NETWORKS

Junos® OS

Chassis Cluster Feature Guide for High-End SRX Series Devices

Release

15.1X49-D70

Modified: 2016-12-08

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

Junos® OS Chassis Cluster Feature Guide for High-End SRX Series Devices
15.1X49-D70

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 http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- SRX Series
- vSRX

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

[edit]
user@host# **edit system scripts**
[edit system scripts]

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   [edit system scripts]
   user@host# **load merge relative /var/tmp/ex-script-snippet.conf**
   load complete

For more information about the **load** command, see CLI Explorer.

Documentation Conventions

Table 1 on page xvii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xvii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		    route default {
		      nexthop *address*;
		      retain;
		    }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/

- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Introduction to Chassis Cluster

Chassis Cluster Overview

Supported Platforms SRX Series, vSRX

High Availability Using Chassis Clusters

Modern networks require high availability. In order to accommodate this requirement, Juniper Networks 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 node, providing device, interface, and service level redundancy.

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.

How High Availability Is Achieved by Chassis Cluster

- The network node redundancy is achieved by grouping a pair of the same kind of supported SRX Series devices into a cluster.
- The devices must be running the same version of the Junos operating system (Junos OS).

- SRX Series devices must be the same model, and all SPCs, network processing cards (NPCs), and input/output cards (IOCs) on high-end platforms must have the same slot placement and hardware revision.
- 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.

Chassis Cluster Active/Active and Active/Passive Modes

A chassis cluster in active/active mode has transit traffic passing through both nodes of the cluster all of the time. Whereas a chassis cluster in active/passive mode only has transit traffic passing through the primary node while the backup node waits in hot standby.

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.

The control plane software operates in active or backup mode.

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.
- Support for 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 Feature Guide for Security Devices*.

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

Starting with Junos OS Release 10.4, SRX Series devices 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.

IPsec and Chassis Cluster

High-end SRX Series devices have a chassis cluster control port that is used to connect two SRX Series devices to form a chassis cluster. To ensure secure login and to prevent attackers from gaining privileged access through this control port, an internal IPsec SA is installed. Besides using internal IPsec to secure RSH and RCP between the primary and backup Routing Engines, the internal IPsec SA is installed on all the Services Processing Units (SPUs). An attacker cannot access any of the RSH services without knowing the internal IPsec key.

The internal IPsec SA requires authorization for RSH on SPU and the Routing Engine. For telnet, authorization is only required for SPU since telnet for Routing Engine requires a password.

You set up the IPsec internal SA using the **security internal-security-association** CLI command. You can configure the **security internal-security-association** on a node and then enable it to activate secure login. The **security internal-security-association** CLI command does not need to be set up on each node. When you commit the configuration, both nodes are synchronized.

NOTE: The SA in this scenario is not the point-to-point security association, because it is used to communicate with any Routing Engine or SPU on the internal network. Only 3des-cbc encryption algorithm is supported.

When secure login is configured, the IPsec-based **rlogin** (for starting a terminal session on a remote host) and **rcmd** (remote command) commands are enforced so an attacker cannot gain privileged access or observe traffic that contains administrator commands and outputs.

Related Documentation

- Preparing Your Equipment for Chassis Cluster Formation on page 39
- Understanding Chassis Cluster Redundancy Groups on page 77
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 85

Chassis Cluster Supported Features

Supported Platforms SRX5400, SRX5600, SRX5800

Table 3 on page 6 lists the features that are supported on high-end SRX Series devices in a chassis cluster.

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Address Books and Address Sets	Address books	Yes	Yes	Yes	Yes
	Address sets	Yes	Yes	Yes	Yes
	Global address objects or sets	Yes	Yes	Yes	Yes
	Nested address groups	Yes	Yes	Yes	Yes
Administrator Authentication Support	Local authentication	Yes	Yes	Yes	Yes
	RADIUS	Yes	Yes	Yes	Yes
	TACACS+	Yes	Yes	Yes	Yes
Alarms	Chassis alarms	Yes	Yes	Yes	Yes
	Interface alarms	Yes	Yes	Yes	Yes
	System alarms	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Application Identification [1]	Application identification–synchronizing in a chassis cluster	Yes	Yes	Yes	Yes
	Application firewall (AppFW)	Yes	Yes	Yes	Yes
	Application QoS (AppQoS)	Yes	Yes	Yes	Yes
	Application tracking (AppTrack)	Yes	Yes	Yes	Yes
	Custom application signatures and signature groups	Yes	Yes	Yes	Yes
	Heuristics-based detection	Yes	Yes	Yes	Yes
	IDP	Yes	Yes	Yes	Yes
	Jumbo frames	Yes	Yes	Yes	Yes
	Nested application identification	Yes	Yes	Yes	Yes
	Onbox application tracking statistics (AppTrack)	Yes	Yes	Yes	Yes
	SSL proxy	Yes	Yes	Yes	Yes
	Subscription license enforcement	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
ALGs	DNS ALG	Yes	Yes	Yes	Yes
	DNS doctoring support	Yes	Yes	Yes	Yes
	DNS, FTP, RTSP, and TFTP ALGs (Layer 2) with chassis clustering	Yes	Yes	Yes	Yes
	DSCP marking for SIP, H.323, MGCP, and SCCP ALGs	Yes	Yes	Yes	Yes
	FTP	Yes	Yes	Yes	Yes
	H.323	Yes	Yes	Yes	Yes
	H.323–Avaya H.323	Yes	Yes	Yes	Yes
	MGCP	Yes	Yes	Yes	Yes
	PPTP	Yes	Yes	Yes	Yes
	RPC–MS RPC	Yes	Yes	Yes	Yes
	RPC–Sun RPC	Yes	Yes	Yes	Yes
	RSH	Yes	Yes	Yes	Yes
	RTSP	Yes	Yes	Yes	Yes
	SIP–NEC SIP	Yes	Yes	Yes	Yes
	SIP–SCCP SIP	Yes	Yes	Yes	Yes
	SQL	Yes	Yes	Yes	Yes
	TALK TFTP	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Attack Detection and Prevention (Screens)	Bad IP option	Yes	Yes	Yes	Yes
	Block fragment traffic	Yes	Yes	Yes	Yes
	FIN flag without ACK flag	Yes	Yes	Yes	Yes
	ICMP flood protection	Yes	Yes	Yes	Yes
	ICMP fragment protection	Yes	Yes	Yes	Yes
	IP address spoof	Yes	Yes	Yes	Yes
	IP address sweep	Yes	Yes	Yes	Yes
	IP record route option	Yes	Yes	Yes	Yes
	IP security option	Yes	Yes	Yes	Yes
	IP stream option	Yes	Yes	Yes	Yes
	IP strict source route option	Yes	Yes	Yes	Yes
	IP timestamp option	Yes	Yes	Yes	Yes
	Land attack protection land	Yes	Yes	Yes	Yes
	Large size ICMP packet protection	Yes	Yes	Yes	Yes
	Loose source route option	Yes	Yes	Yes	Yes
	Ping of death attack protection	Yes	Yes	Yes	Yes
	Port scan	Yes	Yes	Yes	Yes
	Source IP-based session limit	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
	SYN-ACK-ACK proxy protection	Yes	Yes	Yes	Yes
	SYN and FIN flags	Yes	Yes	Yes	Yes
	SYN flood protection	Yes	Yes	Yes	Yes
	SYN fragment protection	Yes	Yes	Yes	Yes
	TCP address sweep	Yes	Yes	Yes	Yes
	TCP packet without flag	Yes	Yes	Yes	Yes
	Teardrop attack protection	Yes	Yes	Yes	Yes
	UDP address sweep	Yes	Yes	Yes	Yes
	UDP flood protection	Yes	Yes	Yes	Yes
	Unknown protocol	Yes	Yes	Yes	Yes
	WinNuke attack protection	Yes	Yes	Yes	Yes
Chassis Management	Allow chassis management	Yes	Yes	Yes	Yes
	CX111 3G adapter support	No	No	No	No
	IEEE 802.3af / 802.3at support	No	No	No	No
	Chassis cluster SPC insert	Not supported for SRX5000 line	Not supported for SRX5000 line	Not supported for SRX5000 line	Not supported for SRX5000 line

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Class of Service	Classifiers	Yes	Yes	Yes	Yes
	Code-point aliases (IEEE 802.1)	Yes	Yes	Yes	Yes
	Egress interface shaping	Yes	Yes	Yes	Yes
	Forwarding classes	Yes	Yes	Yes	Yes
	Ingress interface	Yes	Yes	Yes	Yes
	Policer schedulers (hierarchical schedulers)	Yes	Yes	Yes	Yes
	Simple filters	–	–	–	–
	Transmission queues	Yes	Yes	Yes	Yes
DHCP	DHCP client	Yes	Yes	Yes	Yes
	DHCP relay agent	Yes	Yes	Yes	Yes
	DHCP server	Yes	Yes	Yes	Yes
	DHCP server address pools	Yes	Yes	Yes	Yes
	DHCP server static mapping	Yes	Yes	Yes	Yes
	DHCPv6[2]	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Diagnostics Tools	CLI terminal	Yes	Yes	Yes	Yes
	J-Flow version 5 and version 8	Yes	Yes	Yes	Yes
	J-Flow version 9	No	No	No	No
	Flowd monitoring	Yes	Yes	Yes	Yes
	Ping host	Yes	Yes	Yes	Yes
	Ping MPLS	No	No	No	No
	Traceroute	Yes	Yes	Yes	Yes
Dynamic VPN	Package dynamic VPN client	–	–	–	–
Ethernet Interfaces	10/100/1000 MB Ethernet interface	Yes	Yes	Yes	Yes
	10-Gigabit Ethernet Interface SFP+ slots	Yes	Yes	Yes	Yes
	40/100-Gigabit Ethernet interface MPC slots Gigabit	SRX5000 line devices only	Yes	Yes	Yes
	Ethernet, Copper (10-Mbps, 100-Mbps, or 1000-Mbps port)	Yes	Yes	Yes	Yes
	Gigabit Ethernet interface	Yes	Yes	Yes	Yes
	Promiscuous mode on Ethernet interface	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Ethernet Link Aggregation	LACP/LAG cross IOC (inter-IOC)	Yes	Yes	Yes	Yes
	LACP (port priority) Layer 3 Mode	Yes	Yes	Yes	Yes
	LACP (port priority) Layer 2 Mode	No	No	No	No
	Layer 3 LAG on routed ports	Yes	Yes	Yes	Yes
	Static LAG (routing)	Yes	Yes	Yes	Yes
	Static LAG (switching)	No	No	No	No
	Switching mode	No	No	No	No
File Management	Deletion of backup software image	Yes	Yes	Yes	Yes
	Deletion of individual files	Yes	Yes	Yes	Yes
	Download of system files	Yes	Yes	Yes	Yes
	Encryption and decryption of configuration files	Yes	Yes	Yes	Yes
	Management of account files	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Firewall Authentication	Firewall authentication on Layer 2 transparent authentication	Yes	Yes	Yes	Yes
	LDAP authentication server	Yes	Yes	Yes	Yes
	Local authentication server	Yes	Yes	Yes	Yes
	Pass-through authentication	Yes	Yes	Yes	Yes
	RADIUS authentication server	Yes	Yes	Yes	Yes
	SecurID authentication server	Yes	Yes	Yes	Yes
	Web authentication	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Flow-Based and Packet-Based Processing	Alarms and auditing	Yes	Yes	Yes	Yes
	End-to-end packet debugging	Yes	Yes	Yes	Yes
	Express Path	SRX5000 line only	No	No	No
	Flow-based processing	Yes	Yes	Yes	Yes
	Host bound fragmented traffic	Yes	Yes	Yes	Yes
	Network processor bundling	Yes	Yes	Yes	Yes
	Packet-based processing	No	No	No	No
	Selective stateless packet-based services	No	No	No	No
GPRS	GPRS (transparent mode and route mode)	Yes	Yes	No	No

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
GTPv2	IMSI prefix and APN filtering	Yes	Yes	No	No
	Message-length filtering	Yes	Yes	No	No
	Message-rate limiting	Yes	Yes	No	No
	Message-type filtering	Yes	Yes	No	No
	Packet sanity check	Yes	Yes	No	No
	Policy-based inspection	Yes	Yes	No	No
	Restart GTPv2 path	Yes	Yes	No	No
	Sequence-number and GTP-U validation	Yes	Yes	No	No
	Stateful inspection	Yes	Yes	No	No
	Traffic logging	Yes	Yes	No	No
	Tunnel cleanup	Yes	Yes	No	No

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
IDP	Alarms and auditing	Yes	Yes	Yes	Yes
	Cryptographic key handling	Yes	Yes	Yes	Yes
	DSCP marking	Yes	Yes	Yes	Yes
	IDP and application identification	Yes	Yes	Yes	Yes
	IDP and UAC coordinated threat control	Yes	Yes	Yes	Yes
	IDP class-of-service action	Yes	Yes	Yes	Yes
	IDP inline tap mode	Yes	Yes	Yes	Yes
	IDP logging	Yes	Yes	Yes	Yes
	IDP monitoring and debugging	Yes	Yes	Yes	Yes
	IDP policy	Yes	Yes	Yes	Yes
	IDP security packet capture	Yes	Yes	Yes	Yes
	IDP signature database	Yes	Yes	Yes	Yes
	IDP SSL inspection	Yes	Yes	Yes	Yes
	IPS rule base	Yes	Yes	Yes	Yes
	Jumbo frames	Yes	Yes	Yes	Yes
	Performance and capacity tuning for IDP	Yes	Yes	Yes	Yes
	SNMP MIB for IDP monitoring	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
IPsec [3]	AH protocol	Yes	Yes	Yes	Yes
	Alarms and auditing	Yes	Yes	Yes	Yes
	Antireplay (packet replay attack prevention)	Yes	Yes	Yes	Yes
	Autokey management	Yes	Yes	Yes	Yes
	Dead peer detection (DPD)	Yes	Yes	Yes	Yes
	Dynamic IPsec VPNs	No	No	No	No
	External Extended Authentication (XAuth) to a RADIUS server for remote access connections	Yes	Yes	Yes	Yes
	Group VPN with dynamic policies (server functionality)	Yes	Yes	Yes	Yes
	IKEv1 and IKEv2	Yes	Yes	Yes	Yes
	Manual key management	Yes	Yes	Yes	Yes
	Policy-based and route-based VPNs	Yes	Yes	Yes	Yes
	Route-based VPN support	Yes	Yes	Yes	Yes
	Tunnel mode	Yes	Yes	Yes	Yes
	VPN monitoring (proprietary)	Yes	Yes	Yes	Yes
	Virtual router	Yes	Yes	Yes	Yes
IPv6	IPv6 support	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Layer 2 Mode	802.1x port-based network authentication	No	No	No	No
	Flexible Ethernet services	No	No	No	No
	IRB interface	Yes	Yes	Yes	Yes
	LLDP and LLDP-MED	No	No	No	No
	MAC limit (port security)	Yes	Yes	Yes	Yes
	Q-in-Q tunneling	No	No	No	No
	Spanning Tree Protocol	No	No	No	No
	VLAN retagging	Yes	Yes	Yes	Yes
	VLANs	Yes	Yes	Yes	Yes
Multicast VPN	Basic multicast features in C-instance	No	No	No	No
	Multicast VPN membership discovery with BGP	No	No	No	No
	P2MP LSP support	No	No	No	No
	P2MP OAM to P2MP LSP ping	No	No	No	No
	Reliable multicast VPN routing information exchange	No	No	No	No

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
NAT	Destination IP address translation	Yes	Yes	Yes	Yes
	Disabling source	Yes	Yes	Yes	Yes
	Interface source NAT pool port	Yes	Yes	Yes	Yes
	NAT address pool utilization threshold status	Yes	Yes	Yes	Yes
	NAT port randomization	Yes	Yes	Yes	Yes
	NAT traversal (NAT-T) for site-to-site IPsec VPNs (IPv4)	Yes	Yes	Yes	Yes
	Persistent NAT	Yes	Yes	Yes	Yes
	Persistent NAT binding for wildcard ports	Yes	Yes	Yes	Yes
	Persistent NAT hairpinning	Yes	Yes	Yes	Yes
	Pool translation	Yes	Yes	Yes	Yes
	Proxy ARP (IPv4)	Yes	Yes	Yes	Yes
	Proxy NDP (IPv6)	Yes	Yes	Yes	Yes
	Removal of persistent NAT query bindings	Yes	Yes	Yes	Yes
	Rule-based NAT	Yes	Yes	Yes	Yes
	Rule translation	Yes	Yes	Yes	Yes
	Source address and group address translation for multicast flows	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
	Source IP address translation	Yes	Yes	Yes	Yes
	Static NAT	Yes	Yes	Yes	Yes
Network Operations and Troubleshooting Support	Event policies	Yes	Yes	Yes	Yes
	Event scripts	Yes	Yes	Yes	Yes
	Operation scripts	Yes	Yes	Yes	Yes
	XSLT commit scripts	Yes	Yes	Yes	Yes
Packet Capture	Packet capture	Yes	Yes	Yes	Yes
Public Key Infrastructure	Automated certificate enrollment using SCEP	Yes	Yes	Yes	Yes
	Automatic generation of self-signed certificates	Yes	Yes	Yes	Yes
	CRL update at user-specified interval	Yes	Yes	Yes	Yes
	Digital signature generation	Yes	Yes	Yes	Yes
	Entrust, Microsoft, and Verisign certificate authorities (CAs)	Yes	Yes	Yes	Yes
	IKE support	Yes	Yes	Yes	Yes
	Manual installation of DER-encoded and PEM-encoded CRLs	Yes	Yes	Yes	Yes
Remote Device Access	Reverse Telnet	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
RPM Probe	RPM probe	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
Routing	BGP	Yes	Yes	Yes	Yes
	BGP extensions for IPv6	Yes	Yes	Yes	Yes
	Compressed Real-Time Transport Protocol (CRTP)	Yes	Yes	Yes	Yes
	Internet Group Management Protocol (IGMP)	Yes	Yes	Yes	Yes
	IPv4 options and broadcast Internet diagrams	Yes	Yes	Yes	Yes
	IPv6 routing, forwarding, global address configuration, and Internet Control Message Protocol (ICMP)	Yes	Yes	Yes	Yes
	IS-IS	Yes	Yes	Yes	Yes
	Multiple virtual routers	Yes	Yes	Yes	Yes
	Neighbor Discovery Protocol (NDP) and Secure Neighbor Discovery Protocol (SEND)	Yes	Yes	Yes	Yes
	OSPF v2	Yes	Yes	Yes	Yes
	OSPF v3	Yes	Yes	Yes	Yes
	RIP next generation (RIPng)	Yes	Yes	Yes	Yes
	RIP v1, v2	Yes	Yes	Yes	Yes
	Static routing	Yes	Yes	Yes	Yes
		Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
	Virtual Router Redundancy Protocol (VRRP)				
Secure Web Access	CAs	Yes	Yes	Yes	Yes
	HTTP	Yes	Yes	Yes	Yes
	HTTPS	Yes	Yes	Yes	Yes
Security Policy	Security policy	Yes	Yes	Yes	Yes
Security Zones	Functional zone	Yes	Yes	Yes	Yes
	Security zone	Yes	Yes	Yes	Yes
Session Logging	Acceleration of security and traffic logging	Yes	Yes	Yes	Yes
	Aggressive session aging	Yes	Yes	Yes	Yes
	Getting information about sessions	Yes	Yes	Yes	Yes
	Logging to a single server	Yes	Yes	Yes	Yes
	Session logging with NAT information	Yes	Yes	Yes	Yes
SMTP	SMTP	Yes	Yes	Yes	Yes
SNMP	SNMP v1, v2, v3	Yes	Yes	Yes	Yes
Stateless Firewall Filters	Stateless firewall filters (ACLs)	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
System Log Files	System log archival	Yes	Yes	Yes	Yes
	System log configuration	Yes	Yes	Yes	Yes
	Disabling system logs	Yes	Yes	Yes	Yes
	Filtering system log messages	Yes	Yes	Yes	Yes
	Multiple system log servers (control plane logs)	Yes	Yes	Yes	Yes
	Sending system log messages to a file	Yes	Yes	Yes	Yes
	Sending system log messages to a user terminal	Yes	Yes	Yes	Yes
	Viewing data plane logs	Yes	Yes	Yes	Yes
	Viewing system log messages	Yes	Yes	Yes	Yes
Transparent Mode	Bridge domain and transparent mode	Yes	Yes	Yes	Yes
	Class of service	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
UTM	Antispam	Yes	Yes	No	No
	Antivirus–Express	–	–	–	–
	Antivirus–Full	–	–	–	–
	Antivirus– Sophos	Yes	No	No	No
	Content filtering	Yes	Yes	Yes	Yes
	Stateful active/active cluster mode	No	No	No	No
	Web filtering–Enhanced	Yes	Yes	No	No
	Web filtering–Juniper Networks local	Yes	Yes	Yes	Yes
	Web filtering–Surf-control	–	–	–	–
	Web filtering–Websense redirect	Yes	Yes	No	No
Upgrading and Rebooting	Autorecovery	Yes	Yes	Yes	Yes
	Boot device configuration	Yes	Yes	Yes	Yes
	Boot device recovery	Yes	Yes	Yes	Yes
	Chassis components control	Yes	Yes	Yes	Yes
	Chassis restart	Yes	Yes	Yes	Yes
	Dual-root partitioning	No	No	No	No
	ISSU	Yes	Yes	Yes	Yes
	WELF support	Yes	Yes	Yes	Yes

Table 3: Features Supported on High-End SRX Series Devices in a Chassis Cluster *(continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
User Interfaces	CLI	Yes	Yes	Yes	Yes
	J-Web user interface	No	No	No	No
	Junos XML protocol	Yes	Yes	Yes	Yes
	Network and Security Manager	Yes	Yes	Yes	Yes
	Session and Resource Control (SRC) application	No	No	No	No

[1] When the application ID is identified before session failover, the same action taken before the failover is effective after the failover. That is, the action is published to AppSecure service modules and takes place based on the application ID of the traffic. If the application is in the process of being identified before a failover, the application ID is not identified and the session information is lost. The application identification process will be applied on new sessions created on new primary node.

[2] DHCPv6 is supported on SRX Series devices running Junos OS Release 12.1 and later releases.

[3] IPsec in active/active chassis cluster on high-end SRX Series devices has the limitation that Z-mode traffic is not supported. This limitation pertains to Junos OS Release 12.3X48 and later and must be avoided.

Chassis Cluster Features Support

Table 4 on page 27 lists the chassis cluster features that are supported on high-end SRX Series devices.

Table 4: Chassis Cluster Feature Support on High-End SRX Series Devices

Features	SRX5000 Line
Active/backup Routing Engine group (RG0)	Yes
Active/active data redundancy groups (RGx)	Yes
Aggregate Interfaces (link aggregation)	Yes
Autorecovery of fabric link	Yes
Chassis cluster extended cluster ID	Yes

Table 4: Chassis Cluster Feature Support on High-End SRX Series Devices *(continued)*

Features	SRX5000 Line
Chassis cluster formation	Yes
Encrypted control link	Yes
Chassis clusters (active/backup and active/active)	Yes
Control link recovery	Yes
Control plane failover	Yes
Dampening time between back-to-back redundancy group failovers	Yes
Data plane failover	Yes
Dual control links (redundant link for failover)	Yes
Dual fabric links	Yes
IP monitoring	Yes
Flow forwarding	Yes
Graceful restart routing protocols	Yes
Graceful protocol restart for BGP	Yes
Graceful protocol restart for IS-IS	Yes
Graceful protocol restart for OSPF	Yes
Graceful Routing Engine switchover (GRES) (between nodes)	Yes
HA fabric forwarded packet reordering Interface	Yes
HA monitoring	Yes
In-band cluster upgrade (ICU)	No
Junos OS flow-based routing functionality	Yes
LACP support for Layer 3	Yes
Layer 2 Ethernet switching capability	No
Layer 2 transparent mode LAG	Yes
Layer 3 LAG	Yes

Table 4: Chassis Cluster Feature Support on High-End SRX Series Devices *(continued)*

Features	SRX5000 Line
Local interface support (non-reth)	Yes
Low-Impact ISSU	Yes
Multicast in HA mode	Yes
Network Time Protocol (NTP) time synchronization	Yes
Point-to-Point Protocol over Ethernet (PPPoE) over redundant Ethernet interface	No
Quality of service (QoS)	Yes
Redundancy group 0 (backup for Routing Engine)	Yes
Redundancy groups 1 through 128	Yes
Redundant Ethernet interfaces	Yes
Redundant Ethernet or aggregate Ethernet interface monitoring	Yes
SPU monitoring	Yes
Synchronization–backup node configuration from primary node	Yes
Synchronization–configuration	Yes
Synchronization–Dynamic Routing Protocol (DRP)	Yes
Synchronization–policies	Yes
Synchronization– session state sync (RTO sync)	Yes
TCP support for DNS	Yes
Upstream device IP address monitoring on a backup interface	Yes
Virtual Router Redundancy Protocol (VRRP) version 3	No
WAN interfaces	No

Related Documentation

- Chassis Cluster Overview on page 3
- Chassis Cluster Limitations on page 30

Chassis Cluster Limitations

Supported Platforms SRX Series, vSRX

The SRX Series devices have the following chassis cluster limitations:

Chassis Cluster

- On SRX Series device failover, access points on the Layer 2 switch reboot and all wireless clients lose connectivity for 4 to 6 minutes.
- On all high-end SRX Series devices, screen statistics data can be gathered on the primary device only.
- On all high-end SRX Series devices, in large chassis cluster configurations, you need to increase the 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.
- When an SRX Series device 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.

