure in an SRX Chassis Cluster

### IN THIS SECTION

- Problem | 544
- Diagnosis | 545

## Problem

### Description

The fabric link fails to come up in an SRX chassis cluster.

### Environment

SRX chassis cluster

### Symptoms

The status of the fabric link is displayed as down in the output of the `show chassis cluster interfaces` command. Here are sample outputs for an SRX branch device and a high-end SRX device.

```
{primary:node0}
root@SRX_Branch> show chassis cluster interfaces
Control link 0 name: fxp1
Control link status: Up

Fabric interfaces:
Name Child-interface Status
fab0 ge-0/0/2 down
fab0
fab1 ge-9/0/2 down
fab1
Fabric link status: down
```

```
{primary:node0}
root@SRX_HighEnd> show chassis cluster interfaces
Control link 0 name: em0
Control link 1 name: em1
Control link status: up

Fabric interfaces:
Name Child-interface Status
fab0 ge-0/0/5 down
fab0
Fabric link status: down
```

## Diagnosis

### 1. Are the fabric link ports connected through a switch?

- Yes: Remove the switch and connect the fabric link ports directly. Reboot the secondary node and check whether the fabric link is up.
  - If the link is up, then there might be an issue in the chassis cluster setup on the Layer 2 switch network. See [SRX Series Gateway Cluster Deployment in Layer 2 Network](#).
  - If the link is down, proceed to Step 2.
- No: Proceed to Step 2.

### 2. Are the link LEDs for the fabric link ports on both the nodes lit green?

- Yes: The physical link is up, but the fabric packets are not being processed. To eliminate possible issues with the port:
  - a. Reconfigure the fabric link on a different port, connect the cable to the new port, and reboot the secondary node.
  - b. Check whether the fabric link status is up:
    - If the link is up, the issue is resolved.

There might be a hardware issue with the onboard ports or interface module ports on which you had previously configured the fabric link. Verify the interface statistics by using the `show interfaces interface-name` command. Open a case with your technical support representative to resolve the issue with the ports. Proceed to ["Data Collection for Customer Support" on page 587](#).
    - If the link is still down, open a case with your technical support representative. Proceed to ["Data Collection for Customer Support" on page 587](#).
- No: The fabric link cable might be faulty. Proceed to Step 3.

### 3. Change the cable connecting the fabric link ports and check the link LED. Is the LED lit green?

- Yes: This indicates that the original cable was faulty. Reboot both the nodes simultaneously to come out of the bad state. If the fabric link does not come up after the reboot:
  - a. Reconfigure the fabric link on a different port, connect the cable to the new port, and reboot the secondary node.
  - b. Check whether the fabric link status is up:
    - If the link is up, the issue is resolved.

There might be a hardware issue with the onboard ports or interface module ports on which you had previously configured the fabric link. Verify the interface statistics by using the `show interfaces interface-name` command. Open a case with your technical support representative to resolve the issue with the ports. Proceed to ["Data Collection for Customer Support" on page 587](#).

- If the link is still down, open a case with your technical support representative. Proceed to ["Data Collection for Customer Support" on page 587](#).
  - No: The transceivers might be faulty. Proceed to step 4.
4. If the fabric link port is an SFP or XFP port, change the transceivers on both the nodes. Ensure that you use transceivers provided by Juniper Networks and that the transceivers are of the same type (such as LX or SX). Is the fabric link up now?

- Yes: The issue is resolved.

The original transceivers used on the fabric link ports might be faulty. Open a case with your technical support representative to resolve the issue with the transceivers. Proceed to ["Data Collection for Customer Support" on page 587](#).

- No: Continue to troubleshoot this issue with your technical support representative. Proceed to ["Data Collection for Customer Support" on page 587](#).

## Troubleshooting a Redundancy Group that Does Not Fail Over in an SRX Chassis Cluster

### IN THIS SECTION

- Problem | [547](#)
- Diagnosis | [547](#)
- Resolution | [548](#)
- What's Next | [550](#)



## Problem

### Description

A redundancy group (RG) in a high-availability (HA) SRX chassis cluster does not fail over.

### Environment

SRX chassis cluster

## Diagnosis

From the command prompt of the SRX Series Services Gateway that is part of the chassis cluster, run the `show chassis cluster status` command.

Sample output:

```
Cluster ID: 1
Node          Priority   Status   Preempt   Manual failover

Redundancy group: 0 , Failover count: 0
node0         150      primary  no        no
node1         100      secondary no        no

Redundancy group: 1 , Failover count: 0
node0         255      primary  yes       no
node1         100      secondary yes       no
```

In the sample output check the priority of the redundancy group that does not fail over.

- If the Priority is 255 and the Manual failover field is yes, proceed to ["Redundancy Group Manual Failover" on page 548](#).
- If the priority is 0 or anything between 1 and 254, proceed to ["Redundancy Group Auto Failover" on page 549](#)

## Resolution

### Redundancy Group Manual Failover

1. Check whether a manual failover of the redundancy group was initiated earlier by using the `show chassis cluster status` command.

Sample output:

```
Cluster ID: 1
Node                Priority    Status    Preempt    Manual failover

Redundancy group: 0 , Failover count: 0
node0                150       primary   yes        no
node1                100       secondary yes        no

Redundancy group: 1 , Failover count: 0
node0                255       primary   no         yes
node1                100       secondary no         yes
```

In the sample output, Priority value of redundancy group 1 (RG1) is 255 and the status of Manual failover is yes, which means that a manual failover of the redundancy group was initiated earlier. You must reset the redundancy group priority.

**NOTE:** After a manual failover of a redundancy group, we recommend that you reset the manual failover flag in the cluster status to allow further failovers.

2. Reset the redundancy group priority by using the `request chassis cluster failover reset redundancy-group <1-128>`.

For example:

```
user@host> request chassis cluster failover reset redundancy-group 1
root@srx> request chassis cluster failover reset redundancy-group 1
node0:
-----
Successfully reset manual failover for redundancy group 1
```

```
node1:
```

```
-----  
No reset required for redundancy group 1.
```

3. This must resolve the issue and allow further redundancy group failovers. If these steps do not resolve the issue, proceed to section [Whats Next](#).
4. If you want to initiate a redundancy group x (redundancy groups numbered 1 through 128) failover manually, see [Understanding Chassis Cluster Redundancy Group Manual Failover](#).

### Redundancy Group Auto Failover

1. Check the configuration and link status of the control and fabric links by using the `show chassis cluster interfaces` command.

Sample output for a branch SRX Series Services Gateway:

```
{primary:node0}
root@SRX_Branch> show chassis cluster interfaces
Control link 0 name: fxp1
Control link status: Up

Fabric interfaces:
Name Child-interface Status
fab0 ge-0/0/2 down
fab0
fab1 ge-9/0/2 down
fab1
Fabric link status: down
```

Sample output for a high-end SRX Series Services Gateway:

```
{primary:node0}
root@SRX_HighEnd> show chassis cluster interfaces
Control link 0 name: em0
Control link 1 name: em1
Control link status: up
```

```
Fabric interfaces:
Name Child-interface Status
fab0 ge-0/0/5 down
fab0
Fabric link status: down
```

- If the control link is down, see KB article [KB20698](#) to troubleshoot and bring up the control link and proceed to "3" on page 550.
  - If the fabric link is down, see "[Troubleshooting a Fabric Link Failure in an SRX Chassis Cluster](#)" on page 543 to troubleshoot and bring up the fabric link and proceed to "3" on page 550.
2. Proceed to Step "3" on page 550 if both the control link and fabric link are up.
  3. Check the interface monitoring or IP monitoring configurations that are up. If the configurations are not correct rectify the configurations. If the configurations are correct proceed to step "4" on page 550.
  4. Check the priority of each node in the output of the `show chassis cluster status` command.
    - If the priority is 0, see KB article [KB16869](#) for JSRP (Junos OS Services Redundancy Protocol) chassis clusters and KB article [KB19431](#) for branch SRX Series Firewalls.
    - If the priority is 255, see "[Redundancy Group Manual Failover](#)" on page 548.
    - If the priority is between 1 and 254 and if still the redundancy group does not fail over, proceed to the section [Whats Next](#).

## WHAT'S NEXT

If these steps do not resolve the issue, see KB article [KB15911](#) for redundancy group failover tips.

If you wish to debug further, see KB article [KB21164](#) to check the debug logs.

To open a JTAC case with the Juniper Networks Support team, see "[Data Collection for Customer Support](#)" | [587](#) for the data you should collect to assist in troubleshooting before you open a JTAC case.

# Troubleshooting an SRX Chassis Cluster with One Node in the Primary State and the Other Node in the Disabled State

## IN THIS SECTION

- Problem | 551
- Diagnosis | 552

## Problem

### Description

The nodes of the SRX chassis cluster are in primary and disabled states.

### Environment

SRX chassis cluster

### Symptoms

One node of the cluster is in the primary state and the other node is in the disabled state. Run the `show chassis cluster status` command on each node to view the status of the node. Here is a sample output:

```
{primary:node0}
root@primary-srx> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
```

```
CF Config Sync monitoring      RE Relinquish monitoring
```

```
Cluster ID: 1
```

```
Node   Priority Status           Preempt Manual   Monitor-failures
```

```
Redundancy group: 0 , Failover count: 1
```

```
node0  255    primary      no    no    None
```

```
node1  129    disabled    no    no    None
```

```
Redundancy group: 1 , Failover count: 1
```

```
node0  255    primary      no    no    None
```

```
node1  129    disabled    no    no    None
```

## Diagnosis

1. Run the `show chassis cluster interfaces` command to verify the status of the control and fabric links. Are any of the links down?

Here are sample outputs for a branch SRX Series Firewall and a high-end SRX Series Firewall.

```
root@Branch-SRX> show chassis cluster interfaces
```

```
Control link 0 name: fxp1
```

```
Control link status: Up
```

```
Fabric interfaces:
```

```
Name Child-interface Status
```

```
fab0 ge-0/0/2 up
```

```
fab0 ge-2/0/6 up
```

```
fab1 ge-9/0/2 up
```

```
fab1 ge-11/0/6 up
```

```
Fabric link status: Up
```

```
{primary:node0}
```

```
root@High-end-SRX> show chassis cluster interfaces
```

```
Control link 0 name: em0
```

```
Control link 1 name: em1
```

```
Control link status: Up
```

```
Fabric interfaces:
Name Child-interface Status
fab0 ge-2/0/0 down
fab0
fab1
fab1
Fabric link status: Up
```

- Yes: See ["Troubleshooting a Fabric Link Failure in an SRX Chassis Cluster" on page 543](#) or ["Troubleshooting a Control Link Failure in an SRX Chassis Cluster" on page 541](#).
  - No: Proceed to Step 2.
2. Reboot the disabled node. Does the node come up in the disabled state after the reboot?
- Yes: There might be hardware issues. Proceed to Step 3.
  - No: The issue is resolved.
3. Check the node for any hardware issues. Run the `show chassis fpc pic-status` command on both nodes, and ensure that the FPCs are online. Do you see the status of any FPC listed as Present, OK, or Offline?

Here is a sample output.

```
{primary:node1}
root@J-SRX> show chassis fpc pic-status
node0:
-----
Slot 0 Online FPC
  PIC 0 Online 4x GE Base PIC
Slot 2 Online FPC
  PIC 0 Online 24x GE gPIM
Slot 6 Online FPC
  PIC 0 Online 2x 10G gPIM

node1:
-----
Slot 0 Online FPC
  PIC 0 Online 4x GE Base PIC
Slot 2 Online FPC
  PIC 0 Online 24x GE gPIM
Slot 6 Online FPC
  PIC 0 Online 2x 10G gPIM
```

- Yes: Reseat the cards and reboot the node. If this does not resolve the issue, open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#).
- No: Proceed to Step 4.

4. Run the `show chassis cluster statistics` on both nodes, and analyze the output.

```
{primary:node0}
root@J-SRX> show chassis cluster statistics
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 418410
    Heartbeat packets received: 418406
    Heartbeat packet errors: 0
Fabric link statistics:
  Probes sent: 418407
  Probes received: 414896
  Probe errors: 0
```

Does the `Heartbeat packets received` field show a non-increasing value or zero (0), or does the `Heartbeat packet errors` field show a non-zero value?

- Yes: Open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#).
  - No: Proceed to Step 5.
5. Configure `set chassis cluster no-fabric-monitoring` (hidden option) and commit the configuration to temporarily disable fabric monitoring during the troubleshooting process. Reboot the disabled node. After the node reboots, run the `show chassis cluster statistics` command. Are the probes still lost?
- Yes: Open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#)
  - No: Delete the `set chassis cluster no-fabric-monitoring` configuration, and verify that everything is operational. If you notice any issue, open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#)



# Troubleshooting an SRX Chassis Cluster with One Node in the Primary State and the Other Node in the Lost State

## IN THIS SECTION

- [Problem | 555](#)
- [Diagnosis | 556](#)

## Problem

### Description

The nodes of the SRX chassis cluster are in primary and lost states.

### Environment

SRX chassis cluster

### Symptoms

One node of the cluster is in the primary state and the other node is in the lost state. Run the `show chassis cluster status` command on each node to view the status of the node. Here is a sample output:

```
{primary:node0}
root@primary-srx> show chassis cluster status
Cluster ID: 1
Node                Priority      Status      Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
node0                100         primary    no       no
node1                0          lost       no       no
```

```
Redundancy group: 1 , Failover count: 1
```

node0	100	primary	no	no
node1	0	lost	no	no

## Diagnosis

- Is the node that is in the lost state powered on?
  - Yes: Are you able to access the node that is in the lost state through a console port? Do not use Telnet or SSH to access the node.
    - If you are able to access the node, proceed to Step 3.
    - If you are unable to access the node and if the device is at a remote location, access the node through a console for further troubleshooting. If you have console access, but do not see any output, it might indicate a hardware issue. Open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#).
  - No: Power on the node and proceed to Step 2.
- After both nodes are powered on, run the `show chassis cluster status` command again. Is the node still in the lost state?
  - Yes: Are you able to access the node that is in the lost state through a console port? Do not use Telnet or SSH to access the node.
    - If you are able to access the node, proceed to Step 3.
    - If you are unable to access the node and if the node is at a remote location, access the node through a console for further troubleshooting. If you have console access, but do not see any output, it might indicate a hardware issue. Open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#).
  - No: Powering on the device has resolved the issue.
- Connect a console to the primary node, and run the `show chassis cluster status` command. Does the output show this node as primary and the other node as lost?
  - Yes: This might indicate a split-brain scenario. Each node would show itself as primary and the other node as lost. Run the following commands to verify which node is processing the traffic:
    - `show security monitoring`

- show security flow session summary
- monitor interface traffic

Isolate the node that is not processing the traffic. You can isolate the node from the network by removing all the cables except the control and fabric links. Proceed to Step 4.

- No: Proceed to Step 4.
4. Verify that all the FPCs are online on the node that is in the lost state by running the `show chassis fpc pic-status` command. Are all the FPCs online?
    - Yes: Proceed to Step 5.
    - No: Open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#).
  5. Are the nodes connected through a switch?
    - Yes: See ["Troubleshooting a Fabric Link Failure in an SRX Chassis Cluster" on page 543](#) and ["Troubleshooting a Control Link Failure in an SRX Chassis Cluster" on page 541](#).
    - No: Proceed to Step 6.
  6. Create a backup of the configuration from the node that is currently primary:

```
{primary:node0}
root@primary-srx# show configuration | save /var/tmp/cfg-bkp.txt
```

Copy the configuration to the node that is in the lost state, and load the configuration:

```
root@lost-srx# load override <terminal or filename>
```

**NOTE:** If you use the `terminal` option, paste the complete configuration into the window. Make sure that you use **Ctrl+D** at the end of the configuration.

If you use the `filename` option, provide the path to the configuration file (for example: `/var/tmp/Primary_saved.conf`), and press Enter.

When you connect to the node in the lost state through a console, you might see the state as either primary or hold/disabled. If the node is in the hold/disabled state, a fabric link failure might have occurred before the device went into the lost state. To troubleshoot this issue, follow the steps in ["Troubleshooting a Fabric Link Failure in an SRX Chassis Cluster" on page 543](#).

Commit the changes after the configuration is loaded. If the problem persists, then replace the existing control and fabric links on this device with new cables and reboot the node:

```
{primary:node1}[edit]  
root@lost-srx# request system reboot
```

Is the issue resolved?

- No: Open a case with your technical support representative for further troubleshooting. See ["Data Collection for Customer Support" on page 587](#).

## Troubleshooting an SRX Chassis Cluster with One Node in the Hold State and the Other Node in the Lost State

### IN THIS SECTION

- [Problem | 558](#)
- [Cause | 559](#)
- [Resolution | 560](#)

## Problem

### Description

The nodes of the SRX chassis cluster are in hold and lost states.

### Environment

SRX chassis cluster

## Symptoms

One node of the SRX chassis cluster is in the hold state and the other node is in the lost state after you connect the cables and reboot the devices in cluster mode. Run the `show chassis cluster status` command on each node to view the status of the node. Here is a sample output:

```
{hold:node0} user@node0> show chassis cluster status
```

```
Cluster ID: 1, Redundancy-group: 0
Node name  Priority Status Preempt Manual failover
node0      100    hold   No    No
node1      1      lost   No    No
```

```
{hold:node1}
```

```
user@node1> show chassis cluster status
```

```
Cluster ID: 1, Redundancy-group: 0
Node name  Priority Status Preempt Manual failover
node0      100    lost   No    No
node1      1      hold   No    No
```

If the status of a node is `hold`, the node is not ready to operate in a chassis cluster.

**NOTE:** This issue does not impact high-end SRX Series Firewalls because these devices have dedicated control and management ports.

## Cause

When you boot a branch SRX Series Firewall in cluster mode, two revenue interfaces (depending upon the model of the device) are designated for the out-of-band management link (fxp0) and control link (fxp1) of the chassis cluster. The fxp0 and fxp1 ports cannot be used for transit traffic.

If you configure the fxp0 and fxp1 ports, the chassis cluster goes into the hold/lost state. The following table lists the ports that are designated as fxp0 and fxp1 ports for branch SRX Series Firewalls:

**Table 39: fxp0 and fxp1 Ports on Branch SRX Series Devices**

Device	Management (fxp0)	HA Control (fxp1)	Fabric (fab0 and fab1)—must be configured
SRX300	ge-0/0/0	ge-0/0/1	Any ge interface
SRX320	ge-0/0/0	ge-0/0/1	Any ge interface
SRX340, SRX345, and SRX380	MGMT	ge-0/0/1	Any ge interface

## Resolution

### Remove the Configuration on a Device Running the Factory-Default Configuration

The factory-default configuration includes configuration for the interfaces that are transformed into fxp0 and fxp1 interfaces. You must delete these configurations before enabling chassis cluster mode. A device can have the factory-default configuration in the following scenarios:

- Typically, new devices are used in a chassis cluster. These new devices ship with the factory-default configuration, which includes configuration for the interfaces.
- If a device that is in chassis cluster mode crashes, the device might come up with the factory-default configuration.

To remove the configuration on the interfaces, delete the factory-default configuration and reconfigure the device.



**CAUTION:** The following procedure removes the current configuration.

1. Log in to the device and enter the configuration mode.

2. Run the delete command to delete the current configuration from the device.

```
root# delete
This will delete the entire configuration
Delete everything under this level? [yes,no] (no) yes
```

3. Configure the root password and commit the configuration:

```
root# set system root-authentication plain-text-password
root# commit
```

## Remove the Configuration on a Device Operating as a Standalone Device

If the device is currently running in a production environment, then check whether the interfaces that are designated as the fxp0 and fxp1 interfaces are configured. To determine which interfaces are transformed into fxp0 and fxp1 interfaces, see [Table 39 on page 560](#).

1. Run the following commands to list the configuration for the fxp0 and fxp1 interfaces:

```
show | display set | match <physical interface of the control port (fxp1)>
show | display set | match <physical interface of the management port (fxp0)>
```

For example:

```
show configuration | display set | match ge-0/0/0
show configuration | display set | match ge-0/0/1
```

2. Delete all the configurations related to the interfaces from every configuration hierarchy.

You can also choose to delete the entire configuration and reconfigure the device:

```
root# delete
```

# Troubleshooting Chassis Cluster Management Issues

## IN THIS SECTION

- [Unable to Manage an SRX Series Chassis Cluster Using the Management Port or Revenue Ports | 562](#)
- [Unable to Manage the Secondary Node of a Chassis Cluster Using J-Web | 574](#)
- [Unable to Manage an SRX Series Chassis Cluster Using fxp0 When the Destination in the Backup Router is 0/0 | 576](#)
- [Unable to Upgrade a Chassis Cluster Using In-Service Software Upgrade | 582](#)
- [Configuring backup-router Command on Chassis Cluster | 584](#)
- [Unable to Upgrade a Chassis Cluster Using In-Service Software Upgrade | 585](#)

## Unable to Manage an SRX Series Chassis Cluster Using the Management Port or Revenue Ports

### IN THIS SECTION

- [Problem | 562](#)
- [Diagnosis | 563](#)
- [Resolution | 563](#)

### Problem

### Description

Cannot manage the SRX Series chassis cluster using the management port or revenue ports.

### Environment

SRX Series chassis cluster



## Diagnosis

### 1. Which node in the chassis cluster are you using to manage the cluster?

- Primary node—Proceed to:
  - Manage the Chassis Cluster Using J-Web.

**NOTE:** You can use J-Web to manage only the primary node.

- Manage the Chassis Cluster Using the Revenue Port or fxp0 Management Port.

**NOTE:** You can use the revenue port or fxp0 management port to manage the primary node.

- Secondary node—Proceed to "[Manage the Chassis Cluster Using the fxp0 Management Port](#)" on [page 572](#)

**NOTE:** You can manage the secondary node only by using the fxp0 management port.

## Resolution

### Manage the Chassis Cluster Using J-Web

**NOTE:** You can use J-Web to manage only the primary node.

1. Connect a console to the primary node.
2. Using the CLI, run the **show system services web-management** command.
3. Check whether the loopback interface (lo0) is configured under the Web management HTTP/HTTPS configuration. See [web-management \(System Services\)](#).
4. If the loopback interface (lo0) is configured under the Web management HTTP/HTTPS configuration, remove the loopback interface by running the `delete system services web-management http interface lo.0` command.
5. Commit the change, and check whether you can now manage the chassis cluster.

6. If you still cannot manage the chassis cluster, proceed to ["Manage Chassis Cluster Using the Revenue Port or fxp0 Management Port"](#) on page 569.

## Manage Chassis Cluster Using the Revenue Port or fxp0 Management Port

**NOTE:** You can use both the revenue port or fxp0 management port to manage the primary node.

1. Connect to a console using the revenue port of the primary node which you want to use as a management interface.
2. Verify the configuration of the management interface:
  - a. Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **host-inbound-traffic** hierarchy level in the relevant zone:

```
zones {
  security-zone trust {
    host-inbound-traffic {
      system-services {
        any-service;
      }
      protocols {
        all;
      }
    }
    interfaces {
      reth0.0
      reth0.1;
    }
  }
}
```

- b. Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **system services** hierarchy level:

```
{primary:node1}[edit]
root# show system services {
  http;
  ssh;
```

```
telnet;
}
```

### 3. Does ping to the management interface work?

- **Yes:** See [Unable to Manage an SRX Series Chassis Cluster Using fxp0 When the Destination in the Backup Router is 0/0](#). If this solution doesn't work, proceed to ["What's Next" on page 573](#) to open a case with Juniper Networks technical support.
- **No:** Proceed to step ["4" on page 565](#).

### 4. Using the CLI, run the **show interfaces terse** command:

In the output, is the status of FXP0 interface Up, and does it provide an IP address?

- **Yes:** Proceed to step ["5" on page 566](#).
- **No:** Verify the following:
  - a. Using the CLI, verify that the fxp0 interface is configured correctly: **show groups**.

Sample output:

```
root@srx# show groups
node0 {
  system {
    host-name SRX3400-1;
    backup-router 192.168.1.254 destination 0.0.0.0/0;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet {
          address 192.168.1.1/24;
        }
      }
    }
  }
}
node1 {
  system {
    host-name SRX3400-2;
    backup-router 192.168.1.254 destination 0.0.0.0/0;
  }
  interfaces {
```



6. Using the CLI, check whether there is an ARP entry for the management device on the services gateway: `show arp no-resolve | match <ip>`.
  - a. **Yes:** Check whether the chassis cluster has multiple routes to the management device: `show route <device-ip>`.
    - **Yes:** There could be routes to the management device through the fxp0 interface and other interface leading to asymmetric routing. Proceed to ["What's Next" on page 573](#) to open a case with Juniper Networks technical support.
    - **No:** Proceed to ["Manage the Chassis Cluster Using the fxp0 Management Port" on page 572](#).
  - b. **No:** Proceed to ["What's Next" on page 573](#) to open a case with Juniper Networks technical support.

## Manage the Chassis Cluster Using the fxp0 Management Port

**NOTE:** You can use only the fxp0 management port to manage the secondary node.

1. Verify the configuration of management interface on the secondary node:
  - Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **host-inbound-traffic** hierarchy level:

```
zones {
  security-zone trust {
    host-inbound-traffic {
      system-services {
        any-service;
      }
      protocols {
        all;
      }
    }
    interfaces {
      reth0.0
      reth0.1;
    }
  }
}
```

- Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **system services** hierarchy level:

```
{primary:node1}[edit]
root# show system services {
    http;
    ssh;
    telnet;
}
```

See [Unable to Manage an SRX Series Chassis Cluster Using fxp0 When the Destination in the Backup Router is 0/0 and Configuring backup-router Command on Chassis Cluster](#) for more information about the configuration guidelines.

If the configuration is correct and you still cannot manage the chassis cluster, proceed to step "2" on [page 568](#).

2. Are the IP addresses of the fxp0 interfaces of the primary node and the secondary node in the same subnet?
  - **Yes:** Proceed to ["What's Next" on page 573](#).
  - **No:** Configure the fxp0 interfaces of the primary node and the secondary node in the same subnet. Go to step "1" on [page 567](#) and verify the configuration.

### What's Next

- If the issue persists, see KB Article [KB20795](#).
- If you wish to debug further, see KB Article [KB21164](#) to check the debug logs.
- To open a JTAC case with the Juniper Networks support team, see [Data Collection for Customer Support](#) for the data you should collect to assist in troubleshooting prior to opening a JTAC case.

### Manage the Chassis Cluster Using J-Web

**NOTE:** You can use J-Web to manage only the primary node.

1. Connect a console to the primary node.
2. Using the CLI, run the **show system services web-management** command.

3. Check whether the loopback interface (lo0) is configured under the Web management HTTP/HTTPS configuration. See [web-management \(System Services\)](#).
4. If the loopback interface (lo0) is configured under the Web management HTTP/HTTPS configuration, remove the loopback interface by running the `delete system services web-management http interface lo.0` command.
5. Commit the change, and check whether you can now manage the chassis cluster.
6. If you still cannot manage the chassis cluster, proceed to "[Manage Chassis Cluster Using the Revenue Port or fxp0 Management Port](#)" on page 569.

## Manage Chassis Cluster Using the Revenue Port or fxp0 Management Port

**NOTE:** You can use both the revenue port or fxp0 management port to manage the primary node.

1. Connect to a console using the revenue port of the primary node which you want to use as a management interface.
2. Verify the configuration of the management interface:
  - a. Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **host-inbound-traffic** hierarchy level in the relevant zone:

```
zones {
  security-zone trust {
    host-inbound-traffic {
      system-services {
        any-service;
      }
      protocols {
        all;
      }
    }
    interfaces {
      reth0.0
      reth0.1;
    }
  }
}
```

- b. Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **system services** hierarchy level:

```
{primary:node1}[edit]
root# show system services {
    http;
    ssh;
    telnet;
}
```

3. Does ping to the management interface work?

- **Yes:** See [Unable to Manage an SRX Series Chassis Cluster Using fxp0 When the Destination in the Backup Router is 0/0](#). If this solution doesn't work, proceed to ["What's Next" on page 573](#) to open a case with Juniper Networks technical support.
- **No:** Proceed to step ["4" on page 565](#).

4. Using the CLI, run the **show interfaces terse** command:

In the output, is the status of FXP0 interface Up, and does it provide an IP address?

- **Yes:** Proceed to step ["5" on page 566](#).
- **No:** Verify the following:
  - a. Using the CLI, verify that the fxp0 interface is configured correctly: **show groups**.