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
  ```
- Any packet-based services such as MPLS and CLNS are not supported.
- On all SRX Series devices, the packet-based forwarding for MPLS and ISO protocol families is not supported.

- On high-end SRX Series 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.
- On SRX Series devices 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.

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.

IPv6

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

GPRS

- On all high-end SRX Series 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.

Related Documentation

- Chassis Cluster Overview on page 3
- Preparing Your Equipment for Chassis Cluster Formation on page 39

CHAPTER 2

Understanding Chassis Cluster License Requirements

Understanding Chassis Cluster Licensing Requirements

Supported Platforms SRX Series, vSRX

Some Junos OS software features require a license to activate the feature. To enable a licensed feature, you need to purchase, install, manage, and verify a license key that corresponds to each licensed feature.

There is no separate license required for chassis cluster. However, to configure and use the licensed feature in a chassis cluster setup, you must purchase one license per feature per device and the license needs to be installed on both nodes of the chassis cluster. Each license is tied to one software feature pack, and that license is valid for only one device.

For chassis cluster, you must install licenses that are unique to each device and cannot be shared between the devices. Both devices (which are going to form a chassis cluster) must have the valid, identical features licenses installed on them. If both devices do not have an identical set of licenses, then after a failover, a particular feature (that is, a feature that is not licensed on both devices) might not work or the configuration might not synchronize in chassis cluster formation.

Licensing is usually ordered when the device is purchased, and this information is bound to the chassis serial number. For example, Intrusion Detection and Prevention (IDP) is a licensed feature and the license for this specific feature is tied to the serial number of the device.

For information about how to purchase software licenses, contact your Juniper Networks sales representative at http://www.juniper.net/in/en/contact-us/.

Related Documentation

- Installing Licenses on the Devices in a Chassis Cluster on page 34
- Verifying Licenses for an SRX Series Device in a Chassis Cluster on page 36

Installing Licenses on the Devices in a Chassis Cluster

Supported Platforms SRX Series, vSRX

You can add a license key from a file or a URL, from a terminal, or from the J-Web user interface. Use the ***filename*** option to activate a perpetual license directly on the device. Use the ***url*** option to send a subscription-based license key entitlement (such as unified threat management [UTM]) to the Juniper Networks licensing server for authorization. If authorized, the server downloads the license to the device and activates it.

Before adding new licenses, complete the following tasks:

- Purchase the required licenses.
- Set the chassis cluster node ID and the cluster ID. See *Example: Setting the Chassis Cluster Node ID and Cluster ID for Branch SRX Series Devices* or "Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices" on page 56.
- Ensure that your SRX Series device has a connection to the Internet (if particular feature requires Internet or if (automatic) renewal of license through internet is to be used). For instructions on establishing basic connectivity, see the Getting Started Guide or Quick Start Guide for your device.

To install licenses on the primary node of an SRX Series device in a chassis cluster:

1. Run the **show chassis cluster status** command and identify which node is primary for redundancy group 0 on your SRX Series device.

   {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 and node 1 is secondary.

2. From CLI operational mode, enter one of the following CLI commands:
   - To add a license key from a file or a URL, enter the following command, specifying the filename or the URL where the key is located:

     user@host> **request system license add *filename* | *url***

   - To add a license key from the terminal, enter the following command:

     user@host> **request system license add terminal**

3. When prompted, enter the license key, separating multiple license keys with a blank line.

   If the license key you enter is invalid, an error appears in the CLI output when you press Ctrl+d to exit license entry mode.

4. Verify the installed licenses.

For more details, see Adding New Licenses (CLI Procedure).

To install licenses on the secondary node of an SRX Series device in a chassis cluster:

1. Initiate a failover to change node 1 (secondary node) to be the primary node:

   {primary:node0}

   user@host> request chassis cluster failover redundancy-group 0 node 1

   ```
   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   - -
   Initiated manual failover for redundancy group 0
   ```

   NOTE: Initiating a failover to the secondary node is not required if you are installing licenses manually on the device. However, if you are installing the license directly from the Internet, you must initiate a failover.

2. Repeat the steps described in "Step-by-Step Procedure" on page 34 to install licenses on the secondary node.

3. Reboot the device for licenses to take effect.

NOTE: You must install the updated license on both nodes of the chassis cluster before the existing license expires.

NOTE: In a chassis cluster configuration, when one device has a license installed, and the other device does not have the same license installed, an error message is displayed when you try to configure that specific feature as shown in the following example:

[edit security utm feature-profile web-filtering type]

```
  'type juniper-enhanced'
    warning: requires 'wf_key_websense_ewf' license
<<<<<<<<<<<<<<<<<<<<<<<
configuration check succeeds
```

TIP: If you are not using any specific feature or license , you can delete the license from both devices in a chassis cluster. You need to connect to each node separately to delete the licenses. For details, see *Example: Deleting a License Key*.

Related Documentation

- Verifying Licenses for an SRX Series Device in a Chassis Cluster on page 36
- Understanding Chassis Cluster Licensing Requirements on page 33

Verifying Licenses for an SRX Series Device in a Chassis Cluster

Supported Platforms SRX Series, vSRX

Purpose You can verify the licenses installed on both the devices in a chassis cluster setup by using the **show system license installed** command to view license usage.

Action Licenses details on node 0.

user@host> show system license installed

```
{primary:node0}
user@host> show system license
License usage:
                                 Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  logical-system                        1           26           0    permanent
  services-offload                      0            1           0    permanent

Licenses installed:
  License identifier: JUNOS363684
  License version: 2
  Valid for device: JN111A654AGB
  Features:
    services-offload - services offload mode
      permanent

  License identifier: JUNOS531744
  License version: 4
  Valid for device: JN111A654AGB
  Features:
    services-offload - services offload mode
      permanent

  License identifier: JUNOS558173
  License version: 4
  Valid for device: JN111A654AGB
  Features:
    logical-system-25 - Logical System Capacity
      permanent
```

Licenses details on node 1.

```
{secondary-hold:node1}
user@host> show system license
License usage:
                                 Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  idp-sig                               0            1           0    permanent
  logical-system                        1           26           0    permanent
  services-offload                      0            1           0    permanent

Licenses installed:
  License identifier: JUNOS209661
  License version: 2
  Valid for device: JN111AB4DAGB
  Features:
    idp-sig          - IDP Signature
      permanent

  License identifier: JUNOS336648
  License version: 2
  Valid for device: JN111AB4DAGB
  Features:
    logical-system-25 - Logical System Capacity
      permanent

  License identifier: JUNOS363685
```

```
License version: 2
Valid for device: JN111AB4DAGB
Features:
  services-offload - services offload mode
    permanent

License identifier: JUNOS531745
License version: 4
Valid for device: JN111AB4DAGB
Features:
  services-offload - services offload mode
    permanent
```

Meaning Use the fields **License version** and **Features** to make sure that licenses installed on both the nodes are identical.

Related Documentation

- Installing Licenses on the Devices in a Chassis Cluster on page 34
- Understanding Chassis Cluster Licensing Requirements on page 33

CHAPTER 3

Planning Your Chassis Cluster Configuration

Preparing Your Equipment for Chassis Cluster Formation

Supported Platforms SRX Series, vSRX

To form a chassis cluster, a pair of the same kind of supported SRX Series devices is combined to act as a single system that enforces the same overall security. SRX Series devices must meet the following requirements:

- Device-specific requirements:

  The following are the device-specific matches required to form a chassis cluster:

  - SRX5400, SRX5600, and SRX5800—The placement and type of Services Processing Cards (SPCs) must match in the two clusters.
- 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.

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.

You can deploy up to 255 chassis clusters in a Layer 2 domain. Clusters and nodes are identified in the following way:

- 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 are connected on a separated private VLANs.

Example: Following message is displayed when you are trying to set cluster ID greater than 15 and fabric and control link interfaces are not connected back-to-back or are not connected on a separated private 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.

Related Documentation

- Chassis Cluster Overview on page 3
- Chassis Cluster Limitations on page 30
- SRX Series Chassis Cluster Configuration Overview on page 40

SRX Series Chassis Cluster Configuration Overview

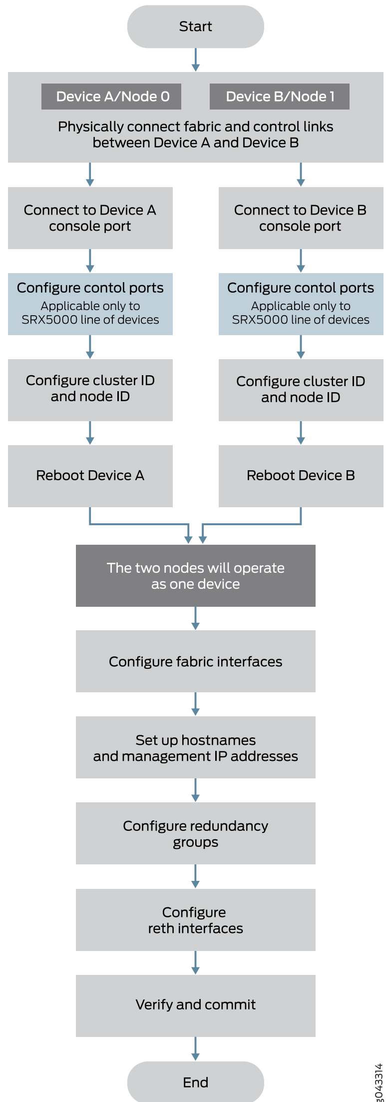
Supported Platforms SRX Series, vSRX

Figure 1 on page 41 shows a chassis cluster flow diagram.

Note the following prerequisites for configuring a chassis cluster:

- Confirm that hardware and software are the same on both devices.
- Confirm that license keys are the same on both devices.

Figure 1: Chassis Cluster Flow Diagram



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

NOTE: **For SRX5000 line chassis clusters, the placement and type of SPCs must match in the two devices.**

To create an SRX Series chassis cluster:

1. Physically connect a pair of the same kind of supported SRX Series devices together.
   a. Create the fabric link between two nodes in a cluster by connecting any pair of Ethernet interfaces.
   b. Connect the control ports that you will use on each device.

For more information, see "Connecting SRX Series Devices to Create a Chassis Cluster" on page 47.

2. Connect the first device to be initialized in the cluster to the console port. This is the node that forms the cluster.

   For connection instructions, see the Getting Started Guide for your device.

3. Configure the control ports (SRX5000 line only). See "Example: Configuring Chassis Cluster Control Ports" on page 73.

4. 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 Chassis Cluster Node ID and Cluster ID" on page 56.

5. Connect to the console port on the other device 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.

6. Configure the management interfaces on the cluster. See "Example: Configuring the Chassis Cluster Management Interface" on page 60

7. Configure the cluster with the CLI. See:

   a. Example: Configuring Chassis Cluster Redundancy Groups on page 81

   b. Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88

   c. Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 93

   d. Example: Configuring Chassis Cluster Interface Monitoring on page 178

8. Initiate manual failover. See "Initiating a Chassis Cluster Manual Redundancy Group Failover" on page 136.

9. Configure conditional route advertisement over redundant Ethernet interfaces. See "Understanding Conditional Route Advertising in a Chassis Cluster" on page 171.

10. Verify the configuration. See "Verifying a Chassis Cluster Configuration" on page 103.

- Fabric Links:

  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 123.

- Control Port:

  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 "Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster" on page 110.

For SRX5600 and SRX5800 devices, control ports must be on corresponding slots in the two devices. Table 5 on page 43 shows the slot numbering offsets:

Table 5: Slot Numbering Offsets

Device	Offset
SRX5800	12 (for example, **fpc3** and **fpc15**)
SRX5600	6 (for example, **fpc3** and **fpc9**)
SRX5400	3 (for example, **fpc3** and **fpc6**)

Related Documentation

- Chassis Cluster Overview on page 3
- Preparing Your Equipment for Chassis Cluster Formation on page 39
- Chassis Cluster Limitations on page 30

PART 2

Setting Up Chassis Cluster in SRX Series Devices

CHAPTER 4

Chassis Cluster Physical Setup

Connecting SRX Series Devices to Create a Chassis Cluster

Supported Platforms SRX Series

An SRX Series chassis cluster is created by physically connecting two identical cluster-supported SRX Series devices 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.

NOTE: You can connect two control links (SRX5600 and SRX5800 devices 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 109 and "Understanding Chassis Cluster Dual Fabric Links" on page 123.

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

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

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.

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

Figure 2 on page 48, Figure 3 on page 48, and Figure 4 on page 48 show pairs of SRX Series devices with the fabric links and control links connected.

Figure 2: Connecting SRX Series Devices in a Cluster (SRX5800 Devices)

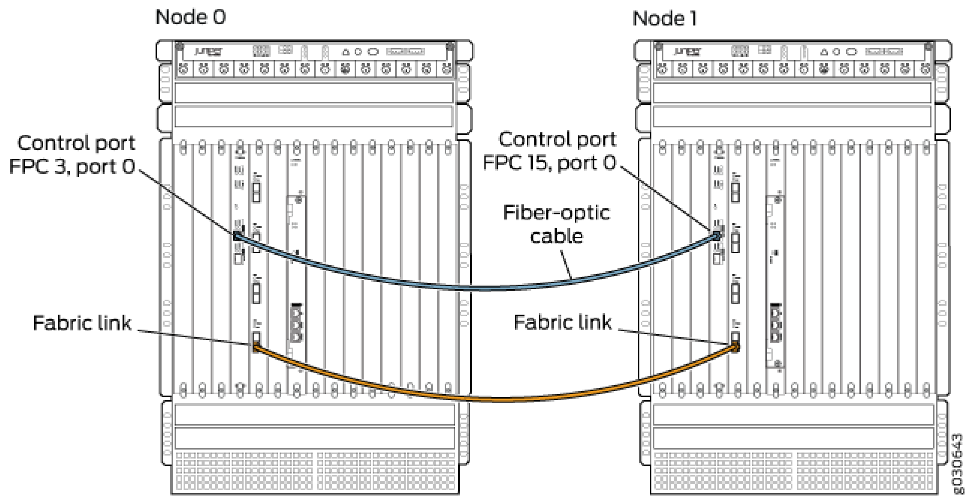


Figure 3: Connecting SRX Series Devices in a Cluster (SRX5600 Devices)

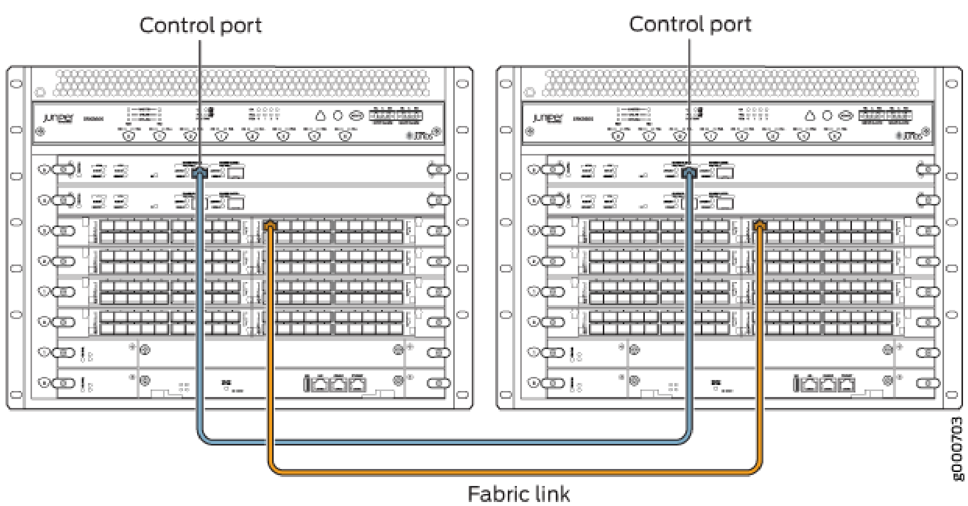


Figure 4: Connecting SRX Series Devices in a Cluster (SRX5400 Devices)

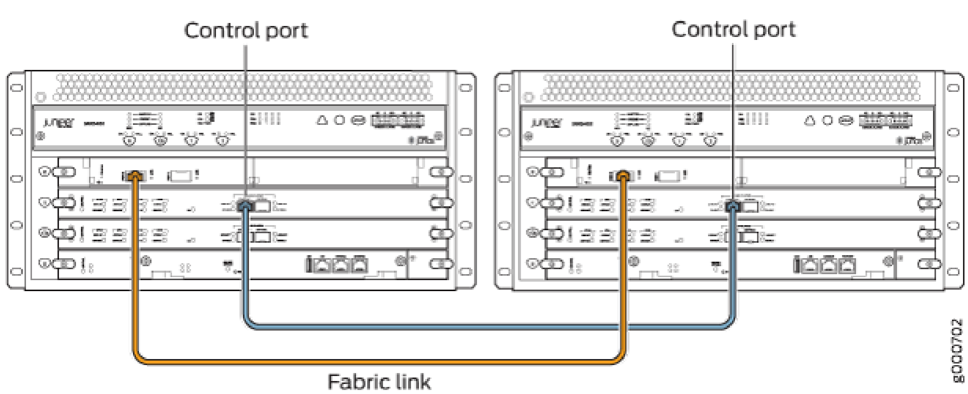


Figure 5 on page 49 and Figure 6 on page 49 show pairs of SRX Series devices with the fabric links and control links connected.

Figure 5: Connecting SRX Series Devices in a Cluster (SRX4100 Devices)

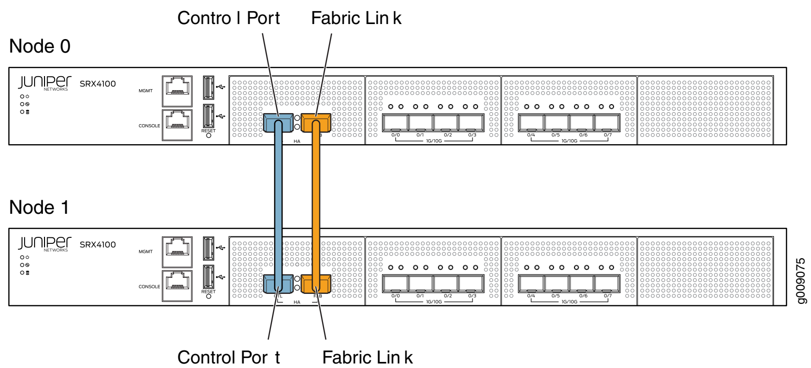
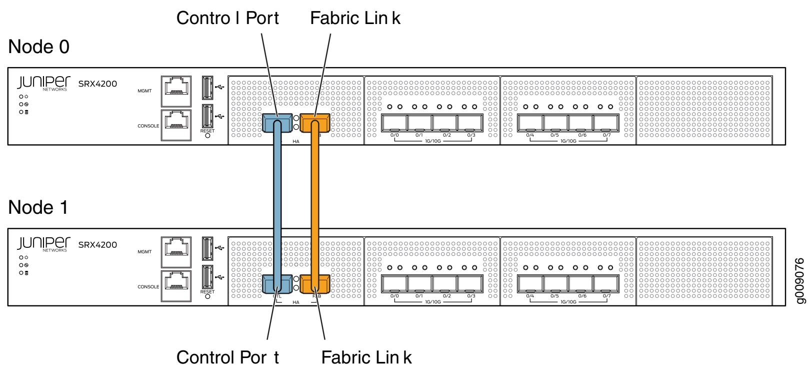


Figure 6: Connecting SRX Series Devices in a Cluster (SRX4200 Devices)



Related Documentation

- Preparing Your Equipment for Chassis Cluster Formation on page 39
- SRX Series Chassis Cluster Configuration Overview on page 40
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47

CHAPTER 5

Setting Up Chassis Cluster Identification

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

Supported Platforms SRX Series, vSRX

Normally, on SRX Series devices, the built-in interfaces are numbered as follows:

Devices	Built-In Interfaces				
For Most SRX Series Devices	ge-0/0/0	ge-0/0/1	ge-0/0/2	ge-0/0/3	. . .

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

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

The SRX4100 and SRX4200 devices use following HA ports:

- 10G Base-T control interface link
- 10G Base-T fabric interface link

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

On SRX4100 and SRX4200 devices, a dedicated fabric port is available for configuring fabric links to forward traffic between the two devices in the cluster. You can also use the dedicated fabric port for configuring dual fabric links on the devices. No configuration is required while using the dedicated fabric port for a single fabric link configuration

Table 6 on page 52 shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series devices that become node 0 and node 1 of the chassis cluster after the cluster is formed.

Table 6: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming

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
5800	Node 0	12 (FPC slots)	0 — 11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1		12 — 23	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1
5600	Node 0	6 (FPC slots)	0 — 5	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1		6 — 11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1
5400	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
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

Table 6: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming *(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
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

NOTE: See the hardware documentation for your particular model (SRX Series Services Gateways) for details about SRX Series devices. See *Interfaces Feature Guide for Security Devices* for a full discussion of interface naming conventions.

FPC Slot Numbering in SRX Series Devices

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 7 on page 53)

Figure 7: FPC Slot Numbering in an SRX Series Chassis Cluster (SRX5800 Devices)

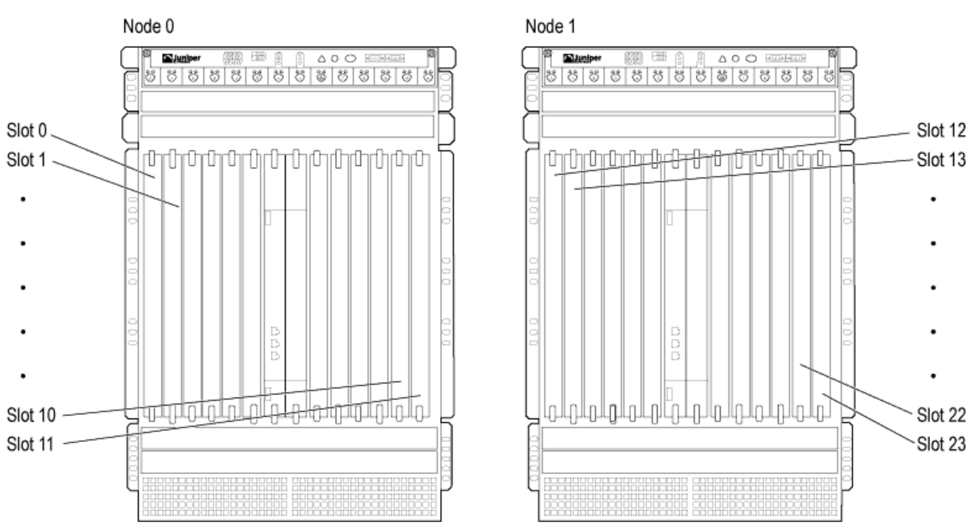


Figure 8 on page 54 and Figure 9 on page 54 shows the slot numbering for both of the SRX Series devices that become node 0 and node 1 of the chassis cluster after the cluster is formed.

Figure 8: FPC Slot Numbering in an SRX4100 Device

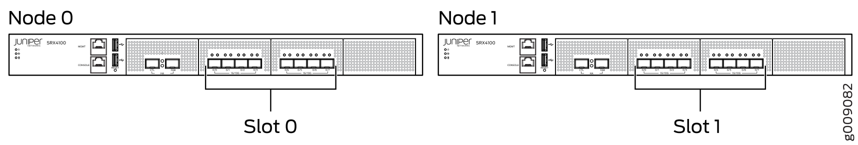
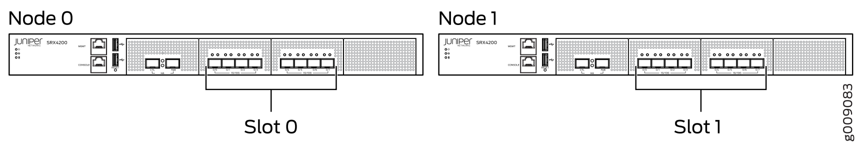


Figure 9: FPC Slot Numbering in an SRX4200 Device



In chassis cluster mode, all FPC related configuration is performed under **edit chassis node *node-id* fpc** hierarchy. In non-cluster mode, the FPC related configuration is performed under **edit chassis fpc** hierarchy.

SRX Series Services Gateways Interface Renumbering

After the devices are connected as a cluster, the slot numbering on one device 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 secondary node are renumbered internally.

The node 1 renumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. For example, see Table 7 on page 54 for interface renumbering on the SRX Series devices.

Table 7: SRX Series Services Gateways Interface Renumbering

Device	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX4100	7	xe-0/0/0	xe-7/0/0
SRX4200	7	xe-0/0/1	xe-7/0/1

NOTE: 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).

FPC Slot Numbering in SRX Series Devices Cards

NOTE: 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 device. 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

- Preparing Your Equipment for Chassis Cluster Formation on page 39
- SRX Series Chassis Cluster Configuration Overview on page 40
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47
- Example: Setting the Chassis Cluster Node ID and Cluster ID on page 56

Example: Setting the Chassis Cluster Node ID and Cluster ID

Supported Platforms SRX Series, vSRX

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.

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.

NOTE: 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 161.

Configuration

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.

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

To do this, physically connect control links between two devices, connect to the console port on the primary device, and configure control ports. Assign it a node ID, identify the cluster it will belong to, and then reboot the system. You can configure control ports with the FPCs and ports as the control link. (For more details, see “Example: Configuring Chassis Cluster Control Ports” on page 73.)

Verification

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
```

Related Documentation

- Connecting SRX Series Devices to Create a Chassis Cluster on page 47
- Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 51
- SRX Series Chassis Cluster Configuration Overview on page 40

CHAPTER 6

Setting up Chassis Cluster Management Interfaces

Management Interface on an Active Chassis Cluster

Supported Platforms SRX Series, vSRX

The **fxp0** interfaces function like standard management interfaces on SRX Series devices 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 devices 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).

Most of the SRX Series devices contain an fxp0 interface.

Related Documentation

- SRX Series Chassis Cluster Configuration Overview on page 40
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47
- Example: Configuring the Chassis Cluster Management Interface on page 60

Example: Configuring the Chassis Cluster Management Interface

Supported Platforms SRX Series, vSRX

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 for Branch SRX Series Devices* or "Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices" on page 56.

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.

NOTE: **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

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
```

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 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]
   set groups node1 system host-name node1-router
   set groups node1 interfaces fxp0 unit 0 family inet address 10.1.1.2/24
   ```

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

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

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 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.

Verification

Verifying the Chassis Cluster Management Interface Configuration

Purpose Verify the chassis cluster management interface configuration.

Action To verify the configuration is working properly, enter the **show config** command.

Related Documentation

- *Management Interface on an Active Chassis Cluster for Branch SRX Series Devices*

CHAPTER 7

Setting Up Fabric Interfaces on a Chassis Cluster

Understanding Chassis Cluster Fabric Interfaces

Supported Platforms SRX Series, vSRX

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.

Understanding Chassis Cluster Fabric Links

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.

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.

CAUTION: **After fabric interfaces have been configured on a chassis cluster, removing the fabric configuration on either node will cause the redundancy group 0 (RG0) secondary node to move to a disabled state. (Resetting a device to the factory default configuration removes the fabric configuration and thereby causes the RG0 secondary node to move to a disabled state.) After the fabric configuration is committed, do not reset either device to the factory default configuration.**

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 8 on page 64 shows the fabric interface types that are supported for SRX Series devices.

Table 8: Supported Fabric Interface Types for SRX Series Devices

SRX5000 line
Fast Ethernet
Gigabit Ethernet
10-Gigabit Ethernet
40-Gigabit Ethernet
100-Gigabit Ethernet NOTE: Starting in Junos OS Release 12.1X46-D10, this interface is supported.

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

NOTE: The SRX5K-MPC 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.

The IOC3 does not support the following command to set a PIC to go offline or online:

request chassis pic fpc-slot <fpc-slot> pic-slot <pic-slot> <offline | online> CLI command.

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.

WARNING:

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-backup 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.

For details about port and interface usage for management, control, and fabric links, see "Example: Setting the Chassis Cluster Node ID and Cluster ID" on page 56.

The fabric data link does not support fragmentation. To accommodate this state, jumbo frame support is enabled by default on the link with an MTU size of 8940 bytes. To ensure that 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 Session RTOs

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

NOTE: **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 must start exchanging probes. As soon as this happens, all the FPCs on the previously ineligible node is reset. They then come to online state and rejoin the cluster.

NOTE: **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.**

NOTE: **Starting with Junos OS Release 12.1X47-D10, the fabric monitoring feature is enabled by default on high-end SRX Series devices.**

Starting with Junos OS Release 12.1X47-D10, 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.

NOTE:

- 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.

Related Documentation

- Preparing Your Equipment for Chassis Cluster Formation on page 39
- SRX Series Chassis Cluster Configuration Overview on page 40
- Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 51
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47

Example: Configuring the Chassis Cluster Fabric Interfaces on page 68

Example: Configuring the Chassis Cluster Fabric Interfaces

Supported Platforms SRX Series, vSRX

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 Chassis Cluster Node ID and Cluster ID” on page 56.

Overview

In most SRX Series devices 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 8980 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.

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**.

NOTE: 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

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

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
```

Related Documentation

- Understanding Chassis Cluster Fabric Interfaces on page 63
- Verifying Chassis Cluster Data Plane Interfaces on page 129

CHAPTER 8

Setting Up Control Plane Interfaces on a Chassis Cluster

Understanding Chassis Cluster Control Plane and Control Links

Supported Platforms SRX Series, vSRX

Understanding the Chassis Cluster Control Plane

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 master Routing Engine fails, the secondary one is ready to assume control.

The control plane software:

- Runs on the Routing Engine and 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 master Routing Engine maintains state for the entire cluster, and only processes running on its node can update state information. The master Routing Engine synchronizes state for the secondary node and also processes all host traffic.

NOTE: For a single control link in a chassis cluster, the same control port must be used for the control link connection and for configuration on both nodes. For example, if port 0 is configured as a control port on node 0, then port 0 must be configured as a control port on node 1 with a cable connection between the two ports. For dual control links, control port 0 on node 0 must be connected to control port 0 on node 1 and control port 1 must be connected to control port 1 on node 1. Cross connections, that is, connecting port 0 on one node to port 1 on the other node and vice versa, do not work.

Understanding 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 is also used to synchronize the configuration between the nodes. When you submit configuration statements to the cluster, the configuration is automatically synchronized over the control link.

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

On SRX5400, SRX5600, and SRX5800 devices, by default, all control ports are disabled. Each SPC in a device has two control ports, and each device can have multiple SPCs plugged into it. To set up the control link in a chassis cluster with SRX5600 or SRX5800 devices, you connect and configure the control ports that you will use on each device (**fpcn** and **fpcn**) and then initialize the device in cluster mode.

For SRX4100 and SRX4200 devices, there are dedicated chassis cluster (HA) control ports available. No control link configuration is needed for SRX4100 and SRX4200 devices.

NOTE: For SRX4100 and SRX4200 devices, when devices are not in cluster mode, dedicated HA ports cannot be used as revenue ports or traffic ports.

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 51.

Related Documentation

- Example: Configuring Chassis Cluster Control Ports on page 73
- Preparing Your Equipment for Chassis Cluster Formation on page 39
- SRX Series Chassis Cluster Configuration Overview on page 40
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47

Example: Configuring Chassis Cluster Control Ports

Supported Platforms SRX Series, vSRX

This example shows how to configure chassis cluster control ports on SRX5400, SRX5600, and SRX5800 devices. 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 71.
- Physically connect the control ports on the devices. See “Connecting SRX Series Devices to Create a Chassis Cluster” on page 47.

Overview

By default, all control ports on SRX5400, SRX5600, and SRX5800 devices are disabled. After connecting the control ports, configuring the control ports, and establishing the chassis cluster, the control link is set up.

This example configures control ports with the following FPCs and ports as the control link:

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

Configuration

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

Step-by-Step Procedure

To configure control ports for use 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
  ```

Results

From 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 10 port 0;
  }
...
```

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

Verification

Verifying the Chassis Cluster Status

Purpose

Verify the chassis cluster status.

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
```

Meaning

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

Related Documentation

- Understanding Chassis Cluster Control Plane and Control Links on page 71
- Preparing Your Equipment for Chassis Cluster Formation on page 39
- SRX Series Chassis Cluster Configuration Overview on page 40
- Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 51
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47

CHAPTER 9

Setting Up Chassis Cluster Redundancy Groups

Understanding Chassis Cluster Redundancy Groups

Supported Platforms SRX Series, vSRX

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 master. By default, preemption is disabled. For more information on preemeption, 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.

NOTE: **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 81.)

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.

CAUTION: **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 master 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*). 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 pseudointerface 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 9 on page 80 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*.

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

Table 9: 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 9 on page 80 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.

Related Documentation

- Example: Configuring Chassis Cluster Redundancy Groups on page 81

Example: Configuring Chassis Cluster Redundancy Groups

Supported Platforms SRX Series, vSRX

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 for Branch SRX Series Devices* or "Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices" on page 56.
2. Configure the chassis cluster management interface. See "Example: Configuring the Chassis Cluster Management Interface" on page 60.
3. Configure the chassis cluster fabric. See "Example: Configuring the Chassis Cluster Fabric Interfaces" on page 68.

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

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. Specify whether a node with a higher priority can initiate a failover to become primary for the redundancy group.

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

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

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         secondary    no       no
    node1                     1           primary      yes      no
```

Related Documentation

- Understanding Chassis Cluster Redundancy Groups on page 77

CHAPTER 10

Setting Up Chassis Cluster Redundant Ethernet Interfaces

Understanding Chassis Cluster Redundant Ethernet Interfaces

Supported Platforms SRX Series, vSRX

A redundant Ethernet interface is a pseudointerface that includes at minimum one physical interface from each node of the cluster.

NOTE: For high-end SRX Series devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a chassis cluster deployment is 4,096.

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.

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

The maximum number of redundant Ethernet interfaces that you can configure varies, depending on the device type you are using, as shown in Table 10 on page 86. Note that the number of redundant Ethernet interfaces configured determines the number of redundancy groups that can be configured.

Table 10: Maximum Number of Redundant Ethernet Interfaces Allowed

Device	Maximum Number of reth Interfaces
SRX4100	128
SRX4200	128
SRX5400	128
SRX5600	128
SRX5800	128

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.

NOTE: 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.

NOTE: When using SRX Series devices in chassis cluster mode, we recommend that you do not configure any local interfaces (or combination of local interfaces) along with redundant Ethernet interfaces.

For example:

The following configuration of chassis cluster redundant Ethernet interfaces, in which interfaces are configured as local interfaces, is not supported:

```
ge-2/0/2 {
  unit 0 {
    family inet {
      address 192.0.2.1/24;
    }
  }
}
```

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 192.0.2.1/24;
      }
    }
  }
}
```

Related Documentation

- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 93
- Preparing Your Equipment for Chassis Cluster Formation on page 39
- SRX Series Chassis Cluster Configuration Overview on page 40
- Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 51
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47

Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses

Supported Platforms SRX Series, vSRX

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 for Branch SRX Series Devices* or "Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices" on page 56.
- 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 68.
- Understand how to set the chassis cluster node redundancy groups. See "Example: Configuring Chassis Cluster Redundancy Groups" on page 81.

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.

NOTE: 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

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 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
```

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 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 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**

   NOTE: 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**

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**

Results From configuration mode, confirm your configuration by entering the **show interfaces reth0** 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 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  ge-7/0/0 {
    gigether-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

Confirm that the configuration is working properly.

- Verifying Chassis Cluster Redundant Ethernet Interfaces on page 92
- Verifying Chassis Cluster Control Links on page 92

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 | match reth1** command:

```
{primary:node0}
user@host> show interfaces | match reth1
ge-0/0/0.0              up    down aenet    --> reth1.0
ge-7/0/0.0              up    down aenet    --> reth0.0
reth1                   up    down
reth1.0                 up    down 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: Down

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

Fabric link status: Down

Fabric interfaces:
    Name    Child-interface    Status
                               (Physical/Monitored)
    fab0
    fab0

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    reth1          Up          1
```

Related Documentation

- *Understanding Chassis Cluster Redundant Ethernet Interfaces*

Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster

Supported Platforms SRX Series, vSRX

This example shows how to specify the number of redundant Ethernet interfaces for a chassis cluster. You must configure the redundant Ethernet interfaces count so that the redundant Ethernet interfaces that you configure are recognized.

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 for Branch SRX Series Devices* or "Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices" on page 56.

Overview

Before you configure redundant Ethernet interfaces for a chassis cluster, you must specify the number of redundant Ethernet interfaces for the chassis cluster.

In this example, you set the number of redundant Ethernet interfaces for a chassis cluster to 2.

Configuration

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
   ```

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

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

Verification

Verifying the Number of Redundant Ethernet Interfaces

Purpose Verify that the configuration is working properly.

Action To verify the configuration, enter the **show configuration chassis cluster** command.

Related Documentation

- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88

CHAPTER 11

Configuring SRX Series Chassis Cluster

Example: Enabling Eight Queue Class of Service on Redundant Ethernet Interfaces

Supported Platforms SRX5400, SRX5600, SRX5800

This example shows how to enable eight-queue CoS on redundant Ethernet interfaces on high-end SRX Series devices in a chassis cluster.

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 Services Gateways

Before you begin:

- Understand chassis cluster configuration. See "Example: Configuring an Active/Passive Chassis Cluster On a High-End SRX Series Services Gateway" on page 241.
- Understand chassis cluster redundant interface configuration. See "Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses" on page 88.

Overview

The SRX Series high-end devices 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.

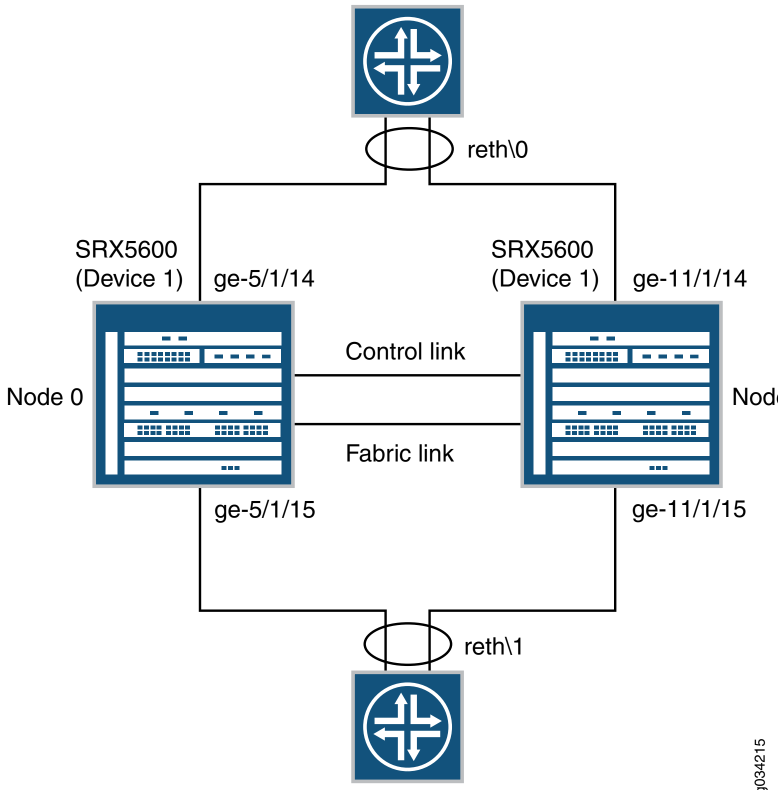
The following high-end SRX Series devices are supported:

- SRX5400
- SRX5600
- SRX5800

NOTE: On all high-end SRX Series devices, eight QoS queues are supported per ae interface.

Figure 10 on page 97 shows how to configure eight-queue CoS on redundant Ethernet interfaces on high-end SRX Series devices in a chassis cluster.

Figure 10: Eight-Queue CoS on Redundant Ethernet Interfaces



Configuration

CLI Quick Configuration

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

```
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
   ```

   NOTE: 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.

NOTE: 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

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

- Preparing Your Equipment for Chassis Cluster Formation on page 39
- SRX Series Chassis Cluster Configuration Overview on page 40
- Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 51
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47
- Verifying a Chassis Cluster Configuration on page 103

Verifying a Chassis Cluster Configuration

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- Verifying Chassis Cluster Statistics on page 104
- Clearing Chassis Cluster Statistics on page 105

Verifying Chassis Cluster Statistics

Supported Platforms SRX Series, vSRX

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                              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

{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
```

Related Documentation

- Verifying a Chassis Cluster Configuration on page 103
- Clearing Chassis Cluster Statistics on page 105

Clearing Chassis Cluster Statistics

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- Verifying a Chassis Cluster Configuration on page 103
- Verifying Chassis Cluster Statistics on page 104

PART 3

Managing Chassis Cluster Operations

CHAPTER 12

Configuring Chassis Cluster Dual Control Links for Managing Control Traffic

Understanding Chassis Cluster Dual Control Links

Supported Platforms SRX5600, SRX5800

Dual control links, where two pairs of control link interfaces are connected between each device in a cluster, are supported for the SRX5600 and SRX5800 Services Gateways. Having two control links helps to avoid a possible single point of failure.

For the SRX5600 and SRX5800 Services Gateways, this functionality requires a second Routing Engine, as well as a second Switch Control Board (SCB) to house the Routing Engine, to be installed on each device in the cluster. The purpose of the second Routing Engine is only to initialize the switch on the SCB.

NOTE: For the SRX5400 Services Gateways, dual control is not supported due to limited slots.

NOTE: Dual control link functionality is not supported on SRX4100 and SRX4200 devices.

NOTE: For the SRX5000 line, the second Routing Engine must be running Junos OS Release 10.0 or later.

The second Routing Engine, to be installed on SRX5000 line devices only, does not provide backup functionality. It does not need to be upgraded, even when there is a software upgrade of the master Routing Engine on the same node. Note the following conditions:

- You cannot run the CLI or enter configuration mode on the second Routing Engine.
- You do not need to set the chassis ID and cluster ID on the second Routing Engine.
- You need only a console connection to the second Routing Engine. (A console connection is not needed unless you want to check that the second Routing Engine booted up or to upgrade a software image.)
- You cannot log in to the second Routing Engine from the master Routing Engine.

NOTE: As long as the first Routing Engine is installed (even if it is rebooting or failing), the second Routing Engine cannot take over the chassis mastership; that is, it cannot control all the hardware on the chassis. The second Routing Engine can only become the master when the master Routing Engine is not present.

CAUTION: 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 master Routing Engine (RE). This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

Related Documentation

- Understanding Chassis Cluster Control Plane and Control Links on page 71
- Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 51
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 110
- Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 111

Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster

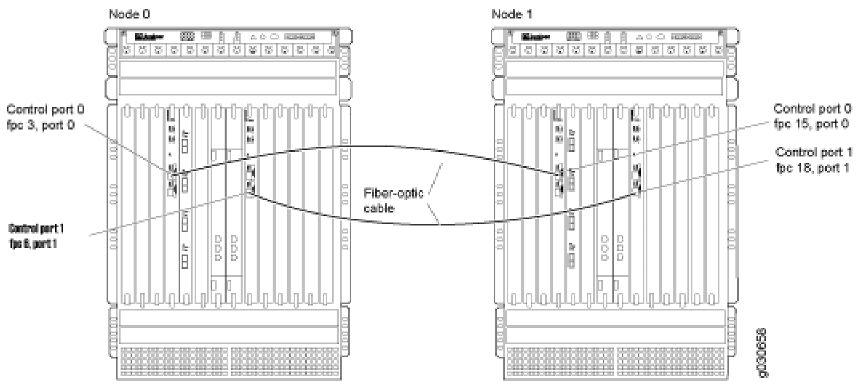
Supported Platforms SRX5600, SRX5800

For SRX5600 and SRX5800 devices, you can connect two control links between the two devices, effectively reducing the chance of control link failure.

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

For SRX5600 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 they be on two different SPCs to provide high availability. Figure 11 on page 111 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 11: Connecting Dual Control Links (SRX5800 Devices)



NOTE: For SRX5600 and SRX5800 devices, you must connect control port 0 on one node to control port 0 on the other node and, likewise, control port 1 to control port 1. If you connect control port 0 to control port 1, the nodes cannot receive heartbeat packets across the control links.

Related Documentation

- Understanding Chassis Cluster Control Plane and Control Links on page 71
- Understanding Chassis Cluster Dual Control Links on page 109
- Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 111

Example: Configuring Chassis Cluster Control Ports for Dual Control Links

Supported Platforms SRX5600, SRX5800

This example shows how to configure chassis cluster control ports for use as dual control links on SRX5600, and SRX5800 devices. You need to configure the control ports that you will use on each device to set up the control links.

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

- Requirements on page 112
- Overview on page 112
- Configuration on page 112
- Verification on page 113

Requirements

Before you begin:

- Understand chassis cluster control links. See "Understanding Chassis Cluster Control Plane and Control Links" on page 71.
- Physically connect the control ports on the devices. See "Connecting SRX Series Devices to Create a Chassis Cluster" on page 47.

Overview

By default, all control ports on SRX5600 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

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 From 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 done configuring the device, enter **commit** from configuration mode.

Verification

Verifying the Chassis Cluster Status

Purpose Verify the chassis cluster status.

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
```

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

Related Documentation

- Understanding Chassis Cluster Dual Control Links on page 109
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 110
- Verifying Chassis Cluster Control Plane Statistics on page 115
- Clearing Chassis Cluster Control Plane Statistics on page 116

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

Supported Platforms SRX5600, SRX5800, vSRX

For SRX5600 and SRX5800 devices, a second Routing Engine is required for each 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 10.0 or later.

NOTE: **You can use the show chassis hardware command to see the serial number and hardware version details of the second Routing Engine (RE1).**

NOTE: **For the SRX5400 Services Gateways, dual control is not supported due to limited slots.**

Because you cannot run the CLI or enter configuration mode on the second Routing Engine, you cannot upgrade the Junos OS image with the usual upgrade commands. Instead, use the master Routing Engine (RE0) to create a bootable USB storage device, which you can then use to install a software image on the second Routing Engine (RE1).

To upgrade the software image on the second Routing Engine (RE1):

1. Use FTP to copy the installation media into the /var/tmp directory of the master Routing Engine (RE0).
2. Insert a USB storage device into the USB port on the master Routing Engine (RE0).
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, install-media-srx5000-10.1R1-domestic.tgz.

The following code example can be used to write the image that you copied to the master Routing Engine (RE0) in step 1 onto the USB storage device:

```
dd if=install-media-srx5000-10.1R1-domestic.tgz 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 (RE1).
8. Move the console connection from the master Routing Engine (RE0) to the second Routing Engine (RE1), if you do not already have a connection.
9. Reboot the second Routing Engine (RE1). Issue the following command:

```
# 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?
```

Related Documentation

- Understanding Chassis Cluster Control Plane and Control Links on page 71
- Understanding Chassis Cluster Dual Control Links on page 109
- Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 111
- Verifying Chassis Cluster Control Plane Statistics on page 115

Verifying Chassis Cluster Control Plane Statistics

Supported Platforms SRX Series, vSRX

Purpose Display chassis cluster control plane statistics.

Action From 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
```

Related Documentation

- Clearing Chassis Cluster Control Plane Statistics on page 116

Clearing Chassis Cluster Control Plane Statistics

Supported Platforms SRX Series, vSRX

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

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

Cleared control-plane statistics
```

Related Documentation

- Verifying Chassis Cluster Control Plane Statistics on page 115

CHAPTER 13

Configuring Chassis Cluster Failover Parameters

Understanding Chassis Cluster Failover Parameters

Supported Platforms SRX5600, SRX5800, vSRX

Understanding Chassis Cluster Control Link Heartbeats

Junos OS transmits heartbeat signals over the control link at a configured interval. The system 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.

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, as the number of logical interfaces is scaled upward, the time before a failover is triggered needs to be increased accordingly. At maximum capacity on an 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.

NOTE: **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 liveliness 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.

> NOTE: 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.

> NOTE: 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.

Related Documentation

- Understanding Chassis Cluster Dual Control Links on page 109
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 110
- Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 111
- Example: Configuring Chassis Cluster Control Link Recovery on page 120

Example: Configuring Chassis Cluster Control Link Recovery

Supported Platforms SRX Series, vSRX

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 71.
- Understand chassis cluster dual control links. See “Understanding Chassis Cluster Dual Control Links” on page 109.
- Connect dual control links in a chassis cluster. See “Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster” on page 110.

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 30 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 on the node that was disabled when the control link failed. When the disabled 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

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
   ```


Verification

To verify the configuration is working properly, enter the **show configuration chassis cluster** command.

Related Documentation

- Understanding Chassis Cluster Failover Parameters on page 117
- Understanding Chassis Cluster Dual Control Links on page 109
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 110
- Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 111

CHAPTER 14

Configuring Chassis Cluster Dual Fabric Links to Increase Redundancy and Performance

Understanding Chassis Cluster Dual Fabric Links

Supported Platforms SRX Series, vSRX

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 devices, you can connect two fabric links between two devices, effectively reducing the chance of a fabric link failure.

In most SRX Series devices 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 must be the same type. For example, both must be Gigabit Ethernet interfaces or 10-Gigabit interfaces.

Related Documentation

- Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports on page 124
- Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports on page 126
- Example: Configuring the Chassis Cluster Fabric Interfaces on page 68

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

Supported Platforms SRX Series, vSRX

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 for Branch SRX Series Devices* or "Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices" on page 56.

Overview

In most SRX Series devices 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 8980 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**.