Sample output:

```
root@srx# show groups
node0 {
    system {
        host-name SRX3400-1;
        backup-router 192.168.1.254 destination 0.0.0.0/0;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 192.168.1.1/24;
                }
            }
        }
    }
}
```



```

    }
  }
  node1 {
    system {
      host-name SRX3400-2;
      backup-router 192.168.1.254 destination 0.0.0.0/0;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.168.1.2/24;
          }
        }
      }
    }
  }
}
apply-groups "${NODE}";
system {
  services {
    ftp;
    ssh;
    telnet;
  }
}

```

b. Check the condition of the cable that is connected to the **fxp0** interface. Is the cable in good condition?

- **Yes:** Proceed to the next step.
- **No:** Replace the cable and try to manage the chassis cluster. If you still cannot manage the chassis cluster, proceed to the next step.

c. Using the CLI, check for incrementing error counters: **show interfaces fxp0.0 extensive**.

- **Yes:** If you find any errors in the output, proceed to ["What's Next" on page 573](#) to open a case with Juniper Networks technical support.
- **No:** If there are no errors in the output and if you still cannot manage the chassis cluster, proceed to step ["5" on page 566](#).

5. Check whether the IP address of the **fxp0** interface and the IP address of the management device are in the same subnet.

- **Yes:** Proceed to the step ["6" on page 567](#).
  - **No:** Using the CLI, check if there is a route for the management device IP address: **show route <management device IP>**:
    - a. If a route does not exist for the management device IP address, add a route for the management subnet in the **inet.0** table with the next-hop as the backup router IP address.
6. Using the CLI, check whether there is an ARP entry for the management device on the services gateway: **show arp no-resolve | match <ip>**.
- a. **Yes:** Check whether the chassis cluster has multiple routes to the management device: **show route <device-ip>**.
    - **Yes:** There could be routes to the management device through the fxp0 interface and other interface leading to asymmetric routing. Proceed to ["What's Next" on page 573](#) to open a case with Juniper Networks technical support.
    - **No:** Proceed to ["Manage the Chassis Cluster Using the fxp0 Management Port" on page 572](#).
  - b. **No:** Proceed to ["What's Next" on page 573](#) to open a case with Juniper Networks technical support.

## Manage the Chassis Cluster Using the fxp0 Management Port

**NOTE:** You can use only the fxp0 management port to manage the secondary node.

1. Verify the configuration of management interface on the secondary node:
  - Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **host-inbound-traffic** hierarchy level:

```
zones {
  security-zone trust {
    host-inbound-traffic {
      system-services {
        any-service;
      }
      protocols {
        all;
      }
    }
  }
  interfaces {
```

```

        reth0.0
        reth0.1;
    }
}

```

- Verify that the required system services (SSH, Telnet, HTTP) are enabled at the **system services** hierarchy level:

```

{primary:node1}[edit]
root# show system services {
    http;
    ssh;
    telnet;
}

```

See [Unable to Manage an SRX Series Chassis Cluster Using fxp0 When the Destination in the Backup Router is 0/0 and Configuring backup-router Command on Chassis Cluster](#) for more information about the configuration guidelines.

If the configuration is correct and you still cannot manage the chassis cluster, proceed to step "2" on [page 568](#).

2. Are the IP addresses of the fxp0 interfaces of the primary node and the secondary node in the same subnet?
  - **Yes:** Proceed to ["What's Next" on page 573](#).
  - **No:** Configure the fxp0 interfaces of the primary node and the secondary node in the same subnet. Go to step "1" on [page 567](#) and verify the configuration.

## What's Next

- If the issue persists, see KB Article [KB20795](#).
- If you wish to debug further, see KB Article [KB21164](#) to check the debug logs.
- To open a JTAC case with the Juniper Networks support team, see [Data Collection for Customer Support](#) for the data you should collect to assist in troubleshooting prior to opening a JTAC case.

## Unable to Manage the Secondary Node of a Chassis Cluster Using J-Web

### IN THIS SECTION

- [Problem | 574](#)
- [Cause | 574](#)
- [Solution | 576](#)

### Problem

#### Description

Cannot manage the secondary node of a chassis cluster using the J-Web interface.

#### Environment

SRX Series chassis cluster

#### Symptoms

When in the Junos Services Redundancy Protocol (JSRP) chassis cluster mode, you cannot manage redundancy group 0 (RG0) on the secondary node using the J-Web interface.

#### Cause

- You can use the J-Web interface to manage redundancy group 0 only on the primary node.
- The processes that J-Web references are not running on the secondary node.

#### Example

The following example shows the output of syslog and system process on both node0 and node1 after RG0 was failed over from node1 to node0.

- On node1, web-management process (httpd-gk) was terminated (exited).
- On node0, web-management process (httpd-gk) was started.

- Two http-related processes (httpd-gk and httpd), run only on node0, which is the new primary node of RGO.

```
{secondary:node1}
root@SRX210HE-B> show chassis cluster status
Cluster ID: 1
Node                Priority      Status      Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
  node0              255         primary    no       yes
  node1              1           secondary  no       yes

Redundancy group: 1 , Failover count: 1
  node0              100        primary    yes      no
  node1              1          secondary  yes      no

{secondary:node1}
root@SRX210HE-B> show log log-any | grep web-management
Jul  5 11:31:52 SRX210HE-B init: web-management (PID 9660) started
Jul  5 12:00:37 SRX210HE-B init: web-management (PID 9660) SIGTERM sent
Jul  5 12:00:37 SRX210HE-B init: web-management (PID 9660) exited with status=0 Normal Exit

{primary:node0}
root@SRX210HE-A> show log log-any | grep web-management
Jul  5 12:00:37 SRX210HE-A init: web-management (PID 9498) started

{primary:node0}
root@SRX210HE-A> show system processes extensive node 0 | grep http
 9498 root      1  76    0 12916K  4604K select 0   0:00  0.00% httpd-gk
 9535 nobody    1  90    0  8860K  3264K select 0   0:00  0.00% httpd

{primary:node0}
root@SRX210HE-A> show system processes extensive node 1 | grep http
=> No httpd-gk and httpd processes running on node 1 (secondary node)
```

**NOTE:** This limits remote procedure calls (RPCs) from the J-Web logic, and subsequently, pages that can be issued from the secondary node.

## Solution

You can manage the secondary node of a chassis cluster using the CLI (SSH, telnet, and console). See [Manage the Chassis Cluster Using the fxp0 Management Port](#)

## Unable to Manage an SRX Series Chassis Cluster Using fxp0 When the Destination in the Backup Router is 0/0

### SUMMARY

This topic explains, with an example, how to manage an SRX Series chassis cluster configured using the backup-router configuration through the fxp0 interface.

### IN THIS SECTION

- [Problem | 576](#)
- [Cause | 578](#)
- [Solution | 578](#)

## Problem

### Description

The management device cannot manage the chassis cluster through an fxp0 interface, but it can ping both fxp0 interfaces.

### Sample Topology

The topology, IP addresses, and configuration are as follows:

- Primary fxp0: 192.168.1.1/24
- Secondary fxp0: 192.168.1.2/24
- Gateway for fxp0: 192.168.1.254
- Management device: 172.16.1.1/24

```
groups {
  node0 {
    system {
      host-name SRX5400-1;
      backup-router 192.168.1.254 destination 0.0.0.0/0;
```

```
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 192.168.1.1/24;
                }
            }
        }
    }
}
node1 {
    system {
        host-name SRX5400-2;
        backup-router 192.168.1.254 destination 0.0.0.0/0;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 192.168.1.2/24;
                }
            }
        }
    }
}
}
apply-groups "${NODE}";
system {
    services {
        ftp;
        ssh;
        telnet;
    }
}
```

## Environment

SRX Series chassis cluster

## Cause

There is a route for 172.16.1.1 through the interfaces other than the fxp0 interface on the cluster devices. We do not recommend using 0.0.0.0/0 as the backup-router destination. Ping works because the echo reply for an incoming echo request to the fxp0 interface is sent out following the route for 172.16.1.1 through interfaces other than fxp0, but Telnet fails.

## Solution

Remove the route for 172.16.1.1 in the routing table, and set a more specific backup-router destination in group node0/node1.

For example:

```
groups {
  node0 {
    ...
    backup-router 192.168.1.254 destination 172.16.1.1/32;
    ...
  }
  node1 {
    ...
    backup-router 192.168.1.254 destination 172.16.1.1/32;
    ...
  }
}
```

No changes are displayed in the routing table after the configuration is applied because the backup-router configuration is intended to facilitate management access on the backup node only. Access to the primary node is enabled through routing on the primary node.

Thus, when the backup router configuration is complete, you can see that a route is injected into the forwarding table on the secondary node. You cannot see the routing table on the secondary node because the routing subsystem does not run on the secondary node.

### Sample Output when the Backup router is Configured with Destination 0/0

- Routing table on primary node:

```
{primary:node0}[edit]
root@SRX5400-1# run show route

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```



```

192.168.1.0/24    *[Direct/0] 00:00:54
                 > via fxp0.0
192.168.1.1/32  *[Local/0] 00:00:54
                 Local via fxp0.0

```

- Forwarding table on secondary node with destination 0/0:

```

root@SRX3400-2# run show route forwarding-table
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index NhRef Netif
default          user  0 28:c0:da:a0:88:0 ucst  345  2 fxp0.0
default          perm  0                               rjct  36   1
0.0.0.0/32       perm  0                               dscd  34   1
192.168.1.0/24   intf  0                               rslv  344  1 fxp0.0
192.168.1.0/32   dest  0 192.168.1.0       recv  342  1 fxp0.0
192.168.1.2/32   intf  0 192.168.1.2       locl  343  2
192.168.1.2/32   dest  0 192.168.1.2       locl  343  2
192.168.1.254/32 dest  0 28:c0:da:a0:88:0 ucst  345  2 fxp0.0
192.168.1.255/32 dest  0 192.168.1.255    bcst  336  1 fxp0.0
224.0.0.0/4      perm  0                               mdsc  35   1
224.0.0.1/32     perm  0 224.0.0.1         mcst  31   1
255.255.255.255/32 perm  0                               bcst  32   1

Routing table: __master.anon__.inet
Internet:
Destination      Type RtRef Next hop          Type Index NhRef Netif
default          perm  0                               rjct  526  1
0.0.0.0/32       perm  0                               dscd  524  1
224.0.0.0/4      perm  0                               mdsc  525  1
224.0.0.1/32     perm  0 224.0.0.1         mcst  521  1
255.255.255.255/32 perm  0                               bcst  522  1

```

### Sample Output when the Backup router is Configured with Destination 172.16.1.1/32

- Routing table on primary node:

```
{primary:node0}[edit]
root@SRX5400-1# run show route

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.0/24    *[Direct/0] 00:17:51
                  > via fxp0.0
192.168.1.1/32  *[Local/0] 00:55:50
                  Local via fxp0.0
```

- Forwarding table on primary node:

**NOTE:** On the primary node, route 172.16.1.1/32 of the backup router is not shown in the sample output.

```
{primary:node0}[edit]
root@SRX5400-1# run show route forwarding-table
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index NhRef Netif
default          perm  0          Type Index NhRef Netif
0.0.0.0/32       perm  0          dscd  34    1
192.168.1.0/24   intf  0          rslv  334   1 fxp0.0
192.168.1.0/32   dest  0 192.168.1.0  recv  331   1 fxp0.0
192.168.1.1/32   intf  0 192.168.1.1  locl  332   2
192.168.1.1/32   dest  0 192.168.1.1  locl  332

2

192.168.1.3/32   dest  0 5c:5e:ab:16:e3:81 ucst  339   1 fxp0.0
192.168.1.6/32   dest  0 0:26:88:4f:c8:8  ucst  340   1 fxp0.0
192.168.1.11/32  dest  0 0:30:48:bc:9f:45 ucst  342   1 fxp0.0
192.168.1.254/32 dest  0 28:c0:da:a0:88:0 ucst  343   1 fxp0.0
192.168.1.255/32 dest  0 192.168.1.255  bcst  329   1 fxp0.0
224.0.0.0/4      perm  0          mdsc  35    1
224.0.0.1/32     perm  0 224.0.0.1      mcst  31    1
```

```
255.255.255.255/32 perm 0 bcst 32 1
```

```
Routing table: __master.anon__.inet
```

```
Internet:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	526	1	
0.0.0.0/32	perm	0		dscd	524	1	
224.0.0.0/4	perm	0		mdsc	525	1	
224.0.0.1/32	perm	0	224.0.0.1	mcst	521	1	
255.255.255.255/32	perm	0		bcst	522	1	

- Forwarding table on the secondary node:

**NOTE:** On the secondary node, route 172.16.1.1/32 of the backup router is shown in the sample output. This facilitates access to the secondary node through the fxp0 interface.

```
{secondary:node1}[edit]
```

```
root@SRX5400-2# run show route forwarding-table
```

```
Routing table: default.inet
```

```
Internet:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	36	1	
0.0.0.0/32	perm	0		dscd	34	1	
172.16.1.1/32	user	0	192.168.1.254	ucst	345	2	fxp0.0
192.168.1.0/24	intf	0		rslv	344	1	fxp0.0
192.168.1.0/32	dest	0	192.168.1.0	recv	342	1	fxp0.0
192.168.1.2/32	intf	0	192.168.1.2	locl	343	2	
192.168.1.2/32	dest	0	192.168.1.2	locl	343	2	
192.168.1.254/32	dest	0	28:c0:da:a0:88:0	ucst	345	2	fxp0.0
192.168.1.255/32	dest	0	192.168.1.255	bcst	336	1	fxp0.0
224.0.0.0/4	perm	0		mdsc	35	1	
224.0.0.1/32	perm	0	224.0.0.1	mcst	31	1	
255.255.255.255/32	perm	0		bcst	32	1	

```
Routing table: __master.anon__.inet
```

```
Internet:
```

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	526	1	
0.0.0.0/32	perm	0		dscd	524	1	
224.0.0.0/4	perm	0		mdsc	525	1	

```

224.0.0.1/32      perm    0 224.0.0.1      mcst  521    1
255.255.255.255/32 perm    0                bcst  522    1

```

If a particular subnet has a route configured through the backup router and a static route under routing-options, there could be problems accessing the fxp0 interface. In the example above, the issue with accessing the fxp0 interface from the management device occurs when :

- The same route exists as a static route and through the backup router.
- There is a static route that is more specific than the route through the backup router.

In the examples, when the routes from the primary node are synchronized to the secondary node's forwarding table, the route configured under static route takes precedence over the route under backup router. If 0/0 is configured under backup-router, the chances of a better matching route under static route is higher. Hence, we recommend avoiding 0/0 under backup router.

If you want to configure routes to the same destination using backup router as well as the static route, split the routes when configuring under backup-router. This makes the routes configured under backup router as the preferred routes and ensures that the fxp0 interface is accessible.

```

[edit routing-options static route]
0.0.0.0/0 next-hop 100.200.200.254;

[edit groups node0 ]
backup-router 192.168.1.254 destination [0.0.0.0/1 128.0.0.0/1];

```

## Unable to Upgrade a Chassis Cluster Using In-Service Software Upgrade

### IN THIS SECTION

- [Problem | 583](#)
- [Cause | 583](#)
- [Solution | 584](#)

## Problem

## Description

Unable to upgrade a chassis cluster using minimal downtime upgrade method.

## Environment

SRX5400 chassis cluster.

## Symptoms

- Cluster stuck in node0 RG1 with IF flag and cannot upgrade.
- Configuration commit error is shown on CLI.

## Cause

Configuration has same prefix on backup-router destinations (on backup RE/node) and user interface address.

```
regress@R1_re# show interfaces ge-0/0/0
```

```
unit 0 {
  family inet {
    address 192.1.1.1/24;
  }
}
```

```
regress@R1_re# show groups re1 system backup-router
```

```
10.204.63.254 destination 192.1.1.1/18;
```

```
regress@R1_re# commit
```

```
re0:
configuration check succeeds
re1:
error: Cannot have same prefix for backup-router destination and interface address. ge-0/0/0.0
inet 192.1.1
error: configuration check-out failed
```

```
re0:
error: remote commit-configuration failed on re1
```

## Solution

In chassis cluster mode, the backup router's destination address for IPv4 and IPv6 routers using the commands `edit system backup-router address destination destination-address` and `edit system inet6-backup-router address destination destination-address` must not be same as interface address configured for IPv4 and IPv6 using the commands `edit interfaces interface-name unit logical-unit-number family inet address ipv4-address` and `edit interfaces interface-name unit logical-unit-number family inet6 address ipv6-address`.

## Configuring backup-router Command on Chassis Cluster

### SUMMARY

How to back up a router in an SRX Series chassis cluster using the backup-router configuration command.

### IN THIS SECTION

- [Problem | 584](#)
- [Cause | 584](#)
- [Solution | 585](#)

## Problem

### Description

Intermittent connectivity issues to NSM and other management hosts from the secondary node.

### Environment

SRX Series chassis cluster

### Cause

Setting a destination of `0.0.0.0/0` on the backup router (without configuration) is not supported.

Example of an incorrect configuration:

```
set groups node0 system backup-router 10.10.10.1 destination 0.0.0.0/0
```

## Solution

See [Configuring a Backup Router](#) for the recommended way to set up a backup router by using a non-zero prefix.

Example of a non-zero subnet backup-router configuration:

```
set groups node0 system backup-router 10.10.10.1 destination 10.100.0.0/16
```

As an alternative to the 0/0 backup-router destination, here is another example where 0/0 gets split into two prefixes:

```
set groups node0 system backup-router 10.10.10.1 destination 0.0.0.0/1
set groups node0 system backup-router 10.10.10.1 destination 128.0.0.0/1
```

**NOTE:** If multiple networks need to be reachable through the backup router, you can add multiple destination entries to the configuration. The backup-router configuration is used only by the RGO secondary node. The primary node continues to use the inet.0 route table.

## Unable to Upgrade a Chassis Cluster Using In-Service Software Upgrade

### IN THIS SECTION

- [Problem | 586](#)
- [Cause | 586](#)
- [Solution | 587](#)

## Problem

## Description

Unable to upgrade a chassis cluster using minimal downtime upgrade method.

## Environment

SRX5400 chassis cluster.

## Symptoms

- Cluster stuck in node0 RG1 with IF flag and cannot upgrade.
- Configuration commit error is shown on CLI.

## Cause

Configuration has same prefix on backup-router destinations (on backup RE/node) and user interface address.

```
regress@R1_re# show interfaces ge-0/0/0
```

```
unit 0 {  
  family inet {  
    address 192.1.1.1/24;  
  }  
}
```

```
regress@R1_re# show groups re1 system backup-router
```

```
10.204.63.254 destination 192.1.1.1/18;
```

```
regress@R1_re# commit
```

```
re0:  
configuration check succeeds  
re1:  
error: Cannot have same prefix for backup-router destination and interface address. ge-0/0/0.0  
inet 192.1.1  
error: configuration check-out failed
```



```
re0:
error: remote commit-configuration failed on re1
```

## Solution

In chassis cluster mode, the backup router's destination address for IPv4 and IPv6 routers using the commands `edit system backup-router address destination destination-address` and `edit system inet6-backup-router address destination destination-address` must not be same as interface address configured for IPv4 and IPv6 using the commands `edit interfaces interface-name unit logical-unit-number family inet address ipv4-address` and `edit interfaces interface-name unit logical-unit-number family inet6 address ipv6-address`.

# Data Collection for Customer Support

Before you contact customer support, collect the data listed in [Table 40 on page 587](#).

**Table 40: Data Collection for Customer Support**

### Background information

---

- All SSH or Telnet session captures
  - Any available topology information
  - Summary of how the device is being used (such as production, lab system, colocation)
  - Summary of device history (new installation, production for x months or years, other recent cases)
  - Summary of any recent changes in the network or on the device
- 

### Request support information (RSI)

---

Collect the RSI:

```
request support information | save /var/tmp/rsi-CURRENT DATE.log
```

---

### Logs

---

Archive the contents of the **/var/log/** folder:

```
file archive compress source /var/log/* destination /var/tmp/logs-CURRENT-DATE.tgz
```

Ensure that the **/var/log/** directory is archived properly by verifying the file size using the file `list /var/tmp/logs-CURRENT-DATE.tgz detail` command.

---

**NOTE:** Upload all the logs to the JTAC support case. For instructions on how to upload the logs, see [KB23337](#).

# 7

CHAPTER

## Configuration Statements and Operational Commands

---

[Junos CLI Reference Overview | 590](#)

---

# Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- [Junos CLI Reference](#)

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- [Configuration Statements](#)
- [Operational Commands](#)



CHAPTER

Chassis Cluster Support on SRX100,  
SRX210, SRX220, SRX240,  
SRX550M, SRX650, SRX1400,  
SRX3400, and SRX3600 Devices

---

[Chassis Cluster Support on SRX100, SRX210, SRX220, SRX240, SRX550M,  
SRX650, SRX1400, SRX3400, and SRX3600 Devices | 592](#)

---

# Chassis Cluster Support on SRX100, SRX210, SRX220, SRX240, SRX550M, SRX650, SRX1400, SRX3400, and SRX3600 Devices

## IN THIS SECTION

- [Example: Configure IRB and VLAN with Members Across Two Nodes on a Security Device using Tagged Traffic | 608](#)
- [Example: Configure IRB and VLAN with Members Across Two Nodes on a Security Device using Untagged Traffic | 619](#)
- [Example: Configuring VLAN with Members Across Two Nodes on a Security Device | 629](#)

This topic includes the supported information for SRX100, SRX210, SRX220, SRX240, SRX550M, SRX650, SRX1400, SRX3400, and SRX3600 devices.

## SRX Series Chassis Cluster Configuration Overview

Following are the prerequisites for configuring a chassis cluster:

### Flow and Processing

Flowd monitoring is supported on SRX100, SRX210, SRX 220, SRX240, SRX550M, and SRX650 devices.

### Monitoring

The maximum number of monitoring IPs that can be configured per cluster is 64 for SRX550M devices. On SRX550M devices, logs cannot be sent to NSM when logging is configured in the stream mode.

## Installation and Upgrade

For SRX550M devices, the reboot parameter is not available, because the devices in a cluster are automatically rebooted following an in-band cluster upgrade (ICU).

ICU is available with the no-sync option only for SRX550M devices.

For SRX550M devices, the devices in a chassis cluster can be upgraded with a minimal service disruption of approximately 30 seconds using ICU with the no-sync option.

## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX100, SRX210, SRX220, SRX240, SRX550M, and SRX650.

Following are the prerequisites for configuring a chassis cluster:

- On SRX550M any existing configurations associated with interfaces that transform to the fxp0 management port and the control port should be removed.
- For SRX550M chassis clusters, the placement and type of GPIMs, XGPIMs, XPIMs, and Mini-PIMs (as applicable) must match in the two devices.

For SRX550M devices, control interfaces are dedicated Gigabit Ethernet ports.

Information about chassis cluster slot numbering is also provided in [Figure 51 on page 593](#), [Figure 52 on page 594](#), [Figure 53 on page 594](#), [Figure 54 on page 594](#), and [Figure 55 on page 594](#).

**Figure 51: Chassis Cluster Slot Numbering for SRX100 Devices**

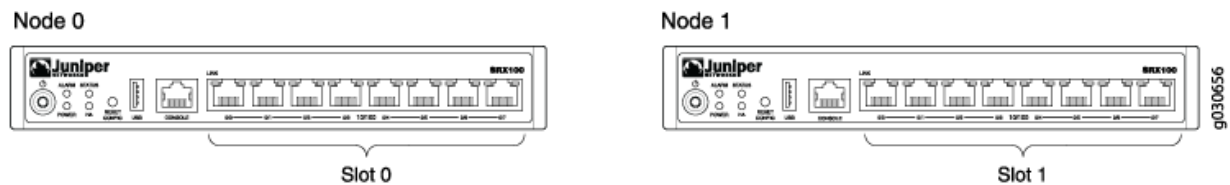


Figure 52: Chassis Cluster Slot Numbering for SRX210 Devices

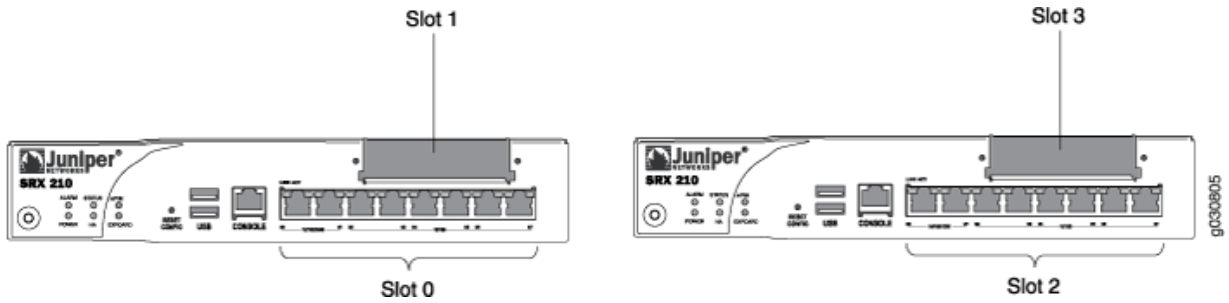


Figure 53: Chassis Cluster Slot Numbering for SRX220 Devices

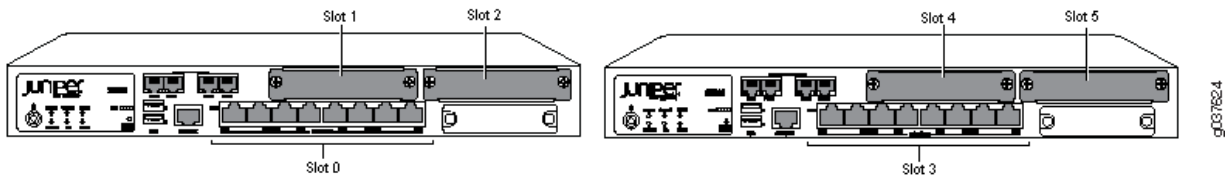


Figure 54: Chassis Cluster Slot Numbering for SRX240 Devices

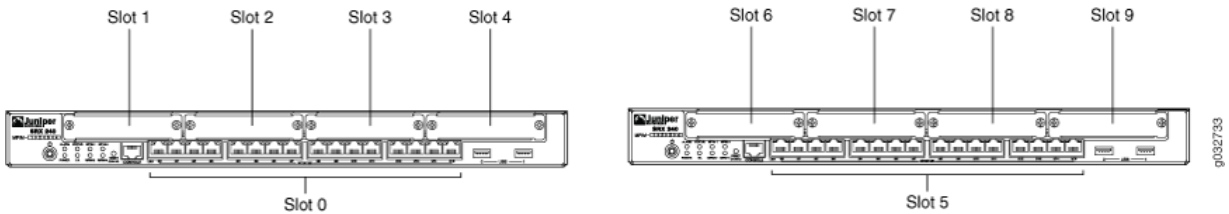
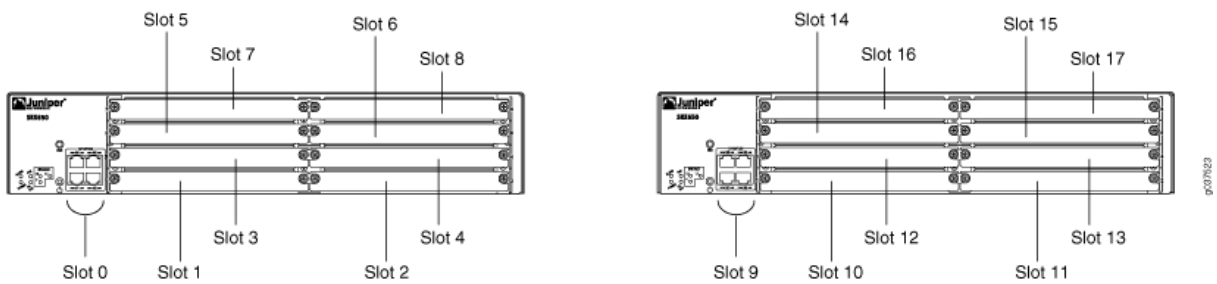


Figure 55: Chassis Cluster Slot Numbering for SRX650 Devices



Layer 2 switching must not be enabled on an SRX Series Firewall when chassis clustering is enabled. If you have enabled Layer 2 switching, make sure you disable it before enabling chassis clustering.



The factory default configuration for SRX100, SRX210, and SRX220 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, if you use the factory default configuration for these devices, you must delete the Ethernet switching configuration before you enable chassis clustering. See [Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering](#).

In chassis cluster mode, the interfaces on the secondary node are renumbered internally. For example, the management interface port on the front panel of each SRX210 device is still labeled fe-0/0/6, but internally, the node 1 port is referred to as fe-2/0/6.

For SRX650 devices, control interfaces are dedicated Gigabit Ethernet ports.

For SRX100, SRX220, and SRX210 devices, after you enable chassis clustering and reboot the system, the built-in interface named fe-0/0/6 is repurposed as the management interface and is automatically renamed fxp0.

For SRX550 devices, control interfaces are dedicated Gigabit Ethernet ports.

For SRX210 devices, after you enable chassis clustering and reboot the system, the built-in interface named fe-0/0/7 is repurposed as the control interface and is automatically renamed fxp1.

In chassis cluster mode, the interfaces on the secondary node are renumbered internally. For example, the management interface port on the front panel of each SRX210 device is still labeled fe-0/0/6, but internally, the node 1 port is referred to as fe-2/0/6.

For SRX240 devices, control interfaces are dedicated Gigabit Ethernet ports. For SRX100 and SRX220 devices, after you enable chassis clustering and reboot the system, the built-in interface named fe-0/0/7 is repurposed as the control interface and is automatically renamed fxp1.

**NOTE:** For SRX210 Services Gateways, the base and enhanced versions of a model can be used to form a cluster. For example:

- SRX210B and SRX210BE
- SRX210H and SRX210HE

However, the following combinations cannot be used to form a cluster:

- SRX210B and SRX210H
- SRX210B and SRX210HE
- SRX210BE and SRX210H
- SRX210BE and SRX210HE

Figure 56 on page 596, Figure 57 on page 596, Figure 58 on page 596, Figure 59 on page 597, Figure 60 on page 597, Figure 61 on page 597 and all show pairs of SRX Series Firewalls with the fabric links and control links connected.

Figure 56: Connecting SRX100 Devices in a Chassis Cluster

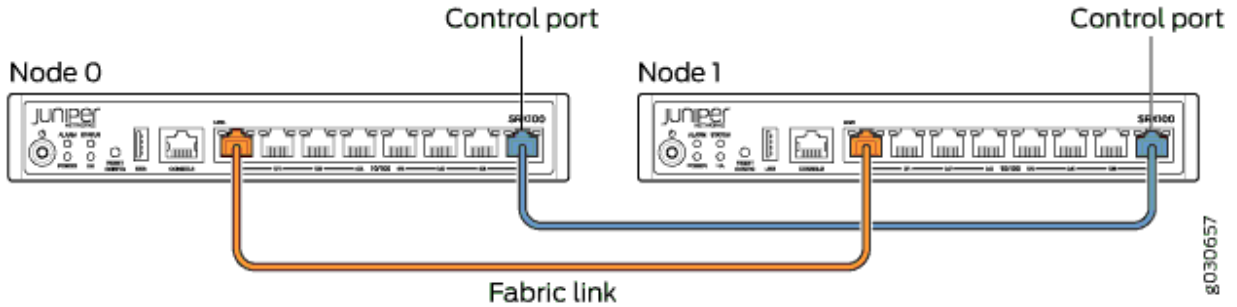


Figure 57: Connecting SRX110 Devices in a Chassis Cluster

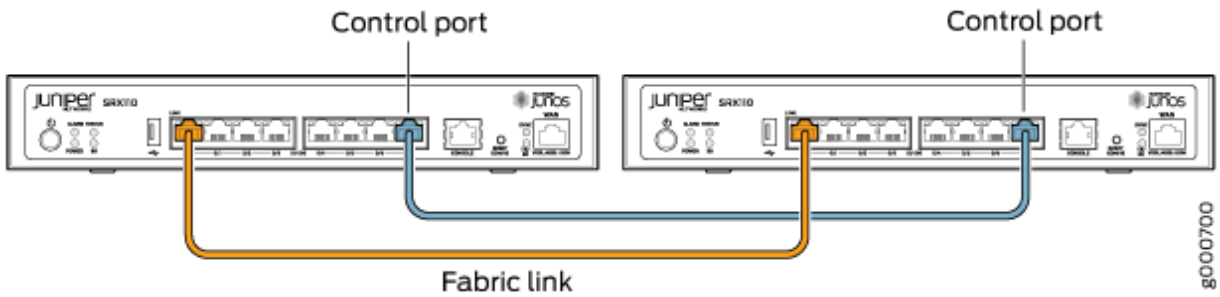


Figure 58: Connecting SRX210 Devices in a Chassis Cluster

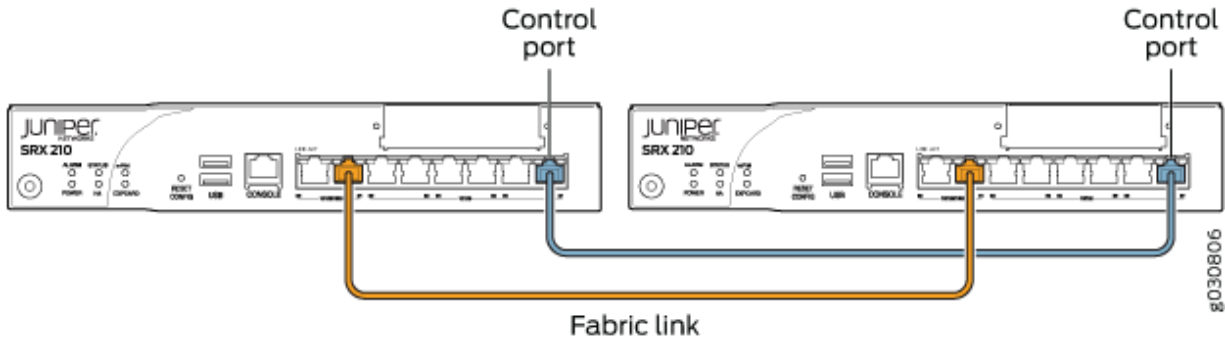


Figure 59: Connecting SRX220 Devices in a Chassis Cluster

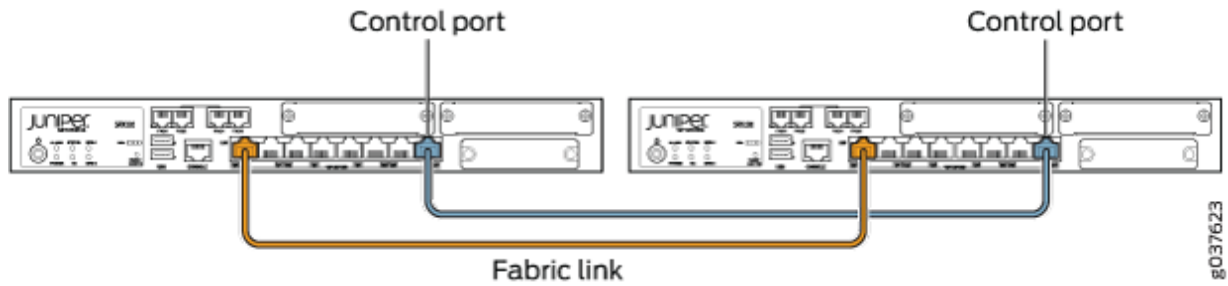


Figure 60: Connecting SRX240 Devices in a Chassis Cluster

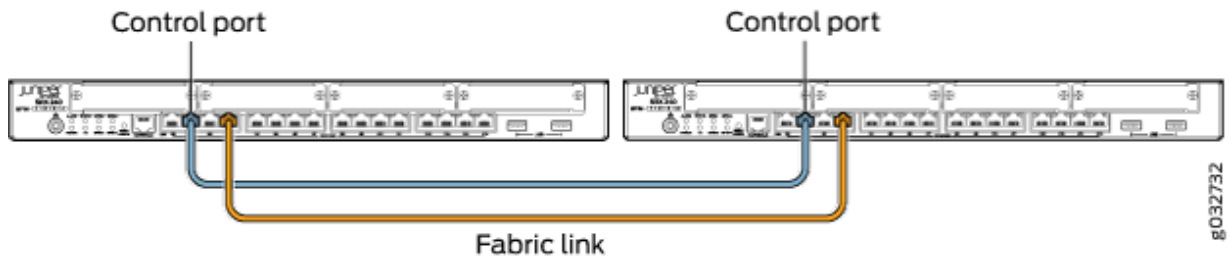
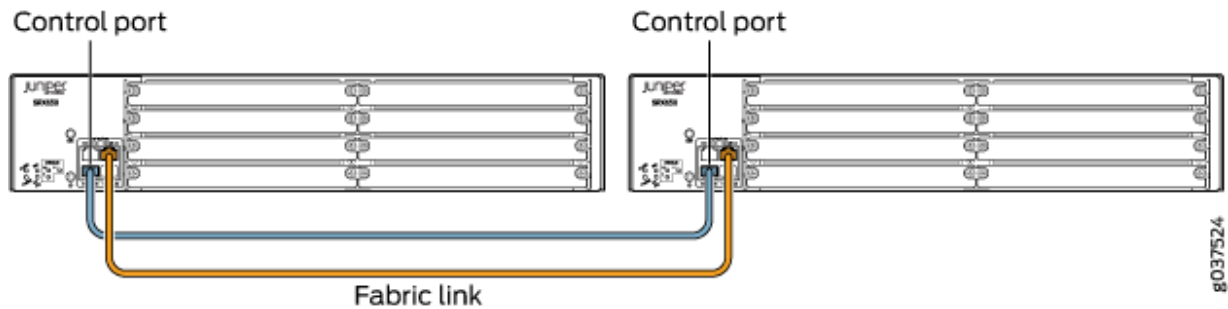


Figure 61: Connecting SRX650 Devices in a Chassis Cluster



The fabric link connection for the SRX100 and SRX210 must be a pair of either Fast Ethernet or Gigabit Ethernet interfaces. The fabric link connection must be any pair of either Gigabit Ethernet or 10-Gigabit Ethernet interfaces on all SRX Series Firewalls.

For some SRX Series Firewalls, such as the SRX100 and SRX200 line devices, do not have a dedicated port for fxp0. For SRX100, SRX210, the fxp0 interface is repurposed from a built-in interface.

**Table 41: SRX Devices Interface Renumbering**

SRX Series Services Gateway	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX550M	9	ge-0/0/0	ge-9/0/0

## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX3600, SRX3400, and SRX1400

Table 42 on page 598 shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series Firewalls that become node 0 and node 1 of the chassis cluster after the cluster is formed.

**Table 42: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX1400, SRX3400, and SRX3600**

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX550M	Node 0	9 (PIM slots)	0-8	ge-0/0/0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1		9–17	ge-9/0/0	ge-9/0/1	Any Ethernet port
				fxp0	fxp1	fab1
SRX3600	Node 0	13 (CFM slots)	0 – 12	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0

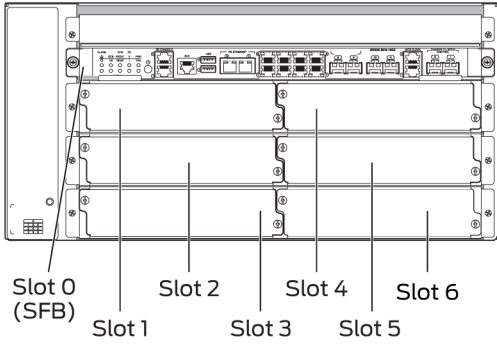
**Table 42: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX1400, SRX3400, and SRX3600 (Continued)**

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
	Node 1		13 – 25	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1
SRX3400	Node 0	8 (CFM slots)	0 – 7	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		8 – 15	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1
SRX1400	Node 0	4 (FPC slots)	0 – 3	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		4 – 7	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1

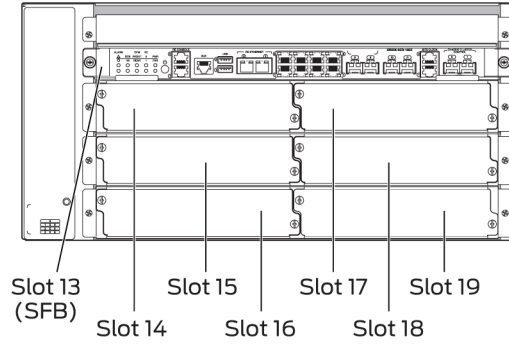
Information about chassis cluster slot numbering is also provided in [Figure 62 on page 600](#), [Figure 63 on page 601](#), [Figure 65 on page 602](#), and [Figure 64 on page 601](#).

**Figure 62: Chassis Cluster Slot Numbering for SRX3600 Devices**

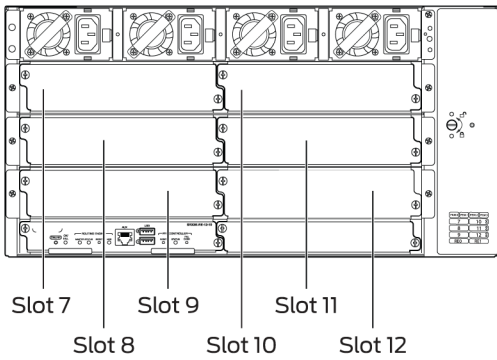
Node 0  
Front



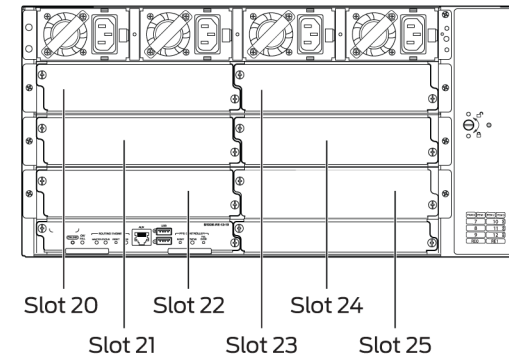
Node 1  
Front



Node 0  
Back



Node 1  
Back



8007344

Figure 63: Chassis Cluster Slot Numbering for SRX3400 Devices

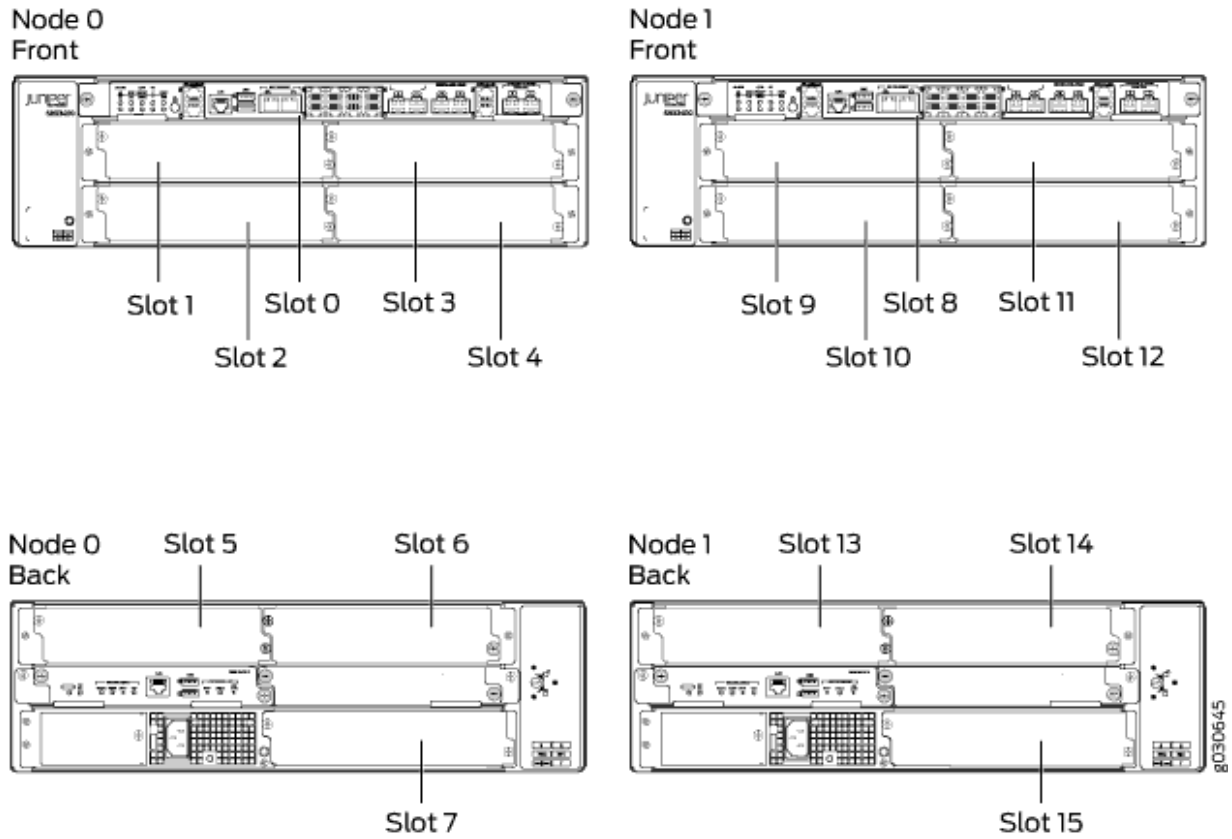
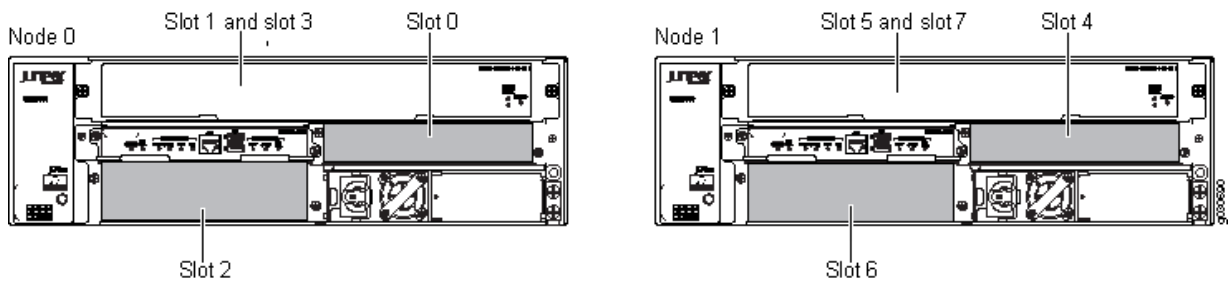
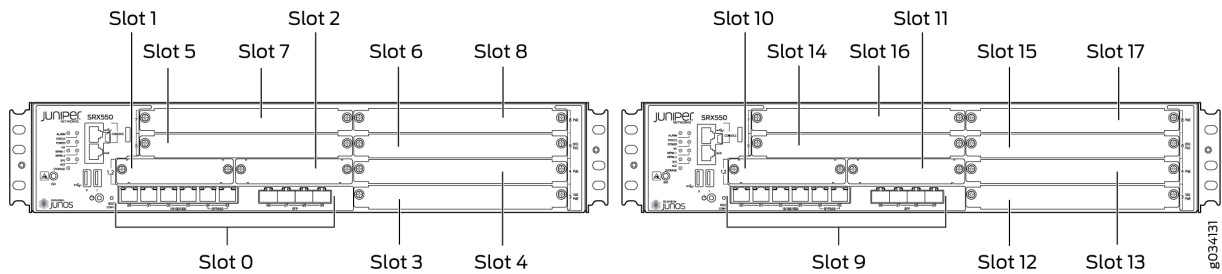


Figure 64: Chassis Cluster Slot Numbering for SRX1400 Devices