NOTE: 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

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

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
```

Related Documentation

- *Understanding Chassis Cluster Dual Fabric Links for Branch SRX Series*
- Understanding Chassis Cluster Dual Fabric Links for High-End SRX Series on page 123
- Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports on page 126
- Example: Configuring the Chassis Cluster Fabric Interfaces on page 68

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

Supported Platforms SRX Series, vSRX

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 for Branch SRX Series Devices* or "Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices" on page 56.

Overview

In most SRX Series devices 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 8980 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 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.

NOTE: **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

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

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

- *Understanding Chassis Cluster Dual Fabric Links for Branch SRX Series*
- Understanding Chassis Cluster Dual Fabric Links for High-End SRX Series on page 123
- Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports on page 124

Verifying Chassis Cluster Data Plane Interfaces

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- *Understanding Chassis Cluster Fabric Interfaces for Branch SRX Series*
- Understanding Chassis Cluster Fabric Interfaces for High-End SRX Series on page 63
- Example: Configuring the Chassis Cluster Fabric Interfaces on page 68
- Verifying Chassis Cluster Data Plane Statistics on page 129
- Clearing Chassis Cluster Data Plane Statistics on page 130

Verifying Chassis Cluster Data Plane Statistics

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- *Understanding Chassis Cluster Fabric Interfaces for Branch SRX Series*
- Understanding Chassis Cluster Fabric Interfaces for High-End SRX Series on page 63
- Example: Configuring the Chassis Cluster Fabric Interfaces on page 68
- Verifying Chassis Cluster Data Plane Interfaces on page 129
- Clearing Chassis Cluster Data Plane Statistics on page 130

Clearing Chassis Cluster Data Plane Statistics

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- *Understanding Chassis Cluster Fabric Interfaces for Branch SRX Series*
- Understanding Chassis Cluster Fabric Interfaces for High-End SRX Series on page 63
- Example: Configuring the Chassis Cluster Fabric Interfaces on page 68
- Verifying Chassis Cluster Data Plane Statistics on page 129
- Verifying Chassis Cluster Data Plane Interfaces on page 129

CHAPTER 15

Managing Chassis Cluster Redundancy Group Failover

Understanding Chassis Cluster Redundancy Group Failover

Supported Platforms SRX Series, vSRX

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 device 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. If a failure occurs on the new primary node during the hold-down interval, the system fails over immediately and overrides the hold-down interval.

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.

The hold-down interval affects manual failovers, as well as automatic failovers associated with monitoring failures.

On SRX Series devices, 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).

Related Documentation

- Example: Configuring Chassis Cluster Redundancy Groups on page 81
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 133
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 135
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 136

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

Supported Platforms SRX Series, vSRX

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 on page 132
- Overview on page 133
- Configuration on page 133

Requirements

Before you begin:

- Understand redundancy group failover. See "Understanding Chassis Cluster Redundancy Group Failover" on page 131.
- Understand redundancy group manual failover. See "Understanding Chassis Cluster Redundancy Group Manual Failover" on page 133.

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

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
   ```

Verification

Purpose

Verify that the configuration is working properly.

Action

To verify the configuration, enter the **show configuration chassis cluster** command.

Related Documentation

- Understanding Chassis Cluster Redundancy Groups on page 77
- Example: Configuring Chassis Cluster Redundancy Groups on page 81
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 133
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 135
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 136

Understanding Chassis Cluster Redundancy Group Manual Failover

Supported Platforms SRX Series, vSRX

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.

NOTE: If preempt is added to a redundancy group configuration, the device with the higher priority in the group can initiate a failover to become master. By default, preemption is disabled. For more information on preemeption, 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.

NOTE: Starting with Junos OS Release 12.1X46-D20 and Junos OS Release 12.1X47-D10, 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.

CAUTION: 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 master Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

NOTE: 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.

Related Documentation

- Understanding Chassis Cluster Redundancy Group Failover on page 131
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 136
- Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 132
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 135
- *Understanding Chassis Cluster Redundant Ethernet Interfaces for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces for High-End SRX Series Devices on page 85

Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover

Supported Platforms SRX Series, vSRX

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.

Related Documentation

- Understanding Chassis Cluster Redundancy Group Manual Failover on page 133
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 136
- Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 132
- *Understanding Chassis Cluster Redundant Ethernet Interfaces for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces for High-End SRX Series Devices on page 85

Initiating a Chassis Cluster Manual Redundancy Group Failover

Supported Platforms SRX Series, vSRX

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.

Before you begin, complete the following tasks:

- Example: Configuring Chassis Cluster Redundancy Groups on page 81
- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88
- Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 132

CAUTION: 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 master Routing Engine (RE). This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

NOTE: 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).

Related Documentation

- Understanding Chassis Cluster Redundancy Group Manual Failover on page 133
- Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 132
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 135
- *Understanding Chassis Cluster Redundant Ethernet Interfaces for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces for High-End SRX Series Devices on page 85

Verifying Chassis Cluster Failover Status

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 136
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 93
- Verifying a Chassis Cluster Configuration on page 103
- Verifying Chassis Cluster Statistics on page 104
- Clearing Chassis Cluster Failover Status on page 139

Clearing Chassis Cluster Failover Status

Supported Platforms SRX Series, vSRX

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
```

Related Documentation

- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 136

- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 93
- Verifying a Chassis Cluster Configuration on page 103
- Verifying Chassis Cluster Statistics on page 104
- Verifying Chassis Cluster Failover Status on page 138

CHAPTER 16

Configuring Redundant Ethernet LAG Interfaces for Increasing High Availability and Overall Throughput

Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

Supported Platforms SRX Series, vSRX

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. 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).

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.

NOTE: **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.

NOTE: **SRX Series devices interface-monitoring feature now 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.

NOTE: 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.

NOTE: For more information about Ethernet interface link aggregation and LACP, see the "Aggregated Ethernet" information in the *Interfaces Feature Guide for Security Devices*.

Related Documentation

- *Understanding Chassis Cluster Redundant Ethernet Interfaces for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces for High-End SRX Series Devices on page 85
- Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 153
- Example: Configuring Chassis Cluster Minimum Links on page 151
- Understanding Conditional Route Advertising in a Chassis Cluster on page 171

Understanding Chassis Cluster Redundant Ethernet Interface LAG Failover

Supported Platforms SRX Series, vSRX

To control failover of redundant Ethernet (reth) interfaces, it is important 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 falls to zero or below zero. This triggers a failover of the redundant Ethernet interfaces link aggregation group (LAG) once the number of physical links falls below the **minimum-links** value.

Consider a reth0 interface LAG with four underlying physical links and the **minimum-links** value set as 2. In this case, a failover is triggered only when the number of active physical links is less than 2.

NOTE:

- Interface-monitor and minimum-links values are used to monitor LAG link status and correctly calculate failover weight.
- The minimum-links value is used to keep the redundant Ethernet link status. However, to trigger a failover, interface-monitor must be set.

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
```

Set up interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.

The following scenarios provide examples of how to monitor redundant Ethernet LAG failover:

- Scenario 1: Monitored Interface Weight Is 255 on page 144
- Scenario 2: Monitored Interface Weight Is 75 on page 145
- Scenario 3: Monitored Interface Weight Is 100 on page 145

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**

In this case, although there are three active physical links and the redundant Ethernet LAG could have handled the traffic because of **minimum-links** configured, one physical link is still down, which triggers a failover based on the computed weight.

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 **minimum-links** configured. However, the failover will not happen, which in turn will result in traffic outage.

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 the three physical links are down, the redundant Ethernet interface will go down because of the **minimum-links** value. However, at the same time a failover would be triggered because of interface monitoring computed weights, ensuring that there is no traffic disruption.

Of all the three scenarios, scenario 3 illustrates the most ideal way to manage redundant Ethernet LAG failover.

Related Documentation

- *Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices on page 141

- Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 153
- Understanding LACP on Chassis Clusters on page 146
- Example: Configuring LACP on Chassis Clusters on page 148
- Example: Configuring Chassis Cluster Minimum Links on page 151

Understanding LACP on Chassis Clusters

Supported Platforms SRX Series

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 for Branch SRX Series Devices* or "Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices" on page 141.

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 151.

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 both redundant Ethernet interface active and standby links, a redundant Ethernet interface can be modeled to consist of two 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.

NOTE: 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 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 148.

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.

Related Documentation

- Example: Configuring LACP on Chassis Clusters on page 148

Example: Configuring LACP on Chassis Clusters

Supported Platforms SRX Series

This example shows how to configure LACP on chassis clusters.

Requirements

Before you begin:

- Add the aggregated Ethernet interfaces using the device count. See *Example: Configuring the Number of Aggregated Ethernet Interfaces on a Device*.
- Associate physical interfaces with the aggregated Ethernet Interfaces. See *Example: Associating Physical Interfaces with Aggregated Ethernet Interfaces*.
- Configure the aggregated Ethernet link speed. See *Example: Configuring Aggregated Ethernet Link Speed*.
- Configure the aggregated Ethernet minimum links speed. See *Example: Configuring Aggregated Ethernet Minimum Links*.
- Configure the LACP on standalone devices. See *Example: Configuring LACP on Standalone Devices*.

Overview

In this example, you set LACP to passive mode for the reth0 interface. 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.

Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see the *CLI User Guide*.

To configure LACP on chassis clusters:

1. Set the first LACP on primary node1.

   ```
   [edit interfaces]
   user@host# set reth0 redundant-ether-options lacp passive
   ```

2. Set the second LACP.

   ```
   [edit interfaces]
   user@host# set reth1 redundant-ether-options lacp active
   user@host# set reth1 redundant-ether-options lacp periodic slow
   ```

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

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


Verification

Verifying LACP on Redundant Ethernet Interfaces

Purpose Display LACP status information for redundant Ethernet interfaces.

Action From operational mode, enter the **show lacp interfaces reth0** command.

```
user@host> show lacp interfaces reth0
Aggregated interface: reth0
    LACP state:        Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      ge-11/0/0       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-11/0/0     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-11/0/1       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-11/0/1     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-11/0/2       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-11/0/2     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-11/0/3       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-11/0/3     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/0        Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/0      Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/1        Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/1      Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/2        Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/2      Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/3        Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-3/0/3      Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
    LACP protocol:        Receive State  Transmit State          Mux State
      ge-11/0/0                 Current   Fast periodic Collecting distributing
      ge-11/0/1                 Current   Fast periodic Collecting distributing
      ge-11/0/2                 Current   Fast periodic Collecting distributing
      ge-11/0/3                 Current   Fast periodic Collecting distributing
      ge-3/0/0                  Current   Fast periodic Collecting distributing
      ge-3/0/1                  Current   Fast periodic Collecting distributing
      ge-3/0/2                  Current   Fast periodic Collecting distributing
      ge-3/0/3                  Current   Fast periodic Collecting distributing
{primary:node1}
```

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.

Related Documentation

- Understanding LACP on Chassis Clusters on page 146
- *Verifying LACP on Redundant Ethernet Interfaces*

Example: Configuring Chassis Cluster Minimum Links

Supported Platforms SRX Series, vSRX

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 for IPv4 and IPv6 Addresses" on page 88.
- Understand redundant Ethernet interface link aggregation groups. See "Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups" on page 153.

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.

NOTE: **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

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

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: Sendbcast-pkt-to-re
```

Related Documentation

- *Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups Branch SRX Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices on page 141
- Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 153
- Understanding Conditional Route Advertising in a Chassis Cluster on page 171

Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

Supported Platforms SRX Series, vSRX

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 for IPv4 and IPv6 Addresses” on page 88.
- Understand chassis cluster redundant Ethernet interface link aggregation groups. See *Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for Branch SRX Series Devices* or “Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices” on page 141.

Overview

NOTE: 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 devices. In cases where LACP is not available on the switch, you must not enable LACP on SRX Series devices.

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

NOTE: 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.

NOTE: Junos OS supports LACP and LAG on a redundant Ethernet interface, which is called RLAG.

Configuration

CLI Quick Configuration

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

```
{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

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
```


Related Documentation

- *Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices on page 141
- Understanding Chassis Cluster Redundant Ethernet Interface LAG Failover on page 143
- Understanding LACP on Chassis Clusters on page 146
- Example: Configuring LACP on Chassis Clusters on page 148
- Example: Configuring Chassis Cluster Minimum Links on page 151

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

Supported Platforms SRX5400, SRX5600, SRX5800

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 on page 156
- Overview on page 156
- Configuration on page 157
- Verification on page 159

Requirements

This example uses the following software and hardware components:

- Junos OS Release 15.1X49-D40 or later for SRX Series devices.
- 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 device 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.

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

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 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 gigether-options redundant-parent reth0
set interfaces xe-3/0/0 gigether-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
```

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 *CLI User Guide*.

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

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.

Related Documentation

- Understanding LACP on Chassis Clusters on page 146
- *Verifying LACP on Redundant Ethernet Interfaces*

CHAPTER 17

Simplifying Chassis Cluster Management

Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes

Supported Platforms SRX Series, vSRX

When you set up an SRX Series chassis cluster, the SRX Series devices 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.

NOTE: 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.

NOTE: 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.

Related Documentation

- Verifying Chassis Cluster Configuration Synchronization Status on page 162
- NTP Time Synchronization on SRX Series Devices on page 163
- Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 163

Verifying Chassis Cluster Configuration Synchronization Status

Supported Platforms SRX Series, vSRX

Purpose Display the configuration synchronization status of a chassis cluster.

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
```

Related Documentation

- Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 161
- NTP Time Synchronization on SRX Series Devices on page 163
- Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 163
- **show chassis cluster information configuration-synchronization on page 398**

NTP Time Synchronization on SRX Series Devices

Supported Platforms SRX Series, vSRX

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.

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.

Related Documentation

- Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 163

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

Supported Platforms SRX Series, vSRX

This example shows how to simplify management by synchronizing the time between two SRX Series devices 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 devices in a chassis cluster in order to manage the following items:

- RTO
- Licenses
- Software updates
- Node failovers
- Analyzing system logs (syslogs)

Requirements

This example uses the following hardware and software components:

- SRX Series devices 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.

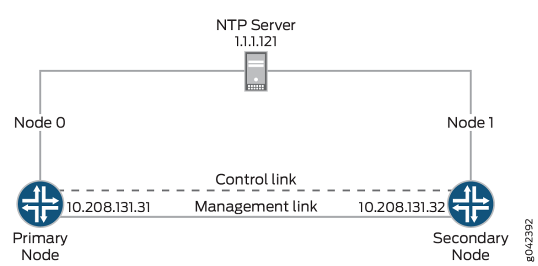
Overview

When SRX Series devices 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 12 on page 164 shows the time synchronization from the peer node using the control link.

Figure 12: 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

- Synchronizing Time from the NTP server on page 165
- Results on page 165

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

Confirm that the configuration is working properly.

- Verifying the NTP Configuration on the Primary Node on page 165
- Verifying the NTP Configuration on the Secondary Node on page 167

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 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 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
```

Meaning The output on the 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 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.

Related Documentation

- Time Management Routing Guide for Administration Devices
- NTP Time Synchronization on SRX Series Devices on page 163
- Verifying Chassis Cluster Configuration Synchronization Status on page 162

CHAPTER 18

Configuring Route Advertisement over Redundant Ethernet Interfaces in a Chassis Cluster

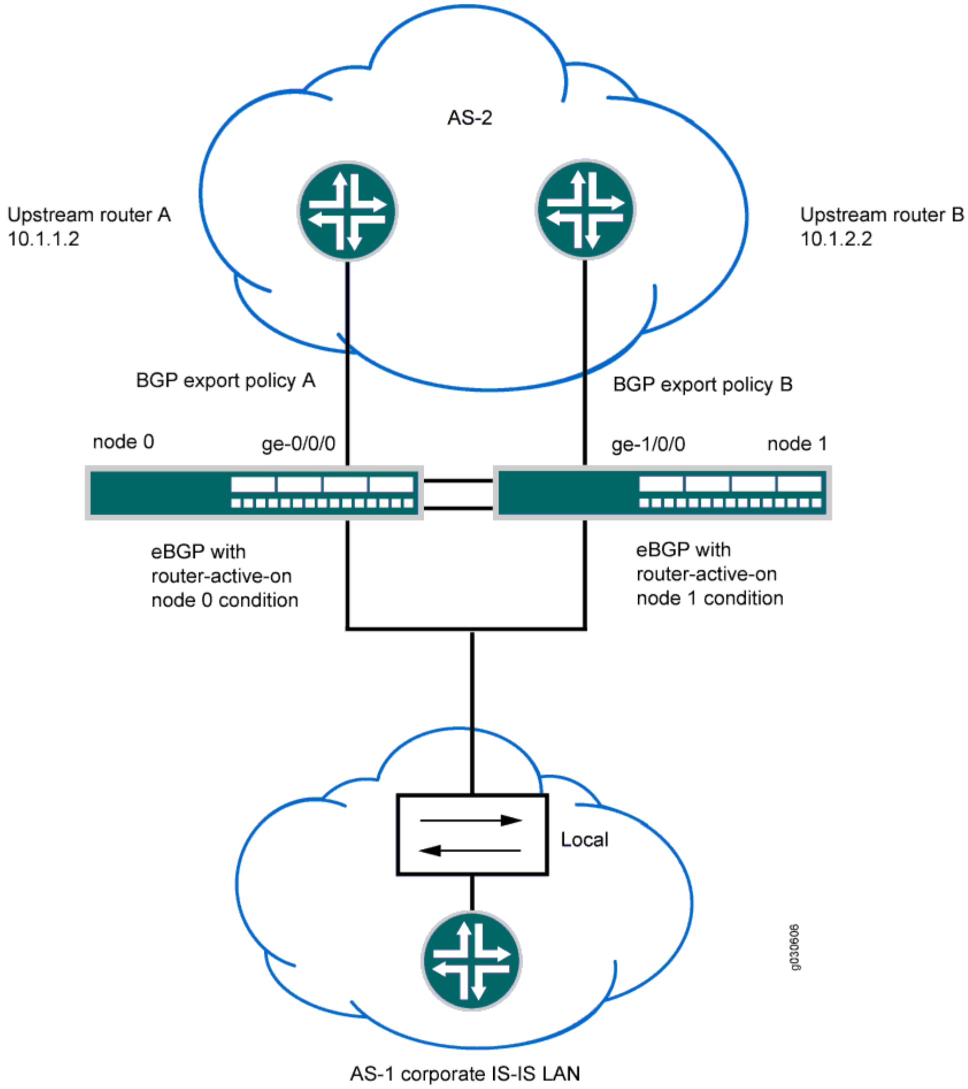
Understanding Conditional Route Advertising in a Chassis Cluster

Supported Platforms SRX Series, vSRX

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. Figure 13 on page 172 shows a typical scenario, with a redundant Ethernet interface connecting the corporate LAN, through a chassis cluster, to an external network segment.

Figure 13: Conditional Route Advertising



Related Documentation

- Example: Configuring Conditional Route Advertising in a Chassis Cluster on page 172
- Verifying a Chassis Cluster Configuration on page 103
- Verifying Chassis Cluster Statistics on page 104

Example: Configuring Conditional Route Advertising in a Chassis Cluster

Supported Platforms SRX Series, vSRX

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 on page 173
- Overview on page 173
- Configuration on page 175

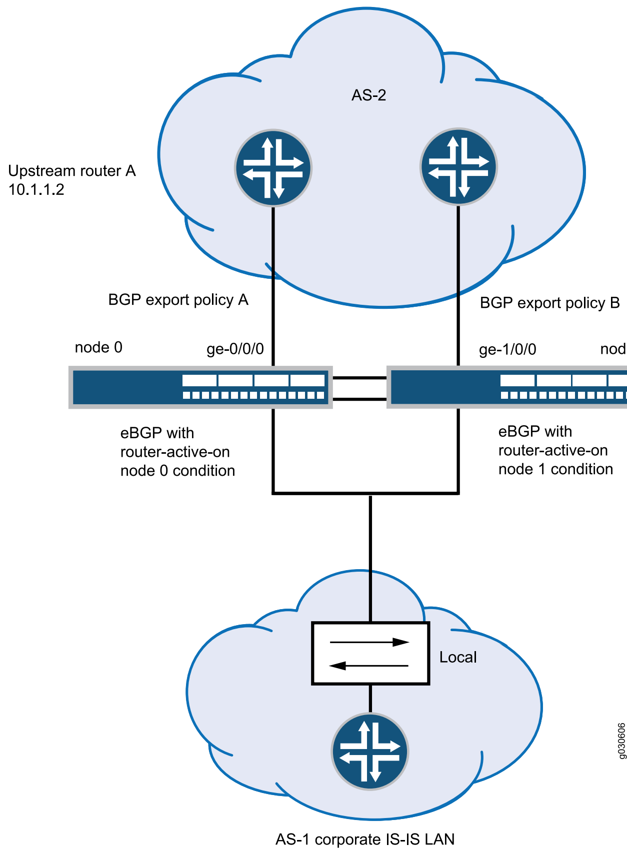
Requirements

Before you begin, understand conditional route advertising in a chassis cluster. See “Understanding Conditional Route Advertising in a Chassis Cluster” on page 171.

Overview

As illustrated in Figure 14 on page 174, 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 t1-1/0/0 and one from t1-1/0/1 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 14: Conditional Route Advertising



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

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 policies.

  ```
  {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.

Related Documentation

- Understanding Conditional Route Advertising in a Chassis Cluster on page 171
- Verifying a Chassis Cluster Configuration on page 103
- Verifying Chassis Cluster Statistics on page 104

CHAPTER 19

Monitoring Chassis Cluster Setup

Understanding Chassis Cluster Redundancy Group Interface Monitoring

Supported Platforms SRX Series, vSRX

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 below commands:

- show chassis cluster information
- show chassis cluster interfaces

NOTE: We do not recommend configuring data plane modules such as interface monitoring and IP monitoring on Redundancy Group 0 (RG0) for SRX Series devices in a chassis cluster.

CAUTION: 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 master 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.

NOTE:

- 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 (RG0), the interface monitoring on the RG0 secondary is disabled for 30 seconds. This prevents failover of other redundancy groups along with RG0 failover.

Related Documentation

- Example: Configuring Chassis Cluster Interface Monitoring on page 178
- Understanding Chassis Cluster Redundancy Groups on page 77
- Example: Configuring Chassis Cluster Redundancy Groups on page 81

Example: Configuring Chassis Cluster Interface Monitoring

Supported Platforms SRX Series, vSRX

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 on page 179
- Overview on page 179
- Configuration on page 180
- Verification on page 184

Requirements

Before you begin, create a redundancy group. See "Example: Configuring Chassis Cluster Redundancy Groups" on page 81.

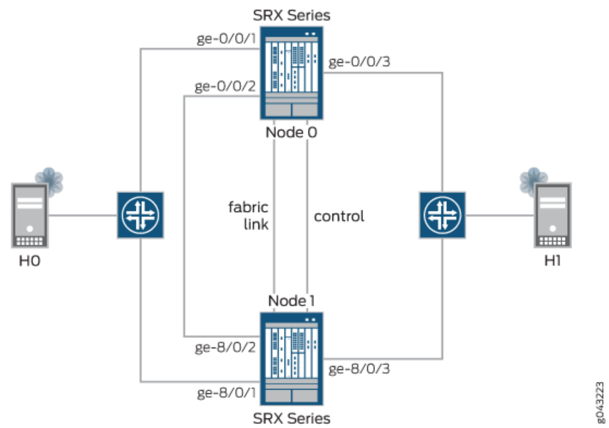
Overview

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 15 on page 180 illustrates the network topology used in this example.

Figure 15: SRX Series Chassis Cluster Interface Monitoring Topology Example



Configuration

CLI Quick Configuration

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

```
set chassis cluster traceoptions flag all
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
```


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 traceoptions for chassis cluster.

   [edit chassis cluster]
   user@host# **set traceoptions flag all**

2. Specify the number of redundant Ethernet interfaces.

   [edit chassis cluster]
   user@host# **set reth-count 3**

3. 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**

4. Set up interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.

   NOTE: We do not recommend interface monitoring for RG0, 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**

   NOTE: Interface failover only occurs after the weight reaches zero.

5. 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 {
    traceoptions {
      flag all;
    }
    reth-count 3;
    node 0; ## Warning: 'node' is deprecated
    node 1; ## Warning: 'node' is deprecated
    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 {
  gigether-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

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 Interfaces After Enabling Interface ge-0/0/3 on page 203
- Verifying Chassis Cluster Information After Enabling Interface ge-0/0/3 on page 204

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.

NOTE: 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.

NOTE: 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.

NOTE: 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.

NOTE: 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.

Related Documentation

- Example: Configuring Chassis Cluster Redundancy Groups on page 81
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 177
- *Understanding Chassis Cluster Redundancy Group IP Address Monitoring for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring for High-End SRX Series Devices on page 205
- Understanding Chassis Cluster Redundancy Group Failover on page 131
- Understanding Chassis Cluster Redundancy Groups on page 77
- *Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for Branch SRX Series Devices*
- Understanding SRX Series Chassis Cluster Slot Numbering, Physical Port and Logical Interface Naming for High-End SRX Series Devices on page 51

Understanding Chassis Cluster Redundancy Group IP Address Monitoring

Supported Platforms SRX Series, vSRX

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.

NOTE: **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 must be monitored are router gateway addresses to ensure that valid traffic coming into the services gateway can be forwarded to the appropriate network router.

NOTE: **Starting in Junos OS Release 12.1X46-D35, for all SRX Series devices, the redundant Ethernet 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.

NOTE: 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.

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 devices.

Starting in Junos OS Release 15.1X49-D60, 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.

Related Documentation

- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 177
- Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 208
- Understanding Chassis Cluster Redundancy Group Failover on page 131
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 211

Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring

Supported Platforms SRX Series, vSRX

This example shows how to configure redundancy group IP address monitoring for an SRX Series device in 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 for Branch SRX Series Devices or Example: Setting the Chassis Cluster Node ID and Cluster ID.
- Configure the chassis cluster management interface. See Example: Configuring the Chassis Cluster Management Interface.
- Configure the chassis cluster fabric. See Example: Configuring the Chassis Cluster Fabric Interfaces.

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

NOTE: **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—150
- Redundant Ethernet interface—reth1.0
- Secondary IP address—10.1.1.101

Configuration

CLI Quick Configuration

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

```
{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 150
  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