**Figure 65: Slot Numbering for SRX550M Devices**



In a large chassis cluster configuration on an SRX3400 or SRX3600 device, the heartbeat timers are recommended to increase the wait time to 8 seconds.

For SRX550M devices, connect the ge-0/0/1 on node 0 to the ge-9/0/1 on node 1.

You can connect two control links (SRX1400 and SRX3000 lines only) and two fabric links between the two devices in the cluster to reduce the chance of control link and fabric link failure. See ["Understanding Chassis Cluster Dual Control Links"](#) on page 163 and ["Understanding Chassis Cluster Dual Fabric Links"](#) on page 173.

[Figure 69 on page 604](#) show pairs of SRX Series Firewalls with the fabric links and control links connected.

[Figure 67 on page 603](#) and [Figure 68 on page 603](#) show pairs of SRX Series Firewalls with the fabric links and control links connected.

**Figure 66: Connecting SRX550M Devices in a Chassis Cluster**

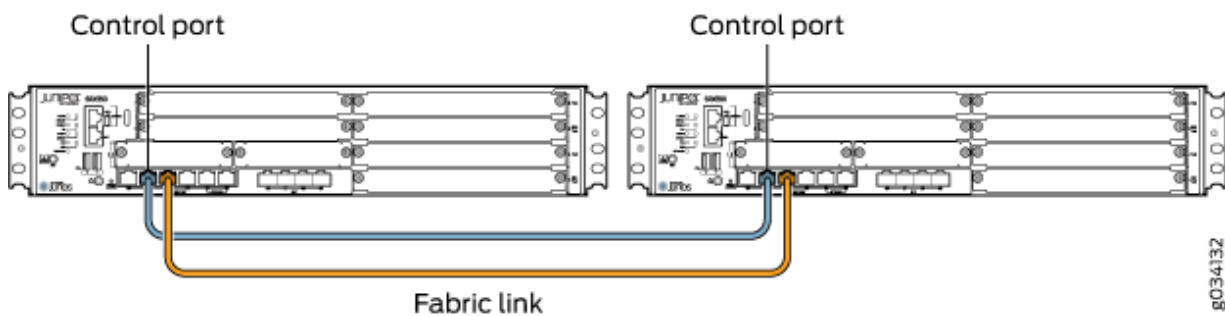




Figure 67: Connecting SRX3600 Devices in a Chassis Cluster

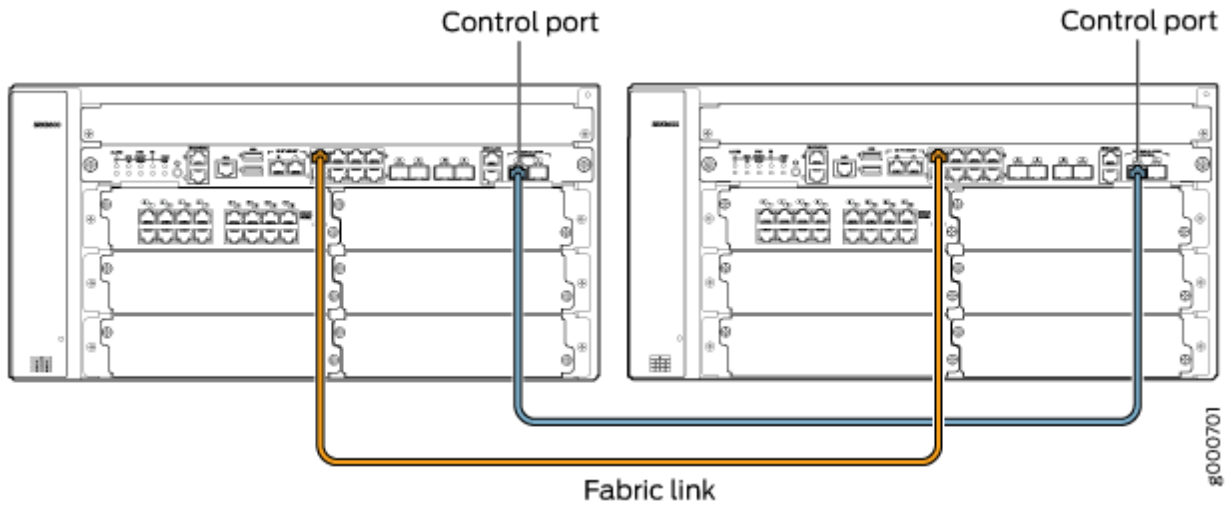
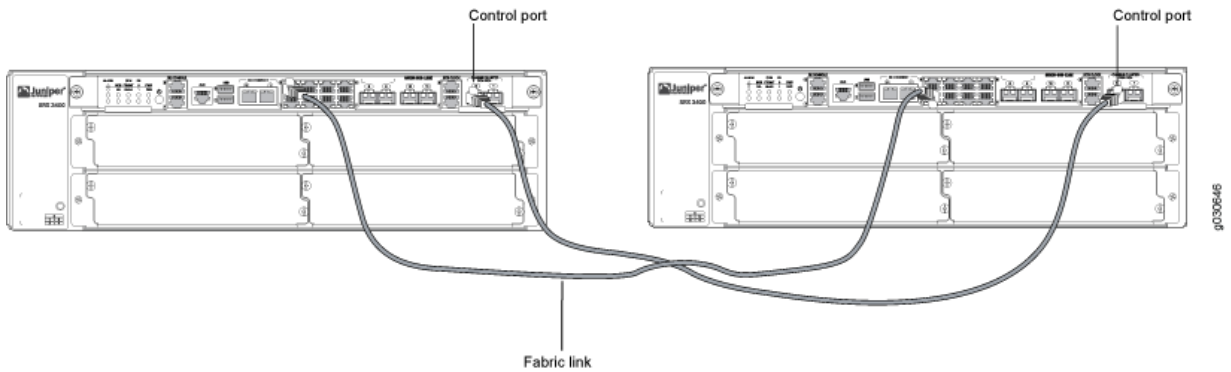
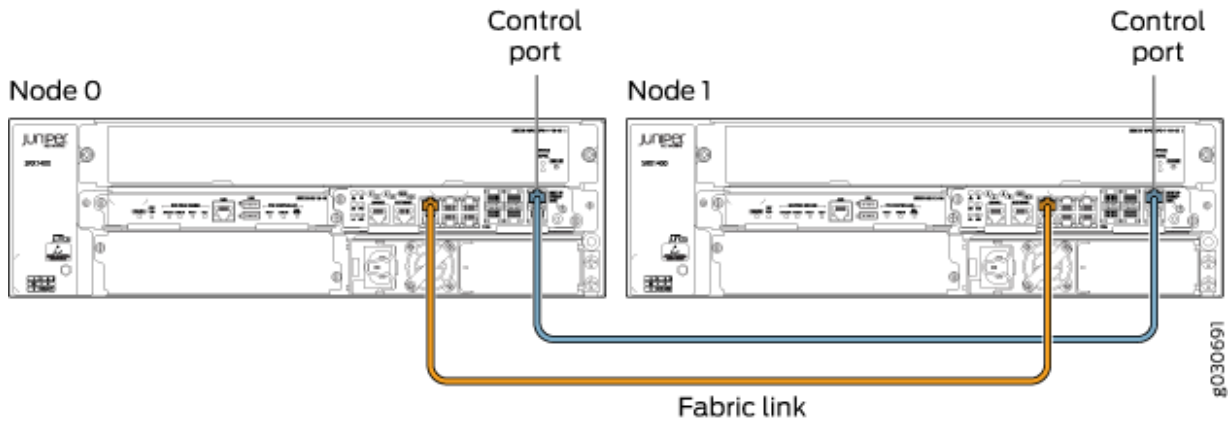


Figure 68: Connecting SRX3400 Devices in a Chassis Cluster



For dual control links on SRX3000 line devices, the Routing Engine must be in slot 0 and the SRX Clustering Module (SCM) in slot 1. The opposite configuration (SCM in slot 0 and Routing Engine in slot 1) is not supported.

Figure 69: Connecting SRX1400 Devices in a Chassis Cluster



## Supported Fabric Interface Types for SRX Series Firewalls (SRX210, SRX240, SRX220, SRX100, and SRX650 Devices)

For SRX210 devices, the fabric link can be any pair of Gigabit Ethernet interfaces or Fast Ethernet interfaces (as applicable). Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

For SRX550 devices, the fabric link can be any pair of Gigabit Ethernet interfaces or Fast Ethernet interfaces (as applicable).

For SRX Series chassis clusters made up of SRX550M devices, SFP interfaces on Mini-PIMs cannot be used as the fabric link.

For SRX550M devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a *chassis cluster* deployment is 1024.

For SRX Series chassis clusters, the fabric link can be any pair of Ethernet interfaces spanning the cluster; the fabric link can be any pair of Gigabit Ethernet interface.

[Table 43 on page 604](#) shows the fabric interface types that are supported for SRX Series Firewalls.

**Table 43: Supported Fabric Interface Types for SRX Series Firewalls**

SRX550	SRX650	SRX240	SRX220	SRX100	SRX210
Fast Ethernet	Fast Ethernet	Fast Ethernet		Fast Ethernet	Fast Ethernet

Table 43: Supported Fabric Interface Types for SRX Series Firewalls (*Continued*)

SRX550	SRX650	SRX240	SRX220	SRX100	SRX210
Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet		Gigabit Ethernet

## Redundant Ethernet Interfaces

Table 44: Maximum Number of Redundant Ethernet Interfaces Allowed (SRX100, SRX220, SRX240, SRX210, and SRX650)

Device	Maximum Number of reth Interfaces
SRX100	8
SRX210	8
SRX220	8
SRX240	24
SRX550M	58
SRX650	68

- Point-to-Point Protocol over Ethernet (PPPoE) over redundant Ethernet (reth) interface is supported on SRX100, SRX210, SRX220, SRX240, SRX550M, and SRX650 devices in chassis cluster mode. This feature allows an existing PPPoE session to continue without starting a new PPPoE session in the event of a failover.
- On SRX550M devices, the number of child interfaces is restricted to 16 on the reth interface (eight per node).

For SRX100, SRX220, and SRX240 devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a *chassis cluster* deployment is 1024.

On SRX550M devices, the speed mode and link mode configuration is available for member interfaces of a reth interface.

IP address monitoring cannot be used on a chassis cluster running in transparent mode. The maximum number of monitoring IP addresses that can be configured per cluster is 32 for the SRX1400 device and the SRX3000 line of devices.

## Control Links

- For SRX100, SRX210, and SRX220 devices, the control link uses the fe-0/0/7 interface.
- For SRX210 devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a chassis cluster deployment is 1024.
- For SRX240, SRX650M, devices, the control link uses the ge-0/0/1 interface.

**Table 45: fxp0 and fxp1 Ports on SRX550M Series Devices**

Device	Management (fxp0)	HA Control (fxp1)	Fabric (fab0 and fab1)—must be configured
SRX550M	ge-0/0/0	ge-0/0/1	Any ge or xe interface

**Table 46: SRX Series Firewalls Interface Settings (SRX100, SRX210, SRX220, SRX240, SRX550M)**

Command	SRX100	SRX210	SRX220	SRX240	SRX550M
set interfaces fab0 fabric-options member-interfaces	fe-0/0/1	ge-0/0/1	ge-0/0/0 to ge-0/0/5	ge-0/0/2	ge-0/0/2
set interfaces fab1 fabric-options member-interfaces	fe-1/0/1	ge-2/0/1	ge-3/0/0 to ge-3/0/5	ge-5/0/2	ge-9/0/2

**Table 46: SRX Series Firewalls Interface Settings (SRX100, SRX210, SRX220, SRX240, SRX550M)**  
*(Continued)*

Command	SRX100	SRX210	SRX220	SRX240	SRX550M
set chassis cluster redundancy-group 1 interface-monitor	fe-0/0/0 weight 255	fe-0/0/3 weight 255	ge-0/0/0 weight 255	ge-0/0/5 weight 255	ge-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-0/0/2 weight 255	fe-0/0/2 weight 255	ge-3/0/0 weight 255	ge-5/0/5 weight 255	ge-10/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-1/0/0 weight 255	fe-2/0/3 weight 255	ge-0/0/1 weight 255	ge-0/0/6 weight 255	ge-1/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-1/0/2 weight 255	fe-2/0/2 weight 255	ge-3/0/1 weight 255	ge-5/0/6 weight 255	ge-10/0/1 weight 255
set interfaces	fe-0/0/2 fastether- options redundant- parent reth1	fe-0/0/2 fastether- options redundant- parent reth1	ge-0/0/2 fastether- options redundant- parent reth0	ge-0/0/5 gigether- options redundant- parent reth1	ge-1/0/0 gigether- options redundant- parent reth1
set interfaces	fe-1/0/2 fastether- options redundant- parent reth1	fe-2/0/2 fastether- options redundant- parent reth1	ge-0/0/3 fastether- options redundant- parent reth1	ge-5/0/5 gigether- options redundant- parent reth1	ge-10/0/0 gigether- options redundant- parent reth1
set interfaces	fe-0/0/0 fastether- options redundant- parent reth0	fe-0/0/3 fastether- options redundant- parent reth0	ge-3/0/2 fastether- options redundant- parent reth0	ge-0/0/6 gigether- options redundant- parent reth0	ge-1/0/1 gigether- options redundant- parent reth0

**Table 46: SRX Series Firewalls Interface Settings (SRX100, SRX210, SRX220, SRX240, SRX550M)**  
(Continued)

Command	SRX100	SRX210	SRX220	SRX240	SRX550M
set interfaces	fe-1/0/0 fastether- options redundant- parent reth0	fe-2/0/3 fastether- options redundant- parent reth0	ge-3/0/3 fastether- options redundant- parent reth1	ge-5/0/6 gigether- options redundant- parent reth0	ge-10/0/1 gigether- options redundant- parent reth0

## ISSU System Requirements for SRX1400, SRX3400 and SRX3600

To perform an ISSU, your device must be running a Junos OS release that supports ISSU for the specific platform. See [Table 47 on page 608](#) for platform support.

**Table 47: ISSU Platform Support SRX1400, SRX3400 and SRX3600**

Device	Junos OS Release
SRX1400	12.1X47-D10
SRX3400	12.1X47-D10
SRX3600	12.1X47-D10

## Example: Configure IRB and VLAN with Members Across Two Nodes on a Security Device using Tagged

### IN THIS SECTION

● Requirements | 609

- Overview | 609
- Configuration | 610

**NOTE:** Our content testing team has validated and updated this example.

## Requirements

This example uses the following hardware and software components:

- configure a switching fabric interface on both nodes to configure Ethernet switching-related features on the nodes. See ["Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device" on page 479](#)
- SRX550 security device
- interface-mode is supported in 15.1X49 release.
- port-mode is supported in 12.1 and 12.3X48 releases.

## Overview

### IN THIS SECTION

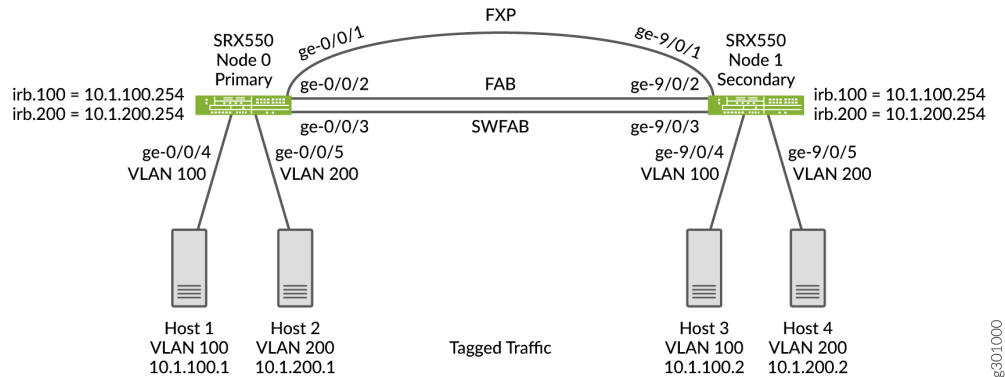
- Topology | 609

This example shows the configuration of a VLAN with members across node 0 and node 1.

## Topology

[Figure 70 on page 610](#) shows the Layer 2 Ethernet switching across chassis cluster nodes using tagged traffic.

Figure 70: Layer 2 Ethernet Switching Across Chassis Cluster using Tagged Traffic



## Configuration

### IN THIS SECTION

- [Verification | 615](#)

## Procedure

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces irb.100
set security zones security-zone trust interfaces irb.200
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members v100
set interfaces ge-0/0/5 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/5 unit 0 family ethernet-switching vlan members v200
set interfaces ge-9/0/4 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-9/0/4 unit 0 family ethernet-switching vlan members v100
set interfaces ge-9/0/5 unit 0 family ethernet-switching interface-mode trunk
```



```

set interfaces ge-9/0/5 unit 0 family ethernet-switching vlan members v200
set interfaces fab0 fabric-options member-interfaces ge-0/0/2
set interfaces fab1 fabric-options member-interfaces ge-9/0/2
set interfaces irb unit 100 family inet address 10.1.100.254/24
set interfaces irb unit 200 family inet address 10.1.200.254/24
set interfaces swfab0 fabric-options member-interfaces ge-0/0/3
set interfaces swfab1 fabric-options member-interfaces ge-9/0/3
set vlans v100 vlan-id 100
set vlans v100 l3-interface irb.100
set vlans v200 vlan-id 200
set vlans v200 l3-interface irb.200