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       220
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       220
10.1.1.101                 reachable       0            n/a       100
```

Related Documentation

- Understanding Chassis Cluster Redundancy Group Interface Monitoring
- *Understanding Chassis Cluster Redundancy Group IP Address Monitoring for Branch SRX Series Devices*
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring for High-End SRX Series Devices on page 205
- Understanding Chassis Cluster Redundancy Group Failover on page 131

Understanding Chassis Cluster Monitoring of Global-Level Objects

Supported Platforms SRX5400, SRX5600, SRX5800, vSRX

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 devices have a flow-based forwarding process, *flowd*, which forwards packets through the device.

- Understanding SPU Monitoring on page 212
- Understanding flowd Monitoring on page 213
- Understanding Cold-Sync Monitoring on page 213

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.

NOTE: **SPU monitoring is supported on high-end SRX Series 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.

NOTE: **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.

NOTE: **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 RG0 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 mastership 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 mastership.

Understanding Cold-Sync Monitoring with SPU Replacement or Expansion

If your SRX5600 or SRX5800 Services Gateway is part of a chassis cluster, when you replace a Services Processing Card (SPC) with a SPC2 on the device, you must fail over all redundancy groups to one node.

NOTE: For SRX5400 devices, only SPC2 is supported.

The following events take place during this scenario:

- When the SPC2 is to be 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.

Related Documentation

- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 177
- Example: Configuring Chassis Cluster Interface Monitoring on page 178
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 205
- Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 208

IP Monitoring Overview

Supported Platforms SRX Series

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 devices, 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.

NOTE: **We do not recommend configuring chassis cluster IP monitoring on Redundancy Group 0 (RG0) for SRX Series devices.**

Table 11 on page 215 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 11: 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

Table 11: IP Monitoring Results and Failover Action *(continued)*

Primary Node Monitored Status	Secondary Node Monitored Status	Failover Action
Unreachable	Unreachable	No action

NOTE:

- You can configure up to 64 IP addresses for IP monitoring on SRX5000 line devices.
- 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 12 on page 216 provides details on multiple interface combinations of IOC2 and IOC3 with maximum MAC numbers.

Table 12: Maximum MACs Supported for IP Monitoring on IOC2 and IOC3

Cards	Interfaces	Maximum MACs Supported for IP Monitoring
IOC2 (SRX5K-MPC)	10XGE	10
	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.

Related Documentation

- SRX5400, SRX5600, and SRX5800 Services Gateway Card Overview
- Example: Configuring IP Monitoring on SRX5000 Line Devices for IOC2 and IOC3 on page 217

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

Supported Platforms SRX5400, SRX5600, SRX5800

This example shows how to monitor IP address on a 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 are also applicable to IOC3 also.

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

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

- Configuring IP Monitoring on a 10x10GE SFP+ MIC on page 219

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 1203.0.113.1
weight 80
set chassis cluster redundancy-group 1 ip-monitoring family inet 203.0.113.1
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 gigether-options redundant-parent reth0
user@host# set interfaces xe-1/2/2 gigether-options redundant-parent reth2
user@host# set interfaces xe-1/2/3 gigether-options redundant-parent reth1
user@host# set interfaces xe-4/2/1 gigether-options redundant-parent reth0
user@host# set interfaces xe-4/2/2 gigether-options redundant-parent reth2
user@host# set interfaces xe-4/2/3 gigether-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 198.51.100.2;
                        }
                        203.0.113.2 {
                            weight 80;
                            interface reth2.0 secondary-ip-address 203.0.113.2;
                        }
                    }
                }
            }
        }
    }
}
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 {
        gigether-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

Confirm the configuration is working properly.

- Verifying IP Monitoring Status on page 222

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.

Related Documentation

- IP Monitoring Overview on page 215
- Example: Configuring an Active/Passive Chassis Cluster On a High-End SRX Series Services Gateway on page 241

PART 4

Additional Chassis Cluster Configurations

CHAPTER 20

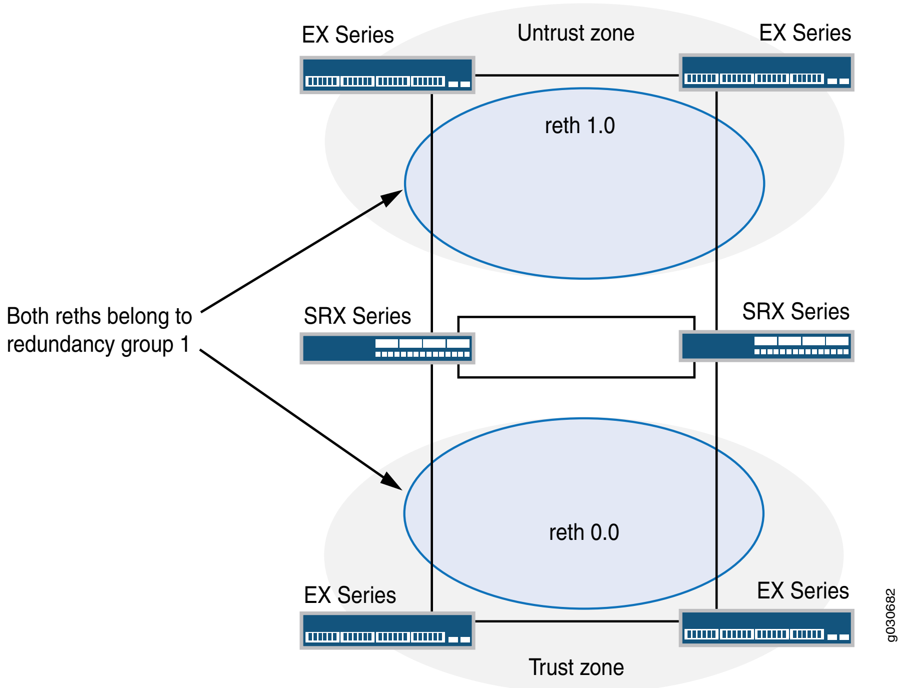
Configuring Active/Passive Chassis Cluster Deployments

Understanding Active/Passive Chassis Cluster Deployment

Supported Platforms SRX Series, vSRX

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 16 on page 228). When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 16: 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.

Related Documentation

- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 228
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 239

Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)

Supported Platforms SRX Series, vSRX

This example shows how to configure active/passive chassis clustering for devices.

- Requirements on page 228
- Overview on page 229
- Configuration on page 231
- Verification on page 236

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**

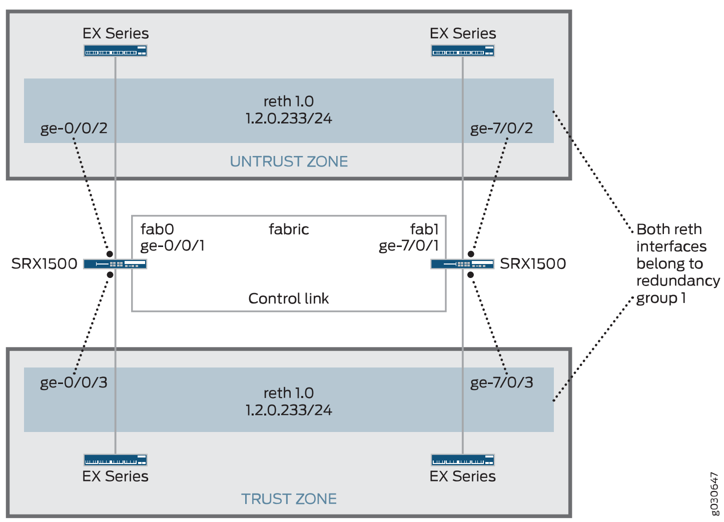
5. Connect to the other device using the console port and set the cluster ID and node number.

   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 17 on page 229.) When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 17: Active/Passive Chassis Cluster Topology



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 13 on page 230 through Table 16 on page 231.

Table 13: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters

Groups	node0	• Hostname: srx1500-A
		• Interface: fxp0
		  ◦ Unit 0
		  ◦ 192.0.2.110/24

	node1	• Hostname: srx1500-B
		• Interface: fxp0
		  ◦ Unit 0
		  ◦ 192.0.2.111/24

Table 14: 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

Redundancy group	0	• Priority:
		  ◦ Node 0: 254
		  ◦ Node 1: 1

	1	• Priority:
		  ◦ Node 0: 254
		  ◦ Node 1: 1

		Interface monitoring • ge-0/0/4 • ge-7/0/4 • ge-0/0/5 • ge-7/0/5
Number of redundant Ethernet interfaces	–	2
Interfaces	ge-0/0/4	Redundant parent: reth1
	ge-7/0/4	Redundant parent: reth1
	ge-0/0/5	Redundant parent: reth0
	ge-7/0/5	Redundant parent: reth0

Table 14: Chassis Cluster Configuration Parameters *(continued)*

Feature	Name	Configuration Parameters
	reth0	Redundancy group: 1
		• Unit 0 • 198.51.100.1/24
	reth1	Redundancy group: 1
		• Unit 0 • 203.0.113.233/24

Table 15: 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 16: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

This security policy permits traffic from the trust zone to the untrust zone.	ANY	• Match criteria:
		  ◦ source-address any
		  ◦ destination-address any
		  ◦ application any
		• Action: permit

Configuration

CLI Quick Configuration

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

```
[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 gigether-options redundant-parent reth1
user@host# set interfaces ge-7/0/5 gigether-options redundant-parent reth1
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 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 {
         gigether–options {
            redundant–parent reth0;
         }
      }
      ge–7/0/4{
         gigether–options {
            redundant–parent reth0;
         }
      }
      ge–0/0/5 {
         gigether–options {
            redundant–parent reth1;
         }
      }
```



```
    ge–7/0/5 {
      gigether–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

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 name: fxp1

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

Related Documentation

- Understanding Active/Passive Chassis Cluster Deployment on page 227
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 239

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

Supported Platforms SRX Series, vSRX

1. Enable clustering. See Step 1 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 228.
2. Configure the management interface. See Step 2 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 228.
3. Configure the fabric interface. See Step 3 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 228.
4. Configure the redundancy groups.
   - Select **Configure>Chassis Cluster**.
   - Enter the following information, and then click **Apply**:

     Redundant ether-Interface Count: **2**

     Heartbeat Interval: **1000**

Heartbeat Threshold: **3**

Nodes: **0**

Group Number: **0**

Priorities: **100**

- Enter the following information, and then click **Apply**:

  Nodes: **0**

  Group Number: **1**

  Priorities: **1**

- Enter the following information, and then click **Apply**:

  Nodes: **1**

  Group Number: **0**

  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.
   - Click **Apply**.
   - Select **ge-0/0/5**.
   - Enter **reth0** in the Redundant Parent box.
   - Click **Apply**.
   - Select **ge-7/0/5**.
   - Enter **reth0** in the Redundant Parent box.
   - Click **Apply**.
   - See Step 5 in "Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)" on page 228 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 228.

7. Configure the security policies. See Step 7 in "Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)" on page 228.

8. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options**>**Commit**.

Related Documentation

- Understanding Active/Passive Chassis Cluster Deployment on page 227
- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 228

Example: Configuring an Active/Passive Chassis Cluster On a High-End SRX Series Services Gateway

Supported Platforms SRX5400, SRX5600, SRX5800

This example shows how to set up basic active/passive chassis clustering on a high-end SRX Series device (SRX5800 device).

Requirements

Before you begin:

- You need two SRX5800 Services Gateways with identical hardware configurations, one MX240 edge router, and one EX8208 Ethernet Switch.
- 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.

NOTE: **Control port configuration is required for SRX5400, SRX5600, and SRX5800 devices.**

- To ensure secure login, configure the internal IPsec SA. When the internal IPsec is configured, IPsec-based rlogin and remote command (rcmd) are enforced, so an attacker cannot gain privileged access or observe traffic containing administrator commands and outputs. You do not need to configure the internal IPsec on both the nodes. When you commit the configuration, both nodes are synchronized. Only 3des-cbc encryption algorithm is supported. You must ensure that the manual encryption key is ascii text and 24 characters long; otherwise, the configuration will result in a commit failure.

You have the option to enable the iked-encryption. The device must be rebooted after this option is configured.

- Enable the iked-encryption:

  user@host# **set security ipsec internal security-association manual encryption ike-ha-link-encryption enable**

- Enable the 3des-cbc encryption algorithm:

  user@host# **set security ipsec internal security-association manual encryption algorithm 3des-cbc**

- Configure the encryption key:

  user@host# **set security ipsec internal security-association manual encryption key ascii-text "$ABC123"**

- Activate internal IPsec:

  user@host> **request security internal-security-association refresh**

- Use the **show chassis cluster interfaces** CLI command to verify that internal SA is enabled:

```
user@host> show chassis cluster interfaces
  Control link status: Up

  Control interfaces:
      Index   Interface        Status       Internal SA <- new column
      0       em0              Up             enabled
      1       em1              Down           enabled
```

- Configure the control port for each device, and commit the configuration.

  Select FPC 1/13, because the central point 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. For example:

  - On node 0:

    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**

  - On node 1:

    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**

- 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@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.

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 a high-end SRX Series device. The basic active/passive example is the most common type of chassis cluster.

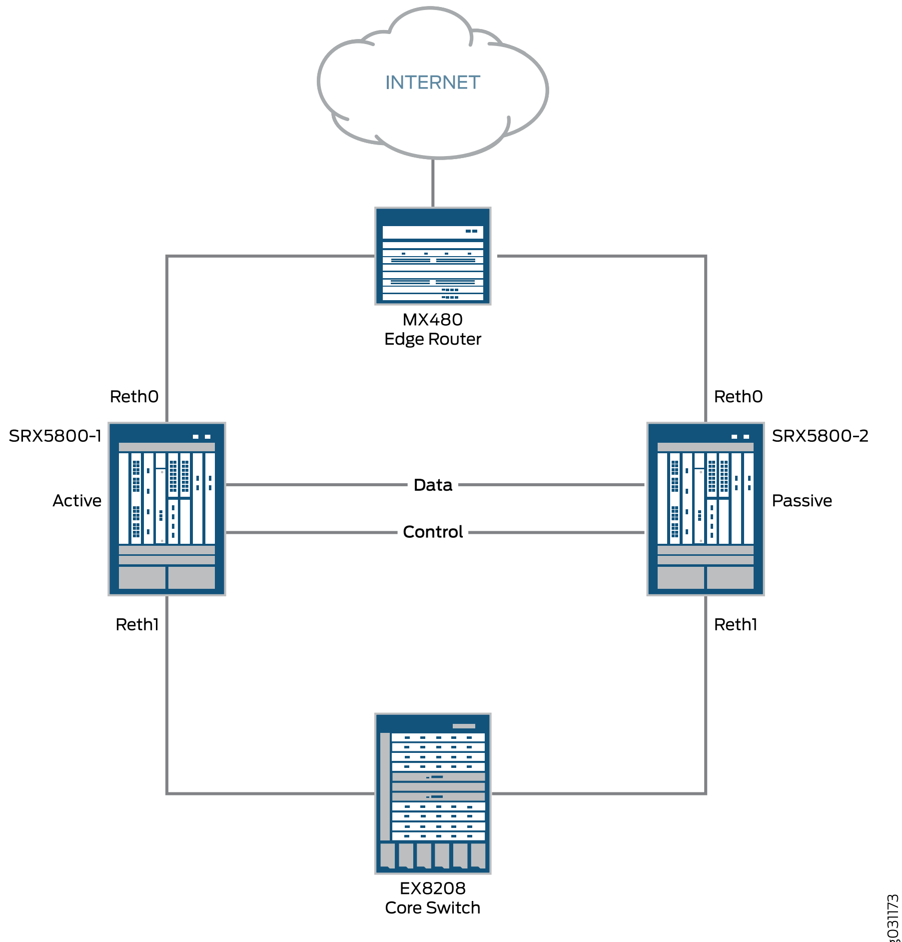
The basic active/passive chassis cluster consists of two devices:

- 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.

NOTE: This active/passive mode example for the SRX5800 Services Gateway 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. See *Introduction to NAT*, *Security Policies Overview*, and *IPsec VPN Overview*. 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 (CLI Procedure)*. You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800 Services Gateway. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

Figure 18 on page 244 shows the topology used in this example.

Figure 18: Basic Active/Passive Chassis Clustering on a High-End SRX Series Device Topology Example



Configuration

CLI Quick Configuration

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

On {primary:node0}

```
[edit]
set interfaces fab0 fabric-options member-interfaces ge-11/3/0
set interfaces fab1 fabric-options member-interfaces ge-23/3/0
set groups node0 system host-name SRX5800-1
set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24
set groups node0 system backup-router 10.3.5.254 destination 10.0.0.0/16
set groups node1 system host-name SRX5800-2
set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24
set groups node1 system backup-router 10.3.5.254 destination 10.0.0.0/16
set apply-groups "${node}"
```



```
set chassis cluster reth-count 2
set chassis cluster redundancy-group 0 node 0 priority 129
set chassis cluster redundancy-group 0 node 1 priority 128
set chassis cluster redundancy-group 1 node 0 priority 129
set chassis cluster redundancy-group 1 node 1 priority 128
set interfaces xe-6/0/0 gigether-options redundant-parent reth0
set interfaces xe-6/1/0 gigether-options redundant-parent reth1
set interfaces xe-18/0/0 gigether-options redundant-parent reth0
set interfaces xe-18/1/0 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 1.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 2.2.2.1/24
set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 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-18/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
set chassis cluster control-link-recovery
set security zones security-zone untrust interfaces reth0.0
set security zones security-zone trust interfaces reth1.0
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
set routing-options static route 2.0.0.0/8 next-hop 2.2.2.254
```

To quickly configure an EX8208 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 {primary:node0}

```
[edit]
set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode access vlan members
  SRX5800
set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode access vlan members
  SRX5800
set interfaces vlan unit 50 family inet address 2.2.2.254/24
set vlans SRX5800 vlan-id 50
set vlans SRX5800 l3-interface vlan.50
set routing-options static route 0.0.0.0/0 next-hop 2.2.2.1/24
```

To quickly configure an MX240 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 {primary:node0}

```
[edit]
set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family ethernet-switching
set interfaces xe-2/0/0 encapsulation ethernet-bridge unit 0 family ethernet-switching
set interfaces irb unit 0 family inet address 1.1.1.254/24
set routing-options static route 2.0.0.0/8 next-hop 1.1.1.1
set routing-options static route 0.0.0.0/0 next-hop (upstream router)
set vlans SRX5800 vlan-id X (could be set to “none”)
set vlans SRX5800 domain-type bridge routing-interface irb.0
set vlans SRX5800 domain-type bridge interface xe-1/0/0
```

set vlans SRX5800 domain-type bridge interface xe-2/0/0

Step-by-Step Procedure

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

To configure a chassis cluster on a high-end SRX Series device:

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 from which device the command is configured.

1. 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 1-Gigabit Ethernet ports because running out of bandwidth using active/passive mode is not an issue. Define two fabric interfaces, one on each chassis, to connect together.

   ```
   {primary:node0}[edit]
   user@host# set interfaces fab0 fabric-options member-interfaces ge-11/3/0
   user@host# set interfaces fab1 fabric-options member-interfaces ge-23/3/0
   ```

2. Because the SRX5800 Services Gateway 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 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.

   ```
   {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 10.3.5.1/24
   user@host# set groups node0 system backup-router 10.3.5.254 destination
     0.0.0.0/16
   user@host# set groups node1 system host-name SRX5800-2
   user@host# set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24
   user@host# set groups node1 system backup-router 10.3.5.254 destination
     0.0.0.0/16
   user@host# 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 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.

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 (z line mode traffic).
- On all high-end SRX Series devices, the IPsec VPN is not supported in active/active chassis cluster configuration (that is, when there are multiple RG1+ redundancy groups).

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
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
```

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 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 neighborship 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@host# set interfaces xe-6/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-6/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-18/0/0 gigether-options redundant-parent reth0
```

```
user@host# set interfaces xe-18/1/0 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 1.1.1.1/24
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 2.2.2.1/24
```

5. Configure the chassis cluster behavior in case of a failure. For the SRX5800 Services Gateway, 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@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0
  weight 255
user@host# set chassis cluster redundancy-group 1 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 1 interface-monitor xe-18/1/0
  weight 255
user@host# set chassis cluster control-link-recovery
```

This step completes the chassis cluster configuration part of the active/passive mode example for the SRX5800 Services Gateway. The rest of this procedure describes how to configure the zone, virtual router, routing, EX8208 Core Switch, and MX240 Edge Router to complete the deployment scenario.

6. 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@host# set security zones security-zone untrust interfaces reth0.0
user@host# set security zones security-zone trust interfaces reth1.0
```

7. For this active/passive mode example, because of the simple network architecture, use static routes to define how to route to the other network devices.

```
{primary:node0}[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
user@host# set routing-options static route 2.0.0.0/8 next-hop 2.2.2.254
```

8. For the EX8208 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 Services Gateway; most notably the VLANs, routing, and interface configuration.

```
{primary:node0}[edit]
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode
  access vlan members SRX5800
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode
  access vlan members SRX5800
user@host# set interfaces vlan unit 50 family inet address 2.2.2.254/24
user@host# set vlans SRX5800 vlan-id 50
user@host# set vlans SRX5800 l3-interface vlan.50
user@host# set routing-options static route 0.0.0.0/0 next-hop 2.2.2.1/24
```

9. For the MX240 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 Services Gateway; most notably you must use an IRB interface within a virtual switch instance on the switch.

   ```
   {primary:node0}[edit]
   user@host# set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family
     ethernet-switching
   user@host# set interfaces xe-2/0/0 encapsulation ethernet-bridge unit 0 family
     ethernet-switching
   user@host# set interfaces irb unit 0 family inet address 1.1.1.254/24
   user@host# set routing-options static route 2.0.0.0/8 next-hop 1.1.1.1
   user@host# set routing-options static route 0.0.0.0/0 next-hop (upstream router)
   user@host# set vlans SRX5800 vlan-id X (could be set to “none”)
   user@host# set vlans SRX5800 domain-type bridge routing-interface irb.0
   user@host# set vlans SRX5800 domain-type bridge interface xe-1/0/0
   user@host# set vlans SRX5800 domain-type bridge interface xe-2/0/0
   ```

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.

```
> show configuration
version x.xx.x;
groups {
    node0 {
        system {
            host-name SRX58001;
            backup-router 10.3.5.254 destination 0.0.0.0/16;

        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.3.5.1/24;
                    }
                }
            }
        }
    }
    node1 {
        system {
            host-name SRX58002;
            backup-router 10.3.5.254 destination 0.0.0.0/16;

        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.3.5.2/24;
                    }
                }
            }
        }
    }
}
```

```
apply-groups "${node}";
system {
        root-authentication {
        encrypted-password "$ABC123.";
    }
    name-server {
        4.2.2.2;
    }
    services {
        ssh {
            root-login allow;
        }
        netconf {
            ssh;
        }
        web-management {
            http {
                interface fxp0.0;
            }
        }
    }
}
chassis {
    cluster {
        control-link-recovery;
        reth-count 2;
        control-ports {
            fpc 1 port 0;
            fpc 13 port 0;
        }
        redundancy-group 0 {
            node 0 priority 129;
            node 1 priority 128;
        }
        redundancy-group 1 {
            node 0 priority 129;
            node 1 priority 128;
            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-6/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-6/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    xe-18/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
```



```
        }
        xe-18/1/0 {
            gigether-options {
                redundant-parent reth1;
            }
        }
        fab0 {
            fabric-options {
                member-interfaces {
                    ge-11/3/0;
                }
            }
        }
        fab1 {
            fabric-options {
                member-interfaces {
                    ge-23/3/0;
                }
            }
        }
        reth0 {
            redundant-ether-options {
                redundancy-group 1;
            }
            unit 0 {
                family inet {
                    address 1.1.1.1/24;
                }
            }
        }
        reth1 {
            redundant-ether-options {
                redundancy-group 1;
            }
            unit 0 {
                family inet {
                    address 2.2.2.1/24;
                }
            }
        }
    }
    routing-options {
        static {
            route 0.0.0.0/0 {
                next-hop 1.1.1.254;
        }
            route 2.0.0.0/8 {
                next-hop 2.2.2.254;
            }
        }
    }
    security {
        zones {
            security-zone trust {
                host-inbound-traffic {
                    system-services {
                        all;
                    }
                }
                interfaces {
                    reth0.0;
```

```
            }
        }
        security-zone untrust {
            interfaces {
                reth1.0;
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy 1 {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        default-policy {
            deny-all;
        }
    }
}
```

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

Verification

Confirm that the configuration is working properly.

- Verifying Chassis Cluster Status on page 252
- Verifying Chassis Cluster Interfaces on page 253
- Verifying Chassis Cluster Statistics on page 253
- Verifying Chassis Cluster Control Plane Statistics on page 254
- Verifying Chassis Cluster Data Plane Statistics on page 254
- Verifying Chassis Cluster Redundancy Group Status on page 255
- Troubleshooting with Logs on page 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}
show chassis cluster status
Cluster ID: 1
Node                          Priority     Status     Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
    node0                       129         primary    no       no
    node1                       128         secondary  no       no
```



```
Redundancy group: 1 , Failover count: 1
    node0                  129          primary   no       no
    node1                  128          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
    reth1        Up          1

Interface Monitoring:
    Interface         Weight    Status    Redundancy-group
    xe-6/0/0          255       Up        1
    xe-6/1/0          255       Up        1
    xe-18/0/0         255       Up        1
    xe-18/1/0         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               100          primary   no       no
    node1               50           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 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

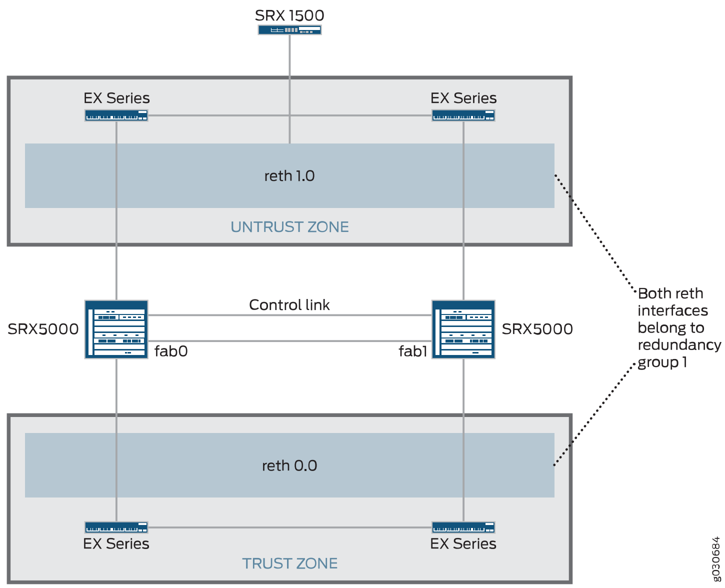
- Preparing Your Equipment for Chassis Cluster Formation on page 39
- Connecting SRX Series Devices to Create a Chassis Cluster on page 47
- Understanding Active/Passive Chassis Cluster Deployment on page 227
- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 228
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 239

Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel

Supported Platforms SRX Series, vSRX

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 19 on page 256). When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 19: Active/Passive Chassis Cluster with IPsec Tunnel Scenario (SRX Series Devices)



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 device becomes active, forcing the tunnel to change endpoints to terminate in the new active SRX Series device. 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.

NOTE: Dynamic tunnels cannot load-balance across different SPCs.

Related Documentation

- Understanding Active/Passive Chassis Cluster Deployment on page 227
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 256
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 271

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

Supported Platforms SRX Series, vSRX

This example shows how to configure active/passive chassis clustering with an IPsec tunnel for SRX Series devices.

Requirements

Before you begin:

- 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.
- 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.

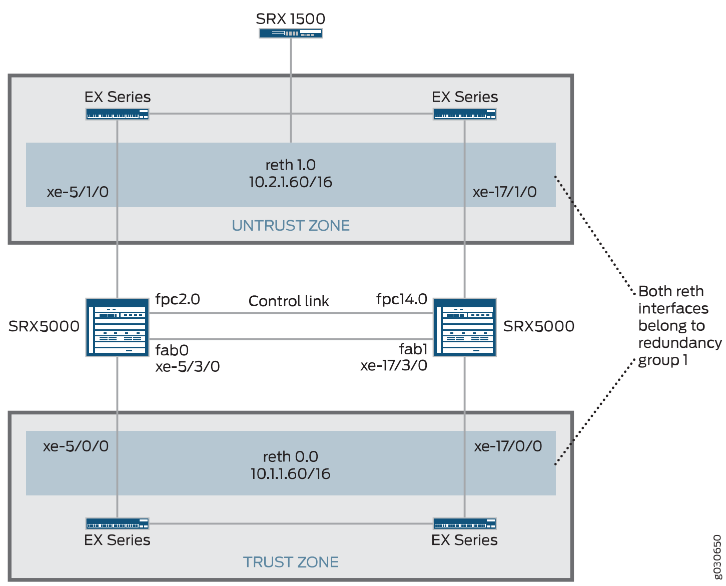
  Cluster ID greater than 15 can only be set when the fabric and control link interfaces are connected back-to-back.
- Get two SRX5000 models with identical hardware configurations, one SRX1500 edge router, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the 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 20 on page 258.) When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 20: Active/Passive Chassis Cluster with IPsec Tunnel Topology (SRX Series Devices)



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 17 on page 258 through Table 23 on page 261.

Table 17: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters

Groups	node0	• Hostname: SRX5800-1
		• Interface: fxp0
		  ◦ Unit 0
		  ◦ 172.19.100.50/24

	node1	• Hostname: SRX5800-2
		• Interface: fxp0
		  ◦ Unit 0
		  ◦ 172.19.100.51/24

Table 18: 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

Table 18: Chassis Cluster Configuration Parameters *(continued)*

Feature	Name	Configuration Parameters
Heartbeat interval	–	1000
Heartbeat threshold	–	3
Redundancy group	0	• Priority: ◦ Node 0: 254 ◦ Node 1: 1
	1	• Priority: ◦ Node 0: 254 ◦ Node 1: 1 •
		Interface monitoring • xe-5/0/0 • xe-5/1/0 • xe-17/0/0 • xe-17/1/0
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
		• Unit 0 • 10.1.1.60/16
	reth1	Redundancy group: 1
		• Multipoint • Unit 0 • 10.10.1.1/30
	st0	
		• Unit 0 • 10.10.1.1/30

Table 19: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	-
Policy	preShared	• Mode: main • Proposal reference: proposal-set standard • IKE Phase 1 policy authentication method: pre-shared-key ascii-text
Gateway	SRX1500-1	• IKE policy reference: perShared • External interface: reth0.0 • Gateway address: 10.1.1.90 NOTE: On all high-end SRX Series devices, only reth interfaces are supported for IKE external interface configuration in IPsec VPN. Other interface types can be configured, but IPsec VPN might not work. On all branch SRX Series devices, reth interfaces and the lo0 interface are supported for IKE external interface configuration in IPsec VPN. Other interface types can be configured, but IPsec VPN might not work. On all high-end SRX Series devices, the lo0 logical interface cannot be configured with RG0 if used as an IKE gateway external interface.

Table 20: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	–
Policy	std	–
VPN	SRX1500-1	• IKE gateway reference: SRX1500-1 • IPsec policy reference: std • Bind to interface: st0.0 • VPN monitoring: vpn-monitor optimized • Tunnels established: establish-tunnels immediately NOTE: The manual VPN name and the site-to-site gateway name cannot be the same.

Table 21: 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 22: Security Zone Configuration Parameters

Name	Configuration Parameters
trust	• All system services are allowed. • All protocols are allowed. • The reth0.0 interface is bound to this zone.
untrust	• All system services are allowed. • All protocols are allowed. • The reth1.0 interface is bound to this zone.
vpn	• All system services are allowed. • All protocols are allowed. • The st0.0 interface is bound to this zone.

Table 23: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

This security policy permits traffic from the trust zone to the untrust zone.	ANY	• Match criteria:
		  ◦ source-address any
		  ◦ destination-address any
		  ◦ application any
		• Action: permit

This security policy permits traffic from the trust zone to the vpn zone.	vpn-any	• Match criteria:
		  ◦ source-address any
		  ◦ destination-address any
		  ◦ application any
		• Action: permit

Configuration

CLI Quick Configuration

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

```
{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

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-panel 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**

Related Documentation

- Understanding Active/Passive Chassis Cluster Deployment on page 227
- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 255
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 271

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

Supported Platforms SRX Series, vSRX

1. Enable clusters. See Step 1 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 256.
2. Configure the management interface. See Step 2 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 256.
3. Configure the fabric interface. See Step 3 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 256.
4. Configure the redundancy groups.
   - Select **Configure>System Properties>Chassis Cluster**.
   - Enter the following information, and then click **Apply**:

     Redundant ether-Interfaces Count: **2**

     Heartbeat Interval: **1000**

     Heartbeat Threshold: **3**

     Nodes: **0**

     Group Number: **0**

     Priorities: **254**
   - Enter the following information, and then click **Apply**:

     Nodes: **0**

     Group Number: **1**

     Priorities: **254**
   - Enter the following information, and then click **Apply**:

     Nodes: **1**

     Group Number: **0**

     Priorities: **1**
   - Enter the following information, and then click **Apply**:

     Nodes: **1**

     Group Number: **1**

     Priorities: **1**

     Preempt: Select the check box.

     Interface Monitor—Interface: **xe-5/0/0**

     Interface Monitor—Weight: **255**

     Interface Monitor—Interface: **xe-5/1/0**

     Interface Monitor—Weight: **255**

     Interface Monitor—Interface: **xe-17/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **xe-17/1/0**

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 256.
6. Configure the IPsec configuration. See Step 6 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 256.
7. Configure the static routes .
   - Select **Configure>Routing>Static Routing**.
   - Click **Add**.
   - Enter the following information, and then click **Apply**:

     Static Route Address: **0.0.0.0/0**

     Next-Hop Addresses: **10.2.1.1**
   - Enter the following information, and then click **Apply**:

     Static Route Address: **10.3.0.0/16**

     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 256.

9. Configure the security policies. See Step 9 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel” on page 256.

10. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options**>**Commit**.

Related Documentation

- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 255
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 256

CHAPTER 21

Configuring Multicast and Asymmetric Routing

Understanding Multicast Routing on a Chassis Cluster

Supported Platforms SRX Series, vSRX

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 Junos OS Routing Protocols Library for Routing Devices 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.

Related Documentation

- Understanding Asymmetric Routing Chassis Cluster Deployment on page 276
- Example: Configuring an Asymmetric Chassis Cluster Pair on page 278

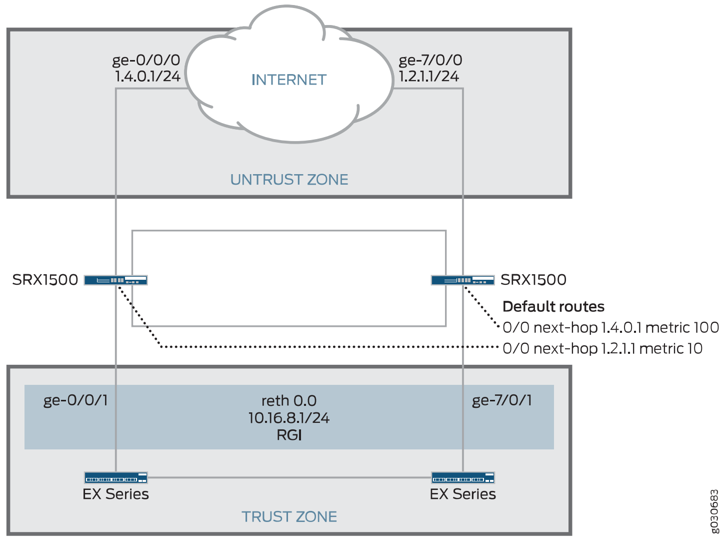
Understanding Asymmetric Routing Chassis Cluster Deployment

Supported Platforms SRX Series, vSRX

In this case, chassis cluster makes use of its asymmetric routing capability (see Figure 21 on page 277). 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.

NOTE: The anchor node for the session can change if there are changes in routing during the session.

Figure 21: 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 on page 277
- Understanding Failures in the Untrust Zone Interfaces on page 277

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 both created in node 0 and synced to node 1. However, session 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.)

Related Documentation

- Understanding Multicast Routing on a Chassis Cluster on page 275
- Example: Configuring an Asymmetric Chassis Cluster Pair on page 278

Example: Configuring an Asymmetric Chassis Cluster Pair

Supported Platforms SRX Series, vSRX

This example shows how to configure a chassis cluster pair of devices 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 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.

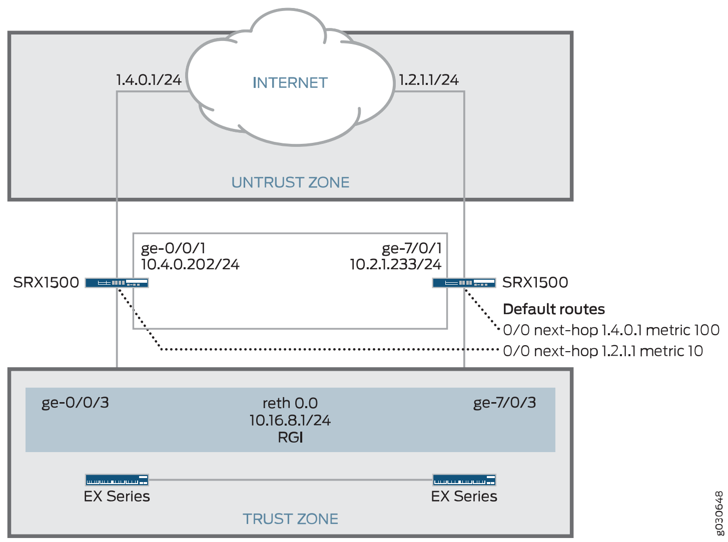
      user@host> **set chassis cluster cluster-id 1 node 0 reboot**
3. Connect to the other device using the console port.
   a. Set the cluster ID and node number.

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 22 on page 279, 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 22: 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 24 on page 279 through Table 27 on page 281.

Table 24: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters

Groups	node0	• Hostname: srxseries-1
		• Interface: fxp0
		  ◦ Unit 0
		  ◦ 192.168.100.50/24

	node1	• Hostname: srxseries-2
		• Interface: fxp0
		  ◦ Unit 0
		  ◦ 192.168.100.51/24

Table 25: 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	• Priority:
		  ◦ Node 0: 100
		  ◦ Node 1: 1

		Interface monitoring • ge-0/0/3 • ge-7/0/3
Number of redundant Ethernet interfaces	–	1
Interfaces	ge-0/0/1	• Unit 0 • 10.4.0.202/24
	ge-7/0/1	• Unit 0 • 10.2.1.233/24
	ge-0/0/3	• Redundant parent: reth0
	ge-7/0/3	• Redundant parent: reth0
	reth0	• Unit 0 • 10.16.8.1/24

Table 26: 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 27: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters

This security policy permits traffic from the trust zone to the untrust zone.	ANY	• Match criteria:
		  • source-address any
		  • destination-address any
		  • application any
		• Action: permit

Configuration

CLI Quick Configuration

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

```
{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 {
        gigether–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

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

- Understanding Multicast Routing on a Chassis Cluster on page 275
- Understanding Asymmetric Routing Chassis Cluster Deployment on page 276

PART 5

Upgrading or Disabling Chassis Cluster

CHAPTER 22

Upgrading Both Devices Separately

Upgrading Individual Devices in a Chassis Cluster Separately

Supported Platforms SRX Series, vSRX

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:

NOTE: **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 Both Devices in a Chassis Cluster Using an ISSU for High-End SRX Series Devices on page 299

- *Upgrading Devices in a Chassis Cluster Using ICU for Branch SRX Series Devices*

CHAPTER 23

Upgrading Both Devices Using Low-Impact ISSU

Understanding the Low-Impact ISSU Process on Devices in a Chassis Cluster

Supported Platforms SRX1500, SRX5400, SRX5600, SRX5800

In-service software upgrade (ISSU) allows a software upgrade from one Junos OS version to a later Junos OS version with little or no downtime.

The chassis cluster ISSU feature allows 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-D70, SRX1500 devices support ISSU.

An 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

NOTE:
The following limitations are related to an ISSU:

- 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 will 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 will 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 can not perform operations such as commit, restart, halt, and so on.
- During an ISSU, operations like fabric monitoring, control link recovery, and RGX preempt are suspended.
- During an ISSU, you cannot commit any configurations.

NOTE: For details on 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 ISSU 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 started. This action ensures that the redundancy groups are all active on only the RG-0 primary node.

   NOTE: 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 an ISSU, ensure that 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 available for the **/var** file system on both nodes, unsupported configurations, and unsupported Physical Interface Cards (PICs).

If there is insufficient disk space available on either of the Routing Engines, the ISSU process fails and returns an error message. However, unsupported PICs do not prevent an ISSU. The software issues a warning to indicate that these PICs will restart during the upgrade. Similarly, an unsupported protocol configuration does not prevent an ISSU. 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. The node 1 is upgraded with the new software image. Before being upgraded, the node 1 gets the configuration file from the 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 the node 0.
5. The chassis cluster process (chassisd) on the node 0 prepares other software processes for the low-impact 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 the node 1. Next, each Packet Forwarding Engine sends a message (ISSU ready) to the chassisd.
7. After receiving the message (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 the node 1 running the new software. The chassisd is also reestablished with the node 0.
8. After all Packet Forwarding Engines have sent a ready message using the chassisd on the node 0, other software processes are prepared for a node switchover. The system is ready for a switchover at this point.
9. The node switchover occurs and the node 1 becomes the new primary node (old secondary node 1).
10. The new secondary node (old primary node 0) is now upgraded to the new software image.

When both nodes are successfully upgraded, the ISSU is complete.

NOTE: When upgrading a version cluster that does not support encryption to a version that does support 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. Then 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.

Related Documentation

- ISSU System Requirements on page 298
- Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 299
- Troubleshooting Chassis Cluster ISSU-Related Problems on page 307
- Understanding Log Error Messages for Troubleshooting ISSU-Related Problems on page 302

ISSU System Requirements

Supported Platforms SRX1500, SRX5400, SRX5600, SRX5800

You can use ISSU to upgrade from an ISSU-capable software release to a newer software release.

To perform an ISSU, your device must be running a Junos OS release that supports ISSU for the specific platform. See Table 28 on page 298 for platform support.

Table 28: Low-Impact 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

NOTE: 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 aborted if the Junos OS version specified for installation is a version earlier than the one currently running on the device.
- The ISSU process is aborted 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.
- On all high-end SRX Series devices, ISSU does not support version downgrading.
- ISSU occasionally fails under a heavy CPU load.

NOTE: To downgrade from an ISSU-capable release to a previous software 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 performing ISSU under the following conditions:

- When the devices are operating in chassis cluster mode
- 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 KB17947.

Related Documentation

- Understanding the Low-Impact ISSU Process on Devices in a Chassis Cluster on page 295
- Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 299
- Troubleshooting Chassis Cluster ISSU-Related Problems on page 307
- Understanding Log Error Messages for Troubleshooting ISSU-Related Problems on page 302

Upgrading Both Devices in a Chassis Cluster Using an ISSU

Supported Platforms SRX1500, SRX5400, SRX5600, SRX5800

The chassis cluster ISSU feature allows both devices in a cluster to be upgraded from supported Junos OS versions with a traffic impact similar to that of redundancy group failovers.

Before you begin, note the following guidelines:

- 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
- If you are using Junos OS Release 11.4 or earlier, before starting an ISSU, fail over 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 136.

  If you are using Junos OS Release 12.1 or later, Junos OS software will automatically fail over all RGs to the RG0 primary.
- We recommend that you enable graceful restart for routing protocols before you start an ISSU.

NOTE: On all high-end SRX Series devices, the first recommended unified ISSU *from* release is Junos OS Release 10.4R4.

Starting with Junos OS Release 15.1X49-D70, SRX1500 devices support ISSU.

To perform an ISSU from the CLI:

1. Download the software package from the Juniper Networks Support website.
2. Copy the package on both nodes of the cluster. We recommend that you copy it 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. On the primary node, issue the **show version** command.
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* reboot

   NOTE: For SRX5400, SRX5600, and SRX5800 devices, you must include **reboot** in the command. If **reboot** is not included, the command fails.

   Wait for both nodes to complete the upgrade (you are logged out of the device).

5. Wait a few minutes, and then log in to the device again. Verify that both devices in the cluster are running the new Junos OS build using the **show version** command.
6. Verify that all policies, zones, redundancy groups, and other 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.

NOTE: If you want redundancy groups to automatically return to node 0 as the primary after the ISSU is complete, 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 fail over redundancy group 0.

To set the redundancy group priority and enable the preempt option, see "Example: Configuring Chassis Cluster Redundancy Groups" on page 81.

To manually fail over a redundancy group, see "Initiating a Chassis Cluster Manual Redundancy Group Failover" on page 136.

NOTE: 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 doing the upgrade. If the system finds that the new package is not version 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 abort the upgrade or turn off the feature before doing the upgrade.

Related Documentation

- **request system software in-service-upgrade (Maintenance) on page 379**
- Understanding the Low-Impact ISSU Process on Devices in a Chassis Cluster on page 295
- ISSU System Requirements on page 298
- Troubleshooting Chassis Cluster ISSU-Related Problems on page 307
- Understanding Log Error Messages for Troubleshooting ISSU-Related Problems on page 302
- 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

Supported Platforms SRX1500, SRX5400, SRX5600, SRX5800

If the ISSU fails to complete and only one device in the cluster has been upgraded, you can roll back to the previous configuration on that device alone by using the following commands on the upgraded device:

- **request chassis cluster in-service-upgrade abort**
- **request system software rollback *node node-id***
- **request system reboot**

Related Documentation

- Understanding the Low-Impact ISSU Process on Devices in a Chassis Cluster on page 295
- ISSU System Requirements on page 298
- Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 299
- Troubleshooting Chassis Cluster ISSU-Related Problems on page 307
- Understanding Log Error Messages for Troubleshooting ISSU-Related Problems on page 302

Enabling an Automatic Chassis Cluster Node Failback After an ISSU

Supported Platforms SRX5400, SRX5600, SRX5800

If you want redundancy groups to automatically return to node 0 as the primary after the ISSU is complete, 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 fail over redundancy group 0. To set the redundancy group priority and enable the preempt option, see "Example: Configuring Chassis Cluster Redundancy Groups" on page 81. To manually fail over a redundancy group, see "Initiating a Chassis Cluster Manual Redundancy Group Failover" on page 136.

NOTE: **To upgrade node 0 and make it available in the chassis cluster, manually reboot node 0. Node 0 does not reboot automatically.**

Related Documentation

- Understanding the Low-Impact ISSU Process on Devices in a Chassis Cluster on page 295
- ISSU System Requirements on page 298
- Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 299
- Troubleshooting Chassis Cluster ISSU-Related Problems on page 307
- Understanding Log Error Messages for Troubleshooting ISSU-Related Problems on page 302

Understanding Log Error Messages for Troubleshooting ISSU-Related Problems

Supported Platforms SRX5400, SRX5600, SRX5800, vSRX

The following problems might occur during an ISSU upgrade. You can identify the errors by using the details in the logs. You can also see the details of the error messages in the System Log Explorer.

Chassisd Process Errors

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 ISSU from a chassis perspective. If there is a problem, a log message is created.

Kernel State Synchronization

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 ISSU.

Installation Related Errors

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.

ISSU Support Related Errors

Problem **Description:** Installation failure 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
```

Redundancy Group Failover Errors

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-groupss has been failed over.
       Some redundancy-groups on node1 are already in manual failover mode.
       Please execute 'failover reset all' first..
```

Initial Validation Checks Fail

Problem **Description:** The initial validation checks fail.

Solution The following error messages are displayed when initial validation checks fail when the image is not present and 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

```
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, ISSU aborts and error messages are displayed.

Understanding Common Error Handling for ISSU

Problem **Description:** You might encounter some problems while using 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 ISSU is halted, so as not to create any mismatched versions on both nodes of the chassis cluster. Table 29 on page 305 provides some of the common error conditions and required workarounds. The sample messages used in the Table 29 on page 305 are from the SRX1500 device.

Table 29: ISSU-Related Errors and Solutions

Error Conditions	Solutions
Attempt to initiate an ISSU when previous instance of ISSU is already in progress	The following message is displayed: **warning: ISSU in progress** You can abort the current ISSU process, and initiate ISSU again using the **request chassis cluster in-service-upgrade abort** command.

Reboot failure on the secondary node	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.
	
	`error: [Oct  6 12:30:16]: Reboot secondary node failed (error-code: 4.1)`
	
	`error: [Oct  6 12:30:16]: ISSU Aborted! Backup node maybe in inconsistent state, Please restore backup node`
	`[Oct  6 12:30:16]: 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`

Table 29: ISSU-Related Errors and Solutions *(continued)*

<table>
<tr><th>Error Conditions</th><th>Solutions</th></tr>
<tr><td>Secondary node failed to complete the cold synchronization</td><td>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.<pre>
[Oct 3 14:00:46]: timeout waiting for secondary node node1
to sync(error-code: 6.1)
        Chassis control process started, pid 36707

       error: [Oct 3 14:00:46]: ISSU Aborted! Backup node
has been upgraded, Please restore backup node
       [Oct 3 14:00:46]: 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
</pre></td></tr>
<tr><td>Failover of newly upgraded secondary failed</td><td>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.End<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></td></tr>
<tr><td>Upgrade failure on primary</td><td>No service downtime occurs, because the secondary node fails over as primary and continues to provide required services.</td></tr>
</table>

Table 29: ISSU-Related Errors and Solutions *(continued)*

Error Conditions	Solutions
Reboot failure on primary node	Before the reboot of the primary node, devices will be out of ISSU setup and no ISSU-related error messages are displayed. The following reboot error message is displayed if there is any other failure detected: `Reboot failure on     Before the reboot of primary node, devices will be out of ISSU setup and no primary node error messages will be displayed. Primary node`

Related Documentation

- Understanding the Low-Impact ISSU Process on Devices in a Chassis Cluster
- ISSU System Requirements
- Upgrading Both Devices in a Chassis Cluster Using an ISSU
- Troubleshooting Chassis Cluster ISSU-Related Problems

Troubleshooting Chassis Cluster ISSU-Related Problems

Supported Platforms SRX5400, SRX5600, SRX5800

This topic includes the following sections:

Viewing the ISSU Progress

Problem **Description:** Rather than wait for an ISSU failure, you can display the progress of the ISSU as it occurs, noting any message where the ISSU was unsuccessful. Providing that message to TAC can help resolve the issue.

Solution After starting an ISSU, issue the **show chassis cluster information issu** command. Output similar to the following is sent to the console to indicate 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 When it Halts During an Upgrade

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 has been upgraded, you can roll back to the previous OS on that device alone by using the following commands on the upgraded device:

- Abort ISSU on both nodes using the **request chassis cluster in-service-upgrade abort** command.
- Rollback the image using the **request system software rollback** command with node option.
- Reboot the rolled back node using the **request system reboot** command.