```

## Step-by-Step Procedure

To configure IRB and a VLAN:

1. Configure security zones.

```

{primary:node0} [edit security zones]
user@host# set security-zone trust host-inbound-traffic system-services all
user@host# set security-zone trust interfaces irb.100
user@host# set security-zone trust interfaces irb.200

```

2. Configure Ethernet switching on the node0 interfaces.

```

{primary:node0} [edit interfaces]
user@host# set ge-0/0/4 unit 0 family ethernet-switching interface-mode trunk
user@host# set ge-0/0/4 unit 0 family ethernet-switching vlan members v100
user@host# set ge-0/0/5 unit 0 family ethernet-switching interface-mode trunk
user@host# set ge-0/0/5 unit 0 family ethernet-switching vlan members v200
user@host# set ge-9/0/4 unit 0 family ethernet-switching interface-mode trunk
user@host# set ge-9/0/4 unit 0 family ethernet-switching vlan members v100
user@host# set ge-9/0/5 unit 0 family ethernet-switching interface-mode trunk
user@host# set ge-9/0/5 unit 0 family ethernet-switching vlan members v200

```

3. Define the interfaces used for the fab connection (data plane links for RTOsync) by using physical ports from each node. These interfaces must be connected back-to-back, or through a Layer 2 infrastructure.

```
{primary:node0} [edit interfaces]
user@host# set fab0 fabric-options member-interfaces ge-0/0/2
user@host# set fab1 fabric-options member-interfaces ge-9/0/2
```

4. configure a switching fabric interface on both nodes to configure Ethernet switching-related features on the nodes.

```
{primary:node0} [edit interfaces]
user@host# set swfab0 fabric-options member-interfaces ge-0/0/3
user@host# set swfab1 fabric-options member-interfaces ge-9/0/3
```

5. Configure the irb interface.

```
{primary:node0} [edit interfaces]
user@host# set irb unit 100 family inet address 10.1.100.254/24
user@host# set irb unit 200 family inet address 10.1.200.254/24
```

6. Create and associate a VLAN interface with the VLAN.

```
{primary:node0} [edit vlans]
user@host# set v100 vlan-id 100
user@host# set v100 l3-interface irb.100
user@host# set v200 vlan-id 200
user@host# set v200 l3-interface irb.200
```

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

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

## Results

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

```
[edit]
user@host# show security
zones {
  security-zone trust {
    host-inbound-traffic {
      system-services {
        all;
      }
    }
    interfaces {
      irb.100;
      irb.200;
    }
  }
}
```

```
[edit]
user@host# show interfaces
ge-0/0/4 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members v100;
      }
    }
  }
}
ge-0/0/5 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members v200;
      }
    }
  }
}
```

```
    }
  }
}
ge-9/0/4 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members v100;
      }
    }
  }
}
ge-9/0/5 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members v200;
      }
    }
  }
}
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/2;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-9/0/2;
    }
  }
}
irb {
  unit 100 {
    family inet {
      address 10.1.100.254/24;
    }
  }
}
```

```
unit 200 {
    family inet {
        address 10.1.200.254/24;
    }
}
swfab0 {
    fabric-options {
        member-interfaces {
            ge-0/0/3;
        }
    }
}
swfab1 {
    fabric-options {
        member-interfaces {
            ge-9/0/3;
        }
    }
}
```

```
[edit]
user@host# show vlans
v100 {
    vlan-id 100;
    l3-interface irb.100;
}
v200 {
    vlan-id 200;
    l3-interface irb.200;
}
```

## Verification

### IN THIS SECTION

- [Verifying Tagged VLAN With IRB | 616](#)

## Verifying Tagged VLAN With IRB

### Purpose

Verify that the configuration for tagged VLAN with IRB is working properly.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```
user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      fxp1      Up                Disabled     Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
                    (Physical/Monitored)
  fab0    ge-0/0/2        Up / Up        Enabled
  fab0
  fab1    ge-9/0/2        Up / Up        Enabled
  fab1

Redundant-pseudo-interface Information:
  Name      Status  Redundancy-group
  lo0       Up      0
```

From operational mode, enter the `show ethernet-switching table` command.

```
user@host> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
          SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)

Ethernet switching table : 4 entries, 4 learned
Routing instance : default-switch
```

Vlan	MAC	MAC	Age	Logical	NH
RTR					
name	address	flags		interface	Index
ID					
v100	08:81:f4:8a:eb:52	D	-	ge-9/0/4.0	0
0					
v100	08:81:f4:8a:eb:54	D	-	ge-0/0/4.0	0
0					
v200	08:81:f4:8a:eb:53	D	-	ge-9/0/5.0	0
0					
v200	08:81:f4:8a:eb:55	D	-	ge-0/0/5.0	0 0

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

```
user@host> show arp
```

MAC Address	Address	Name	Interface	Flags
08:81:f4:8a:eb:54	10.1.100.1	10.1.100.1	irb.100	none
08:81:f4:8a:eb:52	10.1.100.2	10.1.100.2	irb.100	none
08:81:f4:8a:eb:55	10.1.200.1	10.1.200.1	irb.200	none
08:81:f4:8a:eb:53	10.1.200.2	10.1.200.2	irb.200	none
ec:3e:f7:c6:81:b0	30.17.0.2	30.17.0.2	fab0.0	permanent
f0:4b:3a:09:cb:30	30.18.0.1	30.18.0.1	fab1.0	permanent
ec:3e:f7:c6:80:81	130.16.0.1	130.16.0.1	fxp1.0	none

Total entries: 7

From operational mode, enter the `show ethernet-switching interface` command to view the information about Ethernet switching interfaces.