Recovering the Node in Case of a Failed ISSU

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 aborts 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 an abort, the system message provides you with detailed information about when and why the upgrade stopped and 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}
```

NOTE: If you attempt to upgrade a device pair running a Junos OS image earlier than Release 9.6, the ISSU will fail without changing anything about either device in the cluster. Devices running Junos OS releases earlier than 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 is initiated, when power is restored the newly upgraded device will still be waiting to end the ISSU. To end the ISSU, issue the **request chassis cluster in-service-upgrade abort** command.

Related Documentation

- Understanding Log Error Messages for Troubleshooting ISSU-Related Problems on page 302

CHAPTER 24

Disabling Chassis Cluster

Disabling Chassis Cluster

Supported Platforms SRX Series, vSRX

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.

NOTE: After the chassis cluster is disabled using this CLI command, you do not have a similar CLI option to enable it back.

You can also use the below CLI commands to disable chassis cluster:

- To disable cluster on node0:

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

- To disable cluster on node1:

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

NOTE: Setting cluster-id to zero disables clustering on a device.

Related Documentation

- Upgrading Individual Devices in a Chassis Cluster Separately on page 293
- Upgrading Both Devices in a Chassis Cluster Using an ISSU for High-End SRX Series Devices on page 299

- *Upgrading Devices in a Chassis Cluster Using ICU for Branch SRX Series Devices*

PART 6

Configuration Statements and Operational Commands

CHAPTER 25

Configuration Statements

apply-groups (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax apply-groups *[${node}]*

Hierarchy Level [edit chassis cluster]

Release Information Statement introduced in Junos OS Release 9.0.

Description Apply node-specific parameters to each node in a chassis cluster.

Options *${node}* —Each node (node0 or node1) in a chassis cluster.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

arp-throttle

Supported Platforms SRX5400, SRX5600, SRX5800

Syntax

```
next-hop {
  arp-throttle seconds;
}
```

Hierarchy Level **[edit forwarding-options next-hop arp-throttle *seconds*]**

Release Information Statement introduced in Junos OS Release 15.1X49-D60.

Description Define the length of time (in seconds) for Address Resolution Protocol (ARP) request throttling. Set a greater time interval for the Routing Engine to process the request more slowly and thereby work more efficiently. For example, if a large number of hosts causes numerous ARP requests, Routing Engine utilization is reduced.

Options ***seconds***—Number of seconds the Routing Engine waits before receiving and processing an ARP request.

Range: 10 through 100 seconds

Default: 10 seconds

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

cluster (Chassis)

Supported Platforms SRX Series, vSRX

Syntax

```
cluster {
  configuration-synchronize {
    no-secondary-bootup-auto;
  }
  control-link-recovery;
  heartbeat-interval milliseconds;
  heartbeat-threshold number;
  network-management {
    cluster-master;
  }
  redundancy-group group-number {
    gratuitous-arp-count number;
    hold-down-interval number;
    interface-monitor interface-name {
      weight number;
    }
    ip-monitoring {
      family {
        inet {
          ipv4-address {
            interface {
              logical-interface-name;
              secondary-ip-address ip-address;
            }
            weight number;
          }
        }
      }
      global-threshold number;
      global-weight number;
      retry-count number;
      retry-interval seconds;
    }
    node (0 | 1 ) {
      priority number;
    }
    preempt;
  }
  reth-count number;
  traceoptions {
    file {
      filename;
      files number;
      match regular-expression;
      (world-readable | no-world-readable);
      size maximum-file-size;
    }
    flag flag;
    level {
      (alert | all | critical | debug | emergency | error | info | notice | warning);
    }
```

```
    no-remote-trace;
  }
}
```

Hierarchy Level [edit chassis]

Release Information Statement introduced in Junos OS Release 9.0.

Description Configure a chassis cluster.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- ip-monitoring on page 335

configuration-synchronize (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax

```
configuration-synchronize {
  no-secondary-bootup-auto;
}
```

Hierarchy Level [edit chassis cluster]

Release Information Statement introduced in Junos OS Release 12.1X47-D10.

Description Disables the automatic chassis cluster synchronization between the primary and secondary nodes. To reenable automatic chassis cluster synchronization, use the **delete chassis cluster configuration-synchronize no-secondary-bootup-auto** command in configuration mode.

Options **no-secondary-bootup-auto**—Disable the automatic chassis cluster synchronization between the primary and secondary nodes.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 161
- request chassis cluster configuration-synchronize on page 367
- show chassis cluster information configuration-synchronization on page 398

control-link-recovery

Supported Platforms SRX Series, vSRX

Syntax control-link-recovery;

Hierarchy Level [edit **chassis cluster**]

Release Information Statement introduced in Junos OS Release 9.5.

Description Enable control link recovery to be done automatically by the system. After the control link recovers, the system checks whether it receives at least 30 consecutive heartbeats on the control link. This is to ensure that the control link is not flapping and is perfectly healthy. Once this criterion is met, the system issues an automatic reboot on the node that was disabled when the control link failed. When the disabled node reboots, the node rejoins the cluster. There is no need for any manual intervention.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- interface (Chassis Cluster) on page 332

control-ports

Supported Platforms SRX5600, SRX5800

Syntax

```
fpc slot-number {
  offline;
  pic slot-number {
    aggregate-ports;
    framing {
      (e1 | e3 | sdh | sonet | t1 | t3);
    }
    max-queues-per-interface (4 | 8);
    mlfr-uni-nni-bundles number;
    no-multi-rate;
    port slot-number {
      framing (e1 | e3 | sdh | sonet | t1 | t3);
      speed (oc12-stm4 | oc3-stm1 | oc48-stm16);
    }
    q-pic-large-buffer (large-scale | small-scale);
    services-offload {
      low-latency;
      per-session-statistics;
    }
    shdsl {
      pic-mode (1-port-atm | 2-port-atm | 4-port-atm | efm);
    }
    sparse-dlcis;
    traffic-manager {
      egress-shaping-overhead number;
      ingress-shaping-overhead number;
      mode (egress-only | ingress-and-egress);
    }
    tunnel-queuing;
  }
}
```

Hierarchy Level [edit chassis cluster]

Release Information Statement introduced in Junos OS Release 9.2. Support for dual control ports added in Junos OS Release 10.0.

Description Enable the specific control port of the Services Processing Card (SPC) for use as a control link for the chassis cluster. By default, all control ports are disabled. User needs to configure a minimum of one control port per chassis of the cluster. If user configures port 0 only, the Juniper Services Redundancy Protocol process (jsrpd) does not send control heartbeats on control link 1 and the counters it sends will show zeroes.

Options

- fpc ***slot-number*** —Flexible PIC Concentrator (FPC) slot number.

NOTE: FPC slot range depends on platform. The maximum range of 0 through 23 applies to SRX5800 devices; for SRX5600 devices, the only applicable range is 0 through 11; for SRX5400 devices, the applicable slot range is 0 through 5.

- port *port-number* —Port number on which to configure the control port.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- cluster (Chassis) on page 318

device-count (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax device-count *number*;

Hierarchy Level [edit chassis aggregated-devices ethernet]
[edit chassis aggregated-devices sonnet]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure the number of aggregated logical devices.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- cluster (Chassis) on page 318
- *Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI)*

ethernet (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax

```
ethernet {
  device-count number;
  lacp {
    link-protection {
      non-revertive;
    }
    system-priority number;
  }
}
```

Hierarchy Level [edit chassis aggregated-devices]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure properties for aggregated Ethernet devices.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318
- *Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI)*

fabric-options

Supported Platforms SRX Series, vSRX

Syntax

```
fabric-options {
  member-interfaces member-interface-name;
}
```

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure fabric interface specific options in chassis clusters.

NOTE: When you run the **system autoinstallation** command, the command will configure unit 0 logical interface for all the active state physical interfaces. However, a few commands such as **fabric-options** do not allow the physical interface to be configured with a logical interface. If the **system autoinstallation** and the **fabric-options** commands are configured together, the following message is displayed:

```
incompatible with 'system autoinstallation'
```

Options The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring the Chassis Cluster Fabric Interfaces on page 68
- member-interfaces on page 337

gigether-options (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax

```
gigether-options {
  802.3ad {
    backup | primary
    lacp {
    port-priority number;
    }
  }
  auto-negotiation {
    remote-fault;
  }
  flow-control | no-flow-control;
  ieee-802-3az-eee ;
  ignore-l3-incompletes;
  loopback | no-loopback
  loopback-remote
  no-auto-negotiation;
  redundant-parent interface-name;
}
```

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Configure Gigabit Ethernet specific interface properties.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 85
- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88

global-threshold

Supported Platforms SRX Series, vSRX

Syntax global-threshold *number*;

Hierarchy Level [edit chassis cluster redundancy-group ***group-number*** ip-monitoring]

Release Information Statement introduced in Junos OS Release 10.1.

Description Specify the failover value for all IP addresses monitored by the redundancy group. When IP addresses with a configured total weight in excess of the threshold have become unreachable, the weight of IP monitoring is deducted from the redundancy group threshold.

Options *number* —Value at which the IP monitoring weight is applied against the redundancy group failover threshold.

Range: 0 through 255

Default: 0

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- ip-monitoring on page 335

global-weight

Supported Platforms SRX Series, vSRX

Syntax global-weight *number*;

Hierarchy Level [edit chassis cluster redundancy-group *group-number* ip-monitoring]

Release Information Statement introduced in Junos OS Release 10.1.

Description Specify the relative importance of all IP address monitored objects to the operation of the redundancy group. Every monitored IP address is assigned a weight. If the monitored address becomes unreachable, the weight of the object is deducted from the global-threshold of IP monitoring objects in its redundancy group. When the global-threshold reaches 0, the global-weight is deducted from the redundancy group. Every redundancy group has a default threshold of 255. If the threshold reaches 0, a failover is triggered. Failover is triggered even if the redundancy group is in manual failover mode and preemption is not enabled.

Options *number* —Combined weight assigned to all monitored IP addresses. A higher weight value indicates a greater importance.

Range: 0 through 255

Default: 255

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- ip-monitoring on page 335

gratuitous-arp-count

Supported Platforms SRX Series, vSRX

Syntax gratuitous-arp-count *number*;

Hierarchy Level [edit chassis cluster redundancy-group group-number]

Release Information Statement introduced in Junos OS Release 9.0.

Description Specify the number of gratuitous Address Resolution Protocol (ARP) requests to send on an active interface after failover.

Options ***number***—Number of gratuitous ARP requests that a newly elected primary device in a chassis cluster sends out to announce its presence to the other network devices.

Range: 1 through 16

Default: 4

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- redundancy-group (Chassis Cluster) on page 341

heartbeat-interval

Supported Platforms SRX Series, vSRX

Syntax heartbeat-interval *milliseconds*;

Hierarchy Level [edit **chassis cluster**]

Release Information Statement introduced in Junos OS Release 9. Statement updated in Junos OS Release 10.4.

Description Set the interval between the periodic signals broadcast to the devices in a chassis cluster to indicate that the active node is operational.

The **heartbeat-interval** option works in combination with the **heartbeat-threshold** option to define the wait time before failover is triggered in a chassis cluster. The default values of these options produce a wait time of 3 seconds. In a large configuration approaching full capacity on an SRX5400 or SRX5600 or SRX5800 device, however, we recommend that you increase the failover wait time to 5 seconds.

For example, a **heartbeat-threshold** of 3 and a **heartbeat-interval** of 1000 milliseconds result in a total wait of 3 seconds before failover is triggered. To increase this wait to 5 seconds, you could increase the **heartbeat-threshold**, the **heartbeat-interval**, or both. 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.

NOTE: In a chassis cluster scaling environment, the **heartbeat-threshold** must always be set to 8.

Options *milliseconds* —Time interval between any two heartbeat messages.

Range: 1000 through 2000 milliseconds

Default: 1000 milliseconds

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

heartbeat-threshold

Supported Platforms SRX Series, vSRX

Syntax heartbeat-threshold *number*;

Hierarchy Level [edit **chassis cluster**]

Release Information Statement introduced in Junos OS Release 9.0. Statement updated in Junos OS Release 10.4.

Description Set the number of consecutive missed heartbeat signals that a device in a chassis cluster must exceed to trigger failover of the active node.

The **heartbeat-threshold** option works in combination with the **heartbeat-interval** option to define the wait time before failover is triggered in a chassis cluster. The default values of these options produce a wait time of 3 seconds. In a large configuration approaching full capacity on an SRX5400 or SRX5600 or SRX5800 device, however, we recommend that you increase the failover wait time to 5 seconds.

For example, a **heartbeat-threshold** of 3 and a **heartbeat-interval** of 1000 milliseconds result in a total wait of 3 seconds before failover is triggered. To increase this wait to 5 seconds, you could increase the **heartbeat-threshold**, the **heartbeat-interval**, or both. 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.

Options *number* —Number of consecutive missed heartbeats.

Range: 3 through 8

Default: 3

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

hold-down-interval

Supported Platforms SRX Series, vSRX

Syntax hold-down-interval *number*;

Hierarchy Level [edit chassis cluster redundancy-group *group-number*]

Release Information Statement introduced in Junos OS Release 10.0.

Description Set the minimum interval to be allowed between back-to-back failovers for the specified redundancy group (affects manual failovers, as well as automatic failovers associated with monitoring failures).

For redundancy group 0, this setting prevents back-to-back failovers from occurring less than 5 minutes (300 seconds) apart. Note that a redundancy group 0 failover implies a Routing Engine failure.

For some configurations, such as ones with a large number of routes or logical interfaces, the default or specified interval for redundancy group 0 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.

Options *number*—Number of seconds specified for the interval.

Range: For redundancy group 0, 300 through 1800 seconds; for redundancy group 1 through 128, 0 through 1800 seconds.

Default: For redundancy group 0, 300 seconds; for redundancy group 1 through 128, 1 second.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

interface (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax
```
interface {
  logical-interface-name;
  secondary-ip-address ip-address;
}
```

Hierarchy Level [edit chassis cluster redundancy-group ***group-number*** ip-monitoring family *family-name* *IP–address*]

Release Information Statement introduced in Junos OS Release 10.1.

Description Specify the redundant Ethernet interface, including its logical-unit-number, through which the monitored IP address must be reachable. The specified redundant Ethernet interface can be in any redundancy group. Likewise specify a secondary IP address to be used as a ping source for monitoring the IP address through the secondary node's redundant Ethernet interface link.

Options

- **reth*X.logical-unit-number***—Redundant Ethernet interface through which the monitored IP address must be reachable. You must specify the redundant Ethernet interface logical-unit-number. Note that you must also configure a secondary ping source IP address (see below).

Range: **reth0.*logical-unit-number*** through **reth128.*logical-unit-number*** (device dependent)

NOTE: **If the redundant Ethernet interface belongs to a VPN routing and forwarding (VRF) routing instance type, then the IP monitoring feature will not work.**

- **secondary-ip-address *IP–address***—Specify the IP address that are used as the source IP address of ping packets for IP monitoring from the secondary child link of the redundant Ethernet interface. An IP address for sourcing the ping packets on the primary link of the redundant Ethernet interface must be configured before you can configure secondary-ip-address. For legacy support reasons, monitoring on an IP address without identifying a redundant Ethernet interface and without configuring a secondary ping source IP address is permitted but not recommended.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

interface-monitor

Supported Platforms SRX Series, vSRX

Syntax

```
interface-monitor interface-name {
  weight number;
}
```

Hierarchy Level [edit **chassis cluster redundancy-group** ***group-number***]

Release Information Statement introduced in Junos OS Release 9.0.

Description Specify a redundancy group interface to be monitored for failover and the relative weight of the interface.

Options ***interface-name*** —Name of the physical interface to monitor.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

internal (Security IPsec)

Supported Platforms SRX5400, SRX5600, SRX5800

Syntax

```
internal {
  security-association {
  manual encryption {
    iked_encryption enabled;
    algorithm 3des-cbc;
    key ascii-text key;
    }
  }
}
```

Hierarchy Level [edit **security ipsec internal-security-association**]

Release Information Statement introduced in Junos OS Release 12.1X45-D10.
Support for **iked_encryption** option added in Junos OS Release 12.1X47-D15.

Description Enable secure login by configuring the internal IP security (IPsec) security association (SA). When the internal IPsec is configured, IPsec-based **rlogin** and remote command (**rcmd**) are enforced, so an attacker cannot gain unauthorized information.

Options **security-association**—Specify an IPsec SA.

manual encryption—Specify a manual SA.

iked_encryption—Select the iked_encryption option.

algorithm—Specify the encryption algorithm for the internal Routing-Engine-to-Routing-Engine IPsec SA configuration.

NOTE: Only the 3des-cbc encryption algorithm is supported.

key—Specify the encryption key. You must ensure that the manual encryption key is in ASCII text and 24 characters long; otherwise, the configuration will result in a commit failure.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Security Configuration Statement Hierarchy*

ip-monitoring

Supported Platforms SRX Series, vSRX

Syntax

```
ip-monitoring {
  family {
    inet {
      ipv4-address {
        interface {
          logical-interface-name;
          secondary-ip-address ip-address;
        }
        weight number;
      }
    }
  }
  global-threshold number;
  global-weight number;
  retry-count number;
  retry-interval seconds;
}
```

Hierarchy Level [edit **chassis cluster redundancy-group** *group-number*]

Release Information Statement updated in Junos OS Release 10.1.

Description Specify a global IP address monitoring threshold and weight, and the interval between pings (**retry-interval**) and the number of consecutive ping failures (**retry-count**) permitted before an IP address is considered unreachable for all IP addresses monitored by the redundancy group. Also specify IP addresses, a monitoring weight, a redundant Ethernet interface number, and a secondary IP monitoring ping source for each IP address, for the redundancy group to monitor.

Options **family inet *IPv4 address***—The address to be continually monitored for reachability.

> NOTE: All monitored object failures, including IP monitoring, are deducted from the redundancy group threshold priority. Other monitored objects include interface monitor, SPU monitor, cold-sync monitor, and NPC monitor (on supported platforms).

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- interface (Chassis Cluster)
- global-threshold on page 326
- global-weight on page 327

- weight
- Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 208

lacp (Interfaces)

Supported Platforms SRX Series

Syntax

```
lacp {
  active;
  passive;
  periodic;
}
```

Hierarchy Level [edit interfaces *interface-name* redundant-ether-options]

Release Information Statement introduced in Junos OS Release 10.2.

Description For redundant Ethernet interfaces in a chassis cluster only, configure Link Aggregation Control Protocol (LACP).

Options

- **active**—Initiate transmission of LACP packets.
- **passive**—Respond to LACP packets.
- **periodic**— Interval for periodic transmission of LACP packets.

Default: If you do not specify **lacp** as either **active** or **passive**, LACP remains off (the default).

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding LACP on Standalone Devices*

link-protection (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax

```
link-protection {
  non-revertive;
}
```

Hierarchy Level [edit chassis aggregated-devices ethernet lacp]

Release Information Statement introduced in Junos OS Release 10.2.

Description Enable Link Aggregation Control Protocol (LACP) link protection at the global (chassis) level.

Options **non-revertive**—Disable the ability to switch to a better priority link (if one is available) once a link is established as active and a collection or distribution is enabled.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318
- *Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI)*

member-interfaces

Supported Platforms SRX Series, vSRX

Syntax member-interfaces *member-interface-name*;

Hierarchy Level [edit interfaces *interface-name* fabric-options]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the member interface name. Member interfaces that connect to each other must be of the same type.

Options *member-interface-name*—Member interface name.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding Interfaces*

network-management

Supported Platforms SRX Series, vSRX

Syntax

```
network-management {
  cluster-master;
}
```

Hierarchy Level [edit chassis cluster]

Release Information Statement introduced in Junos OS Release 11.1.

Description Define parameters for network management. To manage an SRX Series Services Gateway cluster through a non-fxp0 interface, use this command to define the node as a virtual chassis in NSM. This command establishes a single DMI connection from the primary node to the NSM server. This connection is used to manage both nodes in the cluster. Note that the non-fxp0 interface (regardless of which node it is present on) is always controlled by the primary node in the cluster. The output of a *<get-system-information>* RPC returns a *<chassis-cluster>* tag in all SRX Series devices. When NSM receives this tag, it models SRX Series clusters as devices with autonomous control planes.

Options **cluster-master**—Enable in-band management on the primary cluster node through NSM.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

node (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax

```
node (0 | 1 ) {
  priority number;
}
```

Hierarchy Level [edit chassis cluster]

Release Information Statement introduced in Junos OS Release 9.0.

Description Identify the device in a chassis cluster. The node 0 device in the cluster has the chassis ID 1, and the node 1 device in the cluster has the chassis ID 2.

Options ***node-number*** —Cluster node number.

Range: 0 through 1

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

node (Chassis Cluster Redundancy Group)

Supported Platforms SRX Series, vSRX

Syntax

```
node (0 | 1 ) {
  priority number;
}
```

Hierarchy Level [edit **chassis cluster redundancy-group** ***group-number***]

Release Information Statement introduced in Junos OS Release 9.0.

Description Identify each cluster node in a redundancy group and set its relative priority for mastership.

Options ***node-number*** —Cluster node number, set with the **chassis cluster node** ***node-number*** statement.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- redundancy-group (Chassis Cluster) on page 341

preempt (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax preempt;

Hierarchy Level [edit chassis cluster redundancy-group *group-number*]

Release Information Statement introduced in Junos OS Release 9.0.

Description Enable chassis cluster node preemption within a redundancy group. If **preempt** is added to a redundancy group configuration, the device with the higher priority in the group can initiate a failover to become master. By default, preemption is disabled.

Initiating a failover with the **request chassis cluster failover node** or **request chassis cluster failover redundancy-group** command overrides the priority settings and preemption.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- redundancy-group (Chassis Cluster) on page 341

priority (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax priority *number*;

Hierarchy Level [edit chassis cluster redundancy-group *group-number* node *node-number*]

Release Information Statement introduced in Junos OS Release 9.0.

Description Define the priority of a node (device) in a redundancy group. Initiating a failover with the **request chassis cluster failover node** or **request chassis cluster failover redundancy-group** command overrides the priority settings.

Options *number*—Priority value of the node. The eligible node with the highest priority is elected master.

Range: 1 through 254

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- redundancy-group (Chassis Cluster) on page 341

redundancy-group (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax

```
redundancy-group group-number {
  gratuitous-arp-count number;
  hold-down-interval number;
  interface-monitor interface-name {
    weight number;
  }
  ip-monitoring {
    family {
      inet {
        ipv4-address {
          interface {
            logical-interface-name;
            secondary-ip-address ip-address;
          }
          weight number;
        }
      }
    }
    global-threshold number;
    global-weight number;
    retry-count number;
    retry-interval seconds;
  }
  node (0 | 1 ) {
    priority number;
  }
  preempt;
}
```

Hierarchy Level [edit chassis cluster]

Release Information Statement introduced in Junos OS Release 9.0.

Description Define a redundancy group. Except for redundancy group 0, a redundancy group is a logical interface consisting of two physical Ethernet interfaces, one on each chassis. One interface is active, and the other is on standby. When the active interface fails, the standby interface becomes active. The logical interface is called a redundant Ethernet interface (**reth**).

Redundancy group 0 consists of the two Routing Engines in the chassis cluster and controls which Routing Engine is primary. You must define redundancy group 0 in the chassis cluster configuration.

Options ***group-number*** —Redundancy group identification number.

Range: 0 through 128

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

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- ip-monitoring on page 335

redundancy-interface-process

Supported Platforms SRX Series, vSRX

Syntax

```
redundancy-interface-process {
   command binary-file-path;
   disable;
   failover (alternate-media | other-routing-engine);
}
```

Hierarchy Level [edit system processes]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify as an active or backup process of an application server, configure to process traffic for more than one logical application server.

Options

- **command *binary-file-path***—Path to the binary process.
- **disable**—Disable the redundancy interface management process.
- **failover**—Configure the device to reboot if the software process fails four times within 30 seconds, and specify the software to use during the reboot.
  - **alternate-media**—Configure the device to switch to backup media that contains a version of the system if a software process fails repeatedly.
  - **other-routing-engine**—Instruct the secondary Routing Engine to take mastership if a software process fails. If this statement is configured for a process, and that process fails four times within 30 seconds, then the device reboots from the secondary Routing Engine.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

redundant-ether-options

Supported Platforms SRX Series, vSRX

Syntax

```
redundant-ether-options {
  (flow-control | no-flow-control);
  lacp {
    (active | passive);
    periodic (fast | slow);
  }
  link-speed speed;
  (loopback | no-loopback);
  minimum-links number;
  redundancy-group number;
  source-address-filter mac-address;
  (source-filtering | no-source-filtering);
}
```

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Configure Ethernet redundancy options for a chassis cluster.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Enabling Eight Queue Class of Service on Redundant Ethernet Interfaces on page 95
- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88

redundant-parent (Interfaces)

Supported Platforms SRX Series, vSRX

Syntax redundant-parent *redundant-ethernet-interface-name*;

Hierarchy Level [edit interfaces *interface-name* gigether-options]
[edit interfaces *interface-name* fastether-options]

Release Information Statement introduced in Junos OS Release 10.2.

Description Assign local (child) interfaces to the redundant Ethernet (reth) interfaces. A redundant Ethernet interface contains 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).

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88

redundant-pseudo-interface-options

Supported Platforms SRX Series, vSRX

Syntax