```
user@host> show ethernet-switching interface
```

Routing Instance Name : default-switch

Logical Interface flags (DL - disable learning, AD - packet action drop,  
LH - MAC limit hit, DN - interface down,  
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,  
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical interface	Vlan members	TAG	MAC limit	MAC+IP limit	STP state	Logical interface flags	Tagging
ge-0/0/5.0			16383	8192			tagged
	v200	200	1024	1024	Forwarding		tagged

Routing Instance Name : default-switch

Logical Interface flags (DL - disable learning, AD - packet action drop,

```

LH - MAC limit hit, DN - interface down,
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical      Vlan      TAG  MAC  MAC+IP STP      Logical      Tagging
interface    members                                     interface flags
ge-0/0/4.0                                     16383 8192                                     tagged
          v100          100  1024  1024  Forwarding                                     tagged
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
LH - MAC limit hit, DN - interface down,
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical      Vlan      TAG  MAC  MAC+IP STP      Logical      Tagging
interface    members                                     interface flags
ge-9/0/4.0                                     16383 8192                                     tagged
          v100          100  1024  1024  Forwarding                                     tagged
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
LH - MAC limit hit, DN - interface down,
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical      Vlan      TAG  MAC  MAC+IP STP      Logical      Tagging
interface    members                                     interface flags
ge-9/0/5.0                                     16383 8192                                     tagged
          v200          200  1024  1024  Forwarding                                     tagged

```

## Meaning

The output shows the VLANs are configured and working fine.



## Example: Configure IRB and VLAN with Members Across Two Nodes on a Security Device using Untagged Traffic

### IN THIS SECTION

- [Requirements | 619](#)
- [Overview | 619](#)
- [Configuration | 620](#)

**NOTE:** Our content testing team has validated and updated this example.

### Requirements

This example uses the following hardware and software components:

- configure a switching fabric interface on both nodes to configure Ethernet switching-related features on the nodes. See ["Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device"](#) on page 479
- SRX550 security device
- interface-mode is supported in 15.1X49 release.
- port-mode is supported in 12.1 and 12.3X48 releases.

### Overview

#### IN THIS SECTION

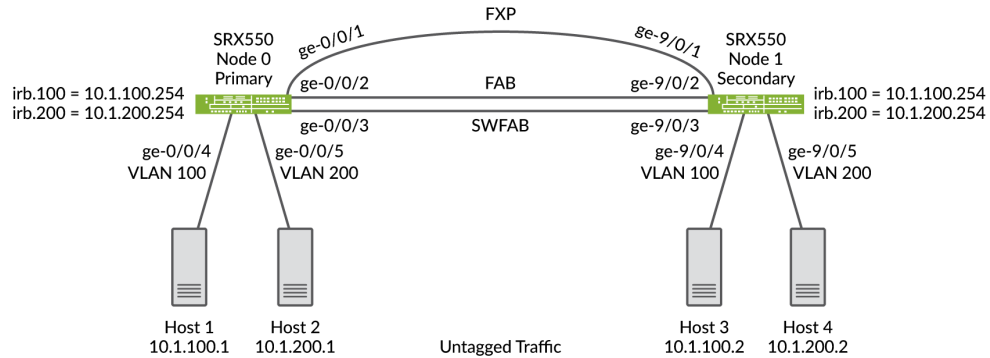
- [Topology | 620](#)

This example shows the configuration of a VLAN with members across node 0 and node 1.

## Topology

Figure 71 on page 620 shows the Layer 2 Ethernet switching across chassis cluster nodes using untagged traffic.

Figure 71: Layer2 Ethernet Switching Across Chassis Cluster Nodes using Untagged Traffic



## Configuration

### IN THIS SECTION

- [Verification | 625](#)

## Procedure

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces irb.100
set security zones security-zone trust interfaces irb.200
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode access
```

```

set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members v100
set interfaces ge-0/0/5 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/5 unit 0 family ethernet-switching vlan members v200
set interfaces ge-9/0/4 unit 0 family ethernet-switching interface-mode access
set interfaces ge-9/0/4 unit 0 family ethernet-switching vlan members v100
set interfaces ge-9/0/5 unit 0 family ethernet-switching interface-mode access
set interfaces ge-9/0/5 unit 0 family ethernet-switching vlan members v200
set interfaces fab0 fabric-options member-interfaces ge-0/0/2
set interfaces fab1 fabric-options member-interfaces ge-9/0/2
set interfaces irb unit 100 family inet address 10.1.100.254/24
set interfaces irb unit 200 family inet address 10.1.200.254/24
set interfaces swfab0 fabric-options member-interfaces ge-0/0/3
set interfaces swfab1 fabric-options member-interfaces ge-9/0/3
set vlans v100 vlan-id 100
set vlans v100 l3-interface irb.100
set vlans v200 vlan-id 200
set vlans v200 l3-interface irb.200

```

## Step-by-Step Procedure

To configure IRB and a VLAN:

1. Configure security zones.

```

{primary:node0} [edit security zones]
user@host# set security-zone trust host-inbound-traffic system-services all
user@host# set security-zone trust interfaces irb.100
user@host# set security-zone trust interfaces irb.200

```

2. Configure Ethernet switching on the node0 interfaces.

```

{primary:node0} [edit interfaces]
user@host# set ge-0/0/4 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/4 unit 0 family ethernet-switching vlan members v100
user@host# set ge-0/0/5 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/5 unit 0 family ethernet-switching vlan members v200
user@host# set ge-9/0/4 unit 0 family ethernet-switching interface-mode access
user@host# set ge-9/0/4 unit 0 family ethernet-switching vlan members v100
user@host# set ge-9/0/5 unit 0 family ethernet-switching interface-mode access
user@host# set ge-9/0/5 unit 0 family ethernet-switching vlan members v200

```

3. Define the interfaces used for the fab connections (data plane links for RTOsync) by using physical ports from each node. These interfaces must be connected back-to-back, or through a Layer 2 infrastructure.

```
{primary:node0} [edit interfaces]
user@host# set fab0 fabric-options member-interfaces ge-0/0/2
user@host# set fab1 fabric-options member-interfaces ge-9/0/2
```

4. configure a switching fabric interface on both nodes to configure Ethernet switching-related features on the nodes.

```
{primary:node0} [edit interfaces]
user@host# set swfab0 fabric-options member-interfaces ge-0/0/3
user@host# set swfab1 fabric-options member-interfaces ge-9/0/3
```

5. Configure the irb interface.

```
{primary:node0} [edit interfaces]
user@host# set irb unit 100 family inet address 10.1.100.254/24
user@host# set irb unit 200 family inet address 10.1.200.254/24
```

6. Create and associate a VLAN interface with the VLAN.

```
{primary:node0} [edit vlans]
user@host# set v100 vlan-id 100
user@host# set v100 l3-interface irb.100
user@host# set v200 vlan-id 200
user@host# set v200 l3-interface irb.200
```

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

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

## Results

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

```
[edit]
user@host# show security
zones {
  security-zone trust {
    host-inbound-traffic {
      system-services {
        all;
      }
    }
    interfaces {
      irb.100;
      irb.200;
    }
  }
}
```

```
[edit]
user@host# show interfaces
ge-0/0/4 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members v100;
      }
    }
  }
}
ge-0/0/5 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members v200;
      }
    }
  }
}
```

```
    }
  }
}
ge-9/0/4 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members v100;
      }
    }
  }
}
ge-9/0/5 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members v200;
      }
    }
  }
}
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/2;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-9/0/2;
    }
  }
}
irb {
  unit 100 {
    family inet {
      address 10.1.100.254/24;
    }
  }
}
```

```
unit 200 {
    family inet {
        address 10.1.200.254/24;
    }
}
swfab0 {
    fabric-options {
        member-interfaces {
            ge-0/0/3;
        }
    }
}
swfab1 {
    fabric-options {
        member-interfaces {
            ge-9/0/3;
        }
    }
}
```

```
[edit]
user@host# show vlans
v100 {
    vlan-id 100;
    l3-interface irb.100;
}
v200 {
    vlan-id 200;
    l3-interface irb.200;
}
```

## Verification

### IN THIS SECTION

- [Verifying Untagged VLAN With IRB | 626](#)

## Verifying Untagged VLAN With IRB

### Purpose

Verify that the configuration of untagged VLAN with IRB is working properly.

### Action

From operational mode, enter the `show chassis cluster interfaces` command.

```

user@host> show chassis cluster interfaces
Control link status: Up

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA  Security
  0      fxp1      Up                Disabled     Disabled

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status                Security
          (Physical/Monitored)
  fab0    ge-0/0/2        Up / Up              Enabled
  fab0
  fab1    ge-9/0/2        Up / Up              Enabled
  fab1

Redundant-pseudo-interface Information:
  Name    Status  Redundancy-group
  lo0     Up      0

```

From operational mode, enter the `show ethernet-switching table` command.

```

user@host> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
          SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)

Ethernet switching table : 4 entries, 4 learned
Routing instance : default-switch

```



Vlan	MAC	MAC	Age	Logical	NH
RTR					
name	address	flags		interface	Index
ID					
v100	08:81:f4:8a:eb:52	D	-	ge-9/0/4.0	0
0					
v100	08:81:f4:8a:eb:54	D	-	ge-0/0/4.0	0
0					
v200	08:81:f4:8a:eb:53	D	-	ge-9/0/5.0	0
0					
v200	08:81:f4:8a:eb:55	D	-	ge-0/0/5.0	0
					0

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

```
user@host> show arp
```

MAC Address	Address	Name	Interface	Flags
08:81:f4:8a:eb:54	10.1.100.1	10.1.100.1	irb.100	none
08:81:f4:8a:eb:52	10.1.100.2	10.1.100.2	irb.100	none
08:81:f4:8a:eb:55	10.1.200.1	10.1.200.1	irb.200	none
08:81:f4:8a:eb:53	10.1.200.2	10.1.200.2	irb.200	none
ec:3e:f7:c6:81:b0	30.17.0.2	30.17.0.2	fab0.0	permanent
f0:4b:3a:09:cb:30	30.18.0.1	30.18.0.1	fab1.0	permanent
ec:3e:f7:c6:80:81	130.16.0.1	130.16.0.1	fxp1.0	none

Total entries: 7

From operational mode, enter the `show ethernet-switching interface` command to view the information about Ethernet switching interfaces.

```
user@host> show ethernet-switching interface
```

Routing Instance Name : default-switch

Logical Interface flags (DL - disable learning, AD - packet action drop,  
LH - MAC limit hit, DN - interface down,  
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,  
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical interface	Vlan members	TAG	MAC limit	MAC+IP limit	STP state	Logical interface flags	Tagging
ge-0/0/5.0			16383	8192			untagged
	v200	200	1024	1024	Forwarding		untagged

Routing Instance Name : default-switch

Logical Interface flags (DL - disable learning, AD - packet action drop,

```

LH - MAC limit hit, DN - interface down,
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical      Vlan      TAG  MAC  MAC+IP STP      Logical      Tagging
interface    members                                     limit limit state  interface flags
ge-0/0/4.0                                     16383 8192                                     untagged
          v100          100  1024  1024  Forwarding                                     untagged
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
LH - MAC limit hit, DN - interface down,
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical      Vlan      TAG  MAC  MAC+IP STP      Logical      Tagging
interface    members                                     limit limit state  interface flags
ge-9/0/4.0                                     16383 8192                                     untagged
          v100          100  1024  1024  Forwarding                                     untagged
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
LH - MAC limit hit, DN - interface down,
MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,
SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical      Vlan      TAG  MAC  MAC+IP STP      Logical      Tagging
interface    members                                     limit limit state  interface flags
ge-9/0/5.0                                     16383 8192                                     untagged
          v200          200  1024  1024  Forwarding                                     untagged

```

## Meaning

The output shows the VLANs are configured and working fine.

## Example: Configuring VLAN with Members Across Two Nodes on a Security Device

### IN THIS SECTION

- [Requirements | 629](#)
- [Overview | 629](#)
- [Configuration | 629](#)
- [Verification | 632](#)

### Requirements

This example uses the following hardware and software components:

- configure a switching fabric interface on both nodes to configure Ethernet switching-related features on the nodes. See ["Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device" on page 479](#)
- SRX240 security device
- Junos OS 12.3X48-D90
- interface-mode is supported in 15.1X49 release.
- port-mode is supported in 12.1 and 12.3X48 releases.

### Overview

This example shows the configuration of a VLAN with members across node 0 and node 1.

### Configuration

### IN THIS SECTION

- [Procedure | 630](#)

## Procedure

### 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, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/3 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge0/0/4 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-7/0/5 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members vlan100
set interfaces vlan unit 100 family inet address 11.1.1.1/24
set vlans vlan100 vlan-id 100
set vlans vlan100 l3-interface vlan.100
```

### Step-by-Step Procedure

To configure VLAN:

1. Configure Ethernet switching on the node0 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching port-mode access
user@host# set interfaces ge0/0/4 unit 0 family ethernet-switching port-mode access
```

2. Configure Ethernet switching on the node1 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching port-mode trunk
```

3. Create VLAN vlan100 with vlan-id 100.

```
{primary:node0} [edit]
user@host# set vlans vlan100 vlan-id 100
```

4. Add interfaces from both nodes to the VLAN.

```
{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members vlan100
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members vlan100
```

5. Create a VLAN interface.

```
user@host# set interfaces vlan unit 100 family inet address 11.1.1.1/24
```

6. Associate an VLAN interface with the VLAN.

```
user@host# set vlans vlan100 l3-interface vlan.100
```

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

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

## Results

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

```
[edit]
user@host# show vlans
vlan100 {
  vlan-id 100;
  l3-interface vlan.100;
}
[edit]
user@host# show interfaces
ge-0/0/3 {
  unit 0 {
    family ethernet-switching {
      port-mode access;
```

```
        vlan {
            members vlan100;
        }
    }
}
ge-0/0/4 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan {
                members vlan100;
            }
        }
    }
}
ge-7/0/5 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members vlan100;
            }
        }
    }
}
```

## Verification

### IN THIS SECTION

- [Verifying VLAN | 632](#)

### Verifying VLAN

#### Purpose

Verify that the configuration of VLAN is working properly.

## Action

From operational mode, enter the `show interfaces terse ge-0/0/3` command to view the node 0 interface.

```
user@host> show interfaces terse ge-0/0/3
Interface          Admin Link Proto  Local          Remote
ge-0/0/3           up    up
ge-0/0/3.0        up    up  eth-switch
```

From operational mode, enter the `show interfaces terse ge-0/0/4` command to view the node 0 interface.

```
user@host> show interfaces terse ge-0/0/4
Interface          Admin Link Proto  Local          Remote
ge-0/0/4           up    up
ge-0/0/4.0        up    up  eth-switch
```

From operational mode, enter the `show interfaces terse ge-7/0/5` command to view the node1 interface.

```
user@host> show interfaces terse ge-7/0/5
Interface          Admin Link Proto  Local          Remote
ge-7/0/5           up    up
ge-7/0/5.0        up    up  eth-switch
```

From operational mode, enter the `show vlans` command to view the VLAN interface.

```
user@host> show vlans
Routing instance  VLAN name  Tag  Interfaces
default-switch   default    1
default-switch   vlan100    100  ge-0/0/3.0*
                 ge-0/0/4.0*
                 ge-7/0/5.0*
```

From operational mode, enter the `show ethernet-switching interface` command to view the information about Ethernet switching interfaces.

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                        LH - MAC limit hit, DN - interface down,
                        MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled,
```

SCTL - shutdown by Storm-control )

Logical interface	Vlan members	TAG	MAC limit	STP state	Logical interface flags	Tagging
ge-0/0/3.0			16383		DN	untagged
	vlan100	100	1024	Discarding		untagged
ge-0/0/4.0			16383		DN	untagged
	vlan100	100	1024	Discarding		untagged
ge-7/0/5.0			16383		DN	tagged
	vlan100	100	1024	Discarding		tagged

## Meaning

The output shows the VLANs are configured and working fine.

## RELATED DOCUMENTATION

| [Example: Configuring an IRB Interface](#)