```
redundant-pseudo-interface-options {
  redundancy-group redundancy-group-number;
}
```

Hierarchy Level [edit interfaces lo0]

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Configure the loopback pseudointerface in a redundancy group.

An Internet Key Exchange (IKE) gateway operating in chassis cluster, needs an external interface to communicate with a peer device. When an external interface (a reth interface or a standalone interface) is used for communication; the interface might go down when the physical interfaces are down. Instead, use loopback interfaces as an alternative to physical interfaces.

Options *redundancy-group-number*— Configure the redundancy group number.

Range: 0 through 255

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding Loopback Interface for a High Availability VPN*

reth-count (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax reth-count *number*;

Hierarchy Level [edit **chassis cluster**]

Release Information Statement introduced in Junos OS Release 9.0.

Description Specify the number of redundant Ethernet (**reth**) interfaces allowed in the chassis cluster. Note that the number of **reth** interfaces configured determines the number of redundancy groups that can be configured.

Options *number* —Number of redundant Ethernet interfaces allowed.

Range: 1 through 128

Default: 0

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

reth (Interfaces)

Supported Platforms SRX Series, vSRX

Syntax

```
reth <0 |1> {
  accounting-profile;
  description;
  disable;
  encapsulation;
  gratuitous-arp-reply;
  hierarchical-scheduler {
    implicit-hierarchy;
    maximum-hierarchy-levels;
  }
  mac;
  mtu;
  native-vlan-id;
  no-gratuitous-arp-reply;
  no-gratuitous-arp-request;
  (per-unit-scheduler | no-per-unit-scheduler);
  promiscuous-mode;
  redundant-ether-options {
    (flow-control | no-flow-control);
    lacp {
      (active | passive);
      periodic (fast | slow);
    }
    link-speed speed;
    (loopback | no-loopback);
    minimum-links number;
    redundancy-group number;
  }
  traceoptions {
    flag (all | event | ipc | media);
    }
  }
  (traps | no-traps);
  unit unit-number {
    accounting-profile name;
    alias;
    bandwidth bandwidth;
    description text;
    disable;
    encapsulation (dix | ether-vpls-fr | frame-relay-ppp | ppp-over-ether | vlan-bridge |
      vlan-ccc | vlan-vpls |vlan-tcc);
    family {
      ethernet-switching {
        bridge-domain-type (svlan| bvlan);
        inner-vlan [members];
        inter-switch-link;
        interface-mode (access | trunk);
        recovery-timeout seconds;
        storm-control;
        vlan [members];
        vlan-auto-sense;
```



```
    vlan-rewrite {
      translate {
        from-vlan-id;
        to-vlan-id ;
      }
    }
  }
  inet {
    accounting {
      destination-class-usage;
      source-class-usage {
        input;
        output;
      }
    }
    address (source–address/prefix) {
      arp destination-address ;
      }
      broadcast address;
      preferred;
      primary;
      vrrp-group group-id {
        (accept-data | no-accept-data);
        advertise-interval seconds;
        advertisements-threshold number;
        authentication-key key-value;
        authentication-type (md5 | simple);
        fast-interval milliseconds;
        inet6-advertise-interval milliseconds
        (preempt <hold-timeseconds> | no-preempt );
        preferred;
        priority value;
        track {
          interface interface-name {
            bandwidth-threshold bandwidth;
            priority-cost value;
          }
          priority-hold-time seconds;
          route route-address{
            routing-instance routing-instance;
            priority-cost value;
          }
        }
        virtual-address [address];
        virtual-link-local-address address;
        vrrp-inherit-from {
          active-group value;
          active-interface interface-name;
        }
      }
      web-authentication {
        http;
        https;
        redirect-to-https;
      }
    }
```

```
dhcp {
  client-identifier {
    (ascii string | hexadecimal string);
  }
  lease-time (length | infinite);
  retransmission-attempt value;
  retransmission-interval seconds;
  server-address server-address;
  update-server;
  vendor-id vendor-id ;
}
dhcp-client {
  client-identifier {
    prefix {
      host-name;
      logical-system-name;
      routing-instance-name;
    }
    use-interface-description (device | logical);
    user-id (ascii string| hexadecimal string);
  }
  lease-time (length | infinite);
  retransmission-attempt value;
  retransmission-interval seconds;
  server-address server-address;
  update-server;
  vendor-id vendor-id ;
}
filter {
  group number;
  input filter-name;
  input-list [filter-name];
  output filter-name;
  output-list [filter-name];
}
mtu value;
no-neighbor-learn;
no-redirects;
policer {
  input input-name;
}
primary;
rpf-check {
  fail-filter filter-name;
  mode {
    loose;
  }
}
sampling {
  input;
  output;
}
simple-filter;
unconditional-src-learn;
unnumbered-address {
  interface-name;
```

```
      preferred-source-address preferred-source-address;
    }
  }
  inet6 {
    accounting {
      destination-class-usage;
      source-class-usage {
        input;
        ouput;
      }
    }
    address source–address/prefix {
      eui-64;
      ndp address {
        (mac mac-address | multicast-mac multicast-mac-address);
        publish;
      }
      preferred;
      primary;
      vrrp-inet6-group group_id {
        (accept-data | no-accept-data);
        advertisements-threshold number;
        authentication-key value;
        authentication-type (md5 | simple);
        fast-interval milliseconds;
        inet6-advertise-interval milliseconds;
        (preempt <hold-time seconds>| no-preempt );
        priority value;
        track {
          interface interface-name {
            bandwidth-threshold value;
            priority-cost value;
          }
          priority-hold-time seconds;
          route route-address{
            routing-instance routing-instance;
          }
        }
        vrrp-inherit-from {
          active-group value;
          active-interface interface-name;
        }
      }
      web-authentication {
        http;
        https;
        redirect-to-https;
      }
    }
    (dad-disable | no-dad-disable);
    filter {
      group number;
      input filter-name;
      input-list [filter-name];
      output filter-name;
      output-list [filter-name];
```

```
            }
            mtu value;
            nd6-stale-time seconds;
            no-neighbor-learn;
            no-redirects;
            rpf-check {
              fail-filter filter-name;
              mode {
                loose;
              }
            }
            sampling {
              input;
              output;
            }
            unnumbered-address;
          }
          iso {
            address source-address;
            mtu value;
          }
          vpls {
            filter {
              group number;
              input filter-name;
              input-list [filter-name];
              output filter-name;
              output-list [filter-name];
            }
            policer {
              input input-name;
              output output-name;
            }
          }
        }
        native-inner-vlan-id value;
        (no-traps | traps);
        proxy-arp (restricted | unrestricted);
        traps;
        vlan-id vlan-id;
        vlan-id-list vlan-id-list;
        vlan-id-range vlan-id1-vlan-id2;
      }
      vlan-tagging;
    }
```

Hierarchy Level [edit interfaces]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure a redundant Ethernet interface (reth) for chassis cluster. It is a pseudointerface that includes at minimum of one physical interface from each node of the cluster.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 88
- **cluster (Chassis) on page 318**
- **redundant-ether-options on page 343**
- **lacp (Interfaces) on page 336**

retry-count (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax retry-count *number*;

Hierarchy Level [edit chassis cluster redundancy-group ***group-number*** ip-monitoring]

Release Information Statement introduced in Junos OS Release 10.1.

Description Specify the number of consecutive ping attempts that must fail before an IP address monitored by the redundancy group is declared unreachable. (See **retry-interval** for a related redundancy group IP address monitoring variable.)

Options ***number*** —Number of consecutive ping attempt failures before a monitored IP address is declared unreachable.

Range: 1 through 15

Default: 5

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- **cluster (Chassis) on page 318**

retry-interval (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax retry-interval *interval*;

Hierarchy Level [edit chassis cluster redundancy-group ***group-number*** ip-monitoring]

Release Information Statement introduced in Junos OS Release 10.1.

Description Specify the ping packet send frequency (in seconds) for each IP address monitored by the redundancy group. (See **retry-count** for a related IP address monitoring configuration variable.)

Options ***interval***—Pause time between each ping sent to each IP address monitored by the redundancy group.

Range: 1 to 30 seconds

Default: 1 second

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- ip-monitoring on page 335

route-active-on

Supported Platforms SRX Series, vSRX

Syntax route-active-on (node0 | node1);

Hierarchy Level [edit policy-options condition *condition-name*]

Release Information Statement introduced in Junos OS Release 9.0.

Description For chassis cluster configurations, identify the device (node) on which a route is active.

Options **node0** | **node1**—Node in a chassis cluster.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

traceoptions (Chassis Cluster)

Supported Platforms SRX Series, vSRX

Syntax

```
traceoptions {
  file {
    filename;
    files number;
    match regular-expression;
    (world-readable | no-world-readable);
    size maximum-file-size;
  }
  flag flag;
  level {
    (alert | all | critical | debug | emergency | error | info | notice | warning);
  }
  no-remote-trace;
}
```

Hierarchy Level [edit **chassis cluster**]

Release Information Statement modified in Junos OS Release 9.5.

Description Define chassis cluster redundancy process tracing operations.

Options

- **file** ***filename*** —Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.
- **files** ***number*** —(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed to ***trace-file*** **.0**, then ***trace-file.1*** **,** and so on, until the maximum number of trace files is reached. The oldest archived file is overwritten.
- If you specify a maximum number of files, you also must specify a maximum file size with the size option and a filename.

Range: 2 through 1000 files

Default: 10 files

- **match** ***regular-expression*** —(Optional) Refine the output to include lines that contain the regular expression.
- **size** ***maximum-file-size*** —(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*** **.0**. When the **trace-file** again reaches its maximum size, ***trace-file*** **.0** is renamed ***trace-file*** **.1** and ***trace-file*** is renamed ***trace-file*** **.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.
- If you specify a maximum file size, you also must specify a maximum number of trace files with the files option and filename.

Syntax: ***x*** **k** to specify KB, ***x*** **m** to specify MB, or ***x*** **g** to specify GB

Range: 0 KB through 1 GB

Default: 128 KB

- **world-readable | no-world-readable**—(Optional) By default, log files can be accessed only by the user who configures the tracing operation. The **world-readable** option enables any user to read the file. To explicitly set the default behavior, use the **no-world-readable** option.
- **flag**—Trace operation or operations to perform on chassis cluster redundancy processes. To specify more than one trace operation, include multiple **flag** statements.
    - **all**—Trace all the events
        - **configuration**—Trace configuration events
        - **routing-socket**—Trace logging of rtsock activity
        - **snmp**—Trace SNMP events

Required Privilege Level

trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

weight

Supported Platforms SRX Series, vSRX

Syntax weight *number*;

Hierarchy Level [edit chassis cluster **redundancy-group** ***group-number*** interface-monitor ***interface***]
[edit chassis cluster **redundancy-group** ***group-number*** ip-monitoring *IP–address*]

Release Information Statement modified in Junos OS Release 10.1.

Description Specify the relative importance of the object to the operation of the redundancy group. This statement is primarily used with interface monitoring and IP address monitoring objects. The failure of an object—such as an interface—with a greater weight brings the group closer to failover. Every monitored object is assigned a weight.

- interface-monitor objects—If the object fails, its weight is deducted from the threshold of its redundancy group;
- ip-monitoring objects—If a monitored IP address becomes unreachable for any reason, the weight assigned to that monitored IP address is deducted from the redundancy group's global-threshold for IP address monitoring.

Every redundancy group has a default threshold of 255. If the threshold reaches 0, a failover is triggered. Failover is triggered even if the redundancy group is in manual failover mode and preemption is not enabled.

Options *number* —Weight assigned to the interface or monitored IP address. A higher weight value indicates a greater importance.

Range: 0 through 255

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- cluster (Chassis) on page 318

CHAPTER 26

Operational Commands

- show chassis cluster statistics
- show chassis cluster status
- show chassis environment (Security)
- show chassis environment cb
- show chassis ethernet-switch
- show chassis fabric plane
- show chassis fabric plane-location
- show chassis fabric summary
- show chassis hardware (View)
- show chassis routing-engine (View)
- show configuration chassis cluster traceoptions
- show security internal-security-association

clear chassis cluster control-plane statistics

Supported Platforms SRX Series, vSRX

Syntax **clear chassis cluster control-plane statistics**

Release Information Command introduced in Junos OS Release 9.3.

Description Clear the control plane statistics of a chassis cluster.

Required Privilege Level clear

Related Documentation

- show chassis cluster control-plane statistics on page 386

List of Sample Output clear chassis cluster control-plane statistics on page 359

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear chassis cluster control-plane statistics

```
user@host> clear chassis cluster control-plane statistics
Cleared control-plane statistics
```


clear chassis cluster data-plane statistics

Supported Platforms SRX Series, vSRX

Syntax **clear chassis cluster data-plane statistics**

Release Information Command introduced in Junos OS Release 9.3.

Description Clear the data plane statistics of a chassis cluster.

Required Privilege Level clear

Related Documentation

- show chassis cluster data-plane statistics on page 389

List of Sample Output clear chassis cluster data-plane statistics on page 360

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear chassis cluster data-plane statistics

```
user@host> clear chassis cluster data-plane statistics
Cleared data-plane statistics
```


clear chassis cluster failover-count

Supported Platforms SRX Series, vSRX

Syntax **clear chassis cluster failover-count**

Release Information Command introduced in Junos OS Release 9.3.

Description Clear the failover count of all redundancy-groups.

Required Privilege Level clear

Related Documentation

- request chassis cluster failover node on page 369
- request chassis cluster failover reset on page 370
- show chassis cluster status on page 414

List of Sample Output

show chassis cluster status on page 361
clear chassis cluster failover-count on page 361
show chassis cluster status on page 361

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

The following example displays the redundancy groups before and after the failover-counts are cleared.

show chassis cluster status

```
user@host> show chassis cluster status

Cluster ID: 3
Node name          Priority     Status     Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
   node0             200          secondary   no         no
   node1             100          primary     no         no

Redundancy group: 1 , Failover count: 2
   node0             100          primary     no         no
   node1             10           secondary   no         no
```

clear chassis cluster failover-count

```
user@host> clear chassis cluster failover-count
Cleared failover-count for all redundancy-groups
```

show chassis cluster status

```
user@host> show chassis cluster status

Cluster ID: 3
Node name         Priority    Status     Preempt  Manual failover
```



```
Redundancy group: 0 , Failover count: 0
  node0           200       secondary    no       no
  node1           100       primary      no       no

Redundancy group: 1 , Failover count: 0
  node0           100       primary      no       no
  node1           10        secondary    no       no
```

clear chassis cluster ip-monitoring failure-count

Supported Platforms SRX Series, vSRX

Syntax clear chassis cluster ip-monitoring failure-count

Release Information Command introduced in Junos OS Release 10.1.

Description Clear the failure count for all IP addresses.

Required Privilege Level clear

Related Documentation
- clear chassis cluster ip-monitoring failure-count
- clear chassis cluster ip-monitoring failure-count ip-address on page 364

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

user@host> **clear chassis cluster ip-monitoring failure-count**

```
node0:
--------------------------------------------------------------------------
Cleared failure count for all IPs

node1:
--------------------------------------------------------------------------
Cleared failure count for all IPs
```

clear chassis cluster ip-monitoring failure-count ip-address

Supported Platforms SRX Series, vSRX

Syntax clear chassis cluster ip-monitoring failure-count ip-address 1.1.1.1

Release Information Command introduced in Junos OS Release 10.1.

Description Clear the failure count for a specified IP address.

NOTE: Entering an IP address at the end of this command is optional. If you do not specify an IP address, the failure count for all monitored IP addresses is cleared.

Required Privilege Level clear

Related Documentation

- clear chassis cluster failover-count on page 361
- clear chassis cluster ip-monitoring failure-count on page 363

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

user@host> **clear chassis cluster ip-monitoring failure-count ip-address 1.1.1.1**

```
node0:
--------------------------------------------------------------------------
Cleared failure count for IP: 1.1.1.1

node1:
--------------------------------------------------------------------------
Cleared failure count for IP: 1.1.1.1
```


clear chassis cluster statistics

Supported Platforms SRX Series, vSRX

Syntax **clear chassis cluster statistics**

Release Information Command introduced in Junos OS Release 9.3.

Description Clear the control plane and data plane statistics of a chassis cluster.

Required Privilege Level clear

Related Documentation

- show chassis cluster statistics on page 410

List of Sample Output clear chassis cluster statistics on page 365

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear chassis cluster statistics

```
user@host> clear chassis cluster statistics
Cleared control-plane statistics
Cleared data-plane statistics
```

request chassis cb

Supported Platforms SRX Series, vSRX

Syntax request chassis cb (offline | online) slot *slot-number*

Release Information Command introduced in Junos OS Release 9.2.
Starting with Junos OS Release 12.1X47-D15, the SRX5K-SCBE (SCB2) is introduced and starting with Junos OS Release 15.1X49-D10, the SRX5K-SCB3 (SCB3) with enhanced midplanes is introduced.

Description SRX Series devices control the operation of the Control Board (CB).

Options **offline**—Take the Control Board offline.

online—Bring the Control Board online.

slot ***slot-number***—Control Board slot number.

Required Privilege Level maintenance

Related Documentation
- show chassis environment cb on page 421

List of Sample Output request chassis cb on page 366

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cb

```
user@host> request chassis cb offline slot 2 node local
node0:
--------------------------------------------------------------------------
Offline initiated, use "show chassis environment cb" to verify
```


request chassis cluster configuration-synchronize

Supported Platforms SRX Series, vSRX

Syntax request chassis cluster configuration-synchronize

Release Information Command introduced in Junos OS Release 12.1X47-D10.

Description Synchronizes the configuration from the primary node to the secondary node when the secondary node joins the primary node in a cluster.

Required Privilege Level maintenance

Related Documentation

- Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 161
- Verifying Chassis Cluster Configuration Synchronization Status on page 162
- NTP Time Synchronization on SRX Series Devices on page 163

List of Sample Output request chassis cluster configuration-synchronize on page 367

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster configuration-synchronize

```
user@host> request chassis cluster configuration-synchronize
Performing configuration synchronization from remote node.
```


request chassis cluster failover redundancy-group

Supported Platforms SRX Series, vSRX

Syntax request chassis cluster failover redundancy-group *redundancy-group-number*

Release Information Command introduced in Junos OS Release 9.0.

Description For chassis cluster configurations, initiate manual failover in a redundancy group from one node to the other, which becomes the primary node, and automatically reset the priority of the group to 255. The failover stays in effect until the new primary node becomes unavailable, the threshold of the redundancy group reaches 0, or you use the **request chassis cluster failover reset** command.

After a manual failover, you must use the **request chassis cluster failover reset** command before initiating another failover.

Options

- **node** ***node-number***-Number of the chassis cluster node to which the redundancy group fails over.
- **Range:** 0 through 1
- **redundancy-group** ***group-number*** —Number of the redundancy group on which to initiate manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

  Range: 0 through 255

Required Privilege Level maintenance

Related Documentation

- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 136
- Verifying Chassis Cluster Failover Status on page 138

List of Sample Output request chassis cluster failover redundancy-group 0 node 1 on page 368

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster failover redundancy-group 0 node 1

```
user@host> request chassis cluster failover redundancy-group 0 node 1
{primary:node0}
user@host> request chassis cluster failover redundancy-group 0 node 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Initiated manual failover for redundancy group 0
```


request chassis cluster failover node

Supported Platforms

Syntax

request chassis cluster failover node *node-number*
redundancy-group *group-number*

Release Information Command introduced in Junos OS Release 9.0.

Description For chassis cluster configurations, initiate manual failover in a redundancy group from one node to the other, which becomes the primary node, and automatically reset the priority of the group to 255. The failover stays in effect until the new primary node becomes unavailable, the threshold of the redundancy group reaches 0, or you use the **request chassis cluster failover reset** command.

After a manual failover, you must use the **request chassis cluster failover reset** command before initiating another failover.

Options

- **node** ***node-number***-Number of the chassis cluster node to which the redundancy group fails over.
- **Range:** 0 through 1
- **redundancy-group** ***group-number*** —Number of the redundancy group on which to initiate manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

  Range: 0 through 255

Required Privilege Level maintenance

Related Documentation

- clear chassis cluster failover-count on page 361
- request chassis cluster failover reset on page 370
- show chassis cluster status on page 414

List of Sample Output request chassis cluster failover node 0 redundancy-group 1 on page 369

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster failover node 0 redundancy-group 1

```
user@host> request chassis cluster failover node 0 redundancy-group 1
Initiated manual failover for redundancy group 1
```


request chassis cluster failover reset

Supported Platforms SRX Series, vSRX

Syntax

request chassis cluster failover reset
redundancy-group ***group-number***

Release Information Command introduced in Junos OS Release 9.0.

Description In chassis cluster configurations, undo the previous manual failover and return the redundancy group to its original settings.

Options **redundancy-group *group-number*** —Number of the redundancy group on which to reset manual failover. Redundancy group 0 is a special group consisting of the two Routing Engines in the chassis cluster.

Range: 0 through 255

Required Privilege Level maintenance

Related Documentation

- clear chassis cluster failover-count on page 361
- request chassis cluster failover node on page 369
- show chassis cluster status on page 414

List of Sample Output request chassis cluster failover reset redundancy-group 0 on page 370

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster failover reset redundancy-group 0

```
user@host> request chassis cluster failover reset redundancy-group 0
```


request chassis cluster in-service-upgrade abort (ISSU)

Supported Platforms SRX1500, SRX5400, SRX5600, SRX5800

Syntax request chassis cluster in-service-upgrade abort

Release Information Command introduced in Junos OS Release 11.2.

Description Abort an upgrade in a chassis cluster during an in-service software upgrade (ISSU). Use this command to end the ISSU on any nodes in a chassis cluster followed by **reboot** to abort the ISSU on that device.

Options This command has no options.

Required Privilege Level view

Related Documentation
- request system software in-service-upgrade (Maintenance) on page 379

List of Sample Output request chassis cluster in-service-upgrade abort on page 371

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request chassis cluster in-service-upgrade abort

```
user@host> request chassis cluster in-service-upgrade abort
Exiting in-service-upgrade window
Chassis ISSU Aborted
```

request security internal-security-association refresh

Supported Platforms SRX5400, SRX5600, SRX5800

Syntax request security internal-security-association refresh

Release Information Command introduced in Junos OS Release 12.1X45-D10.

Description Activate internal IPsec so an attacker cannot gain unauthorized information.

Required Privilege Level maintenance

Related Documentation

- show security internal-security-association on page 454
- internal (Security IPsec) on page 334

List of Sample Output request security internal-security-association refresh on page 372

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request security internal-security-association refresh

```
user@host> request security internal-security-association refresh
```


request system scripts add

Supported Platforms SRX Series

Syntax request system scripts add *package-name* no-copy | unlink
<master>
<backup>

Release Information Command introduced before Junos OS Release 9.0. The options **master** and **backup** are introduced in Junos OS Release 15.1X49-D50.

Description CLI command to install AI-Script install packages on SRX Series devices in chassis cluster.

Options **no-copy**—Do not save a copy of the AI script package file.

unlink—Remove the AI script package after successful installation.

master—Install AI script packages on the primary node.

backup—Install AI script packages on the secondary node.

Additional Information This command eliminates the AI script installation on both primary node and secondary node separately.

Required Privilege Level maintenance

Related Documentation

- Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 161

List of Sample Output request system scripts add package-name on page 373
request system scripts add package-name on page 374

Output Fields

Sample Output

request system scripts add package-name

```
user@host> request system scripts add jais-5.0R1.0-signed.tgz master

[: -a: unexpected operator
grep: /etc/db/pkg/jais/+COMMENT: No such file or directory
Installing package '/var/tmp/jais-5.0R4.0-signed.tgz' ...
Verified jais-5.0R4.0.tgz signed by PackageProductionRSA_2016
Adding jais...
Available space: 798414 require: 1814
Installing AI-Scripts version: 5.0R4
Saving package file in /var/db/scripts/commit/jais-5.0R4.0-signed.tgz ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
ln: ///etc/rc.d/ais: Read-only file system
Model: srx5600
Model: srx5600
```



```
Linking in Junos ES manifest file.
Creating srx5800/srx5600 trend data file.
Creating SRX intelligence attachments file.
Creating SRX events attachments file.
Creating AI-Scripts FIFO
Starting AI-Scripts FIFO handler
77834: old priority 0, new priority 20
77842: old priority 0, new priority 20
RSI parameters are now being set
BIOS validation parameter is now being set
BIOS interval parameter is now being set
JMB cleanup age is now being set
JMB Event file is now being set
JMB User Event file is now being set
PHDC collect parameter is now being set
PHDC duration parameter is now being set
PHDC commands file is now being set
JMB Progress Logging parameter is now being set
iJMB generation parameters are now being set
AI-Scripts platform support flag is now being set
Interval event commands file is now being set
Interval event enabled parameter is now being set
All node log collect parameter is now being set
Disk Warning Threshold is now being set
Disk Full Threshold is now being set
RSI Lite Enabled is now being set
Removing any old files that need to be updated
Copying updated files
Restarting eventd ...
Event processing process started, pid 78147
Installation completed
Saving package file in /var/sw/pkg/jais-5.0R4.0-signed.tgz ...
Saving state for rollback ...
```

request system scripts add package-name

```
user@host> request system scripts add jais-5.0R1.0-signed.tgz backup
Pushing bundle to node1
[: -a: unexpected operator
grep: /etc/db/pkg/jais/+COMMENT: No such file or directory
Installing package '/var/tmp/jais-5.0R4.0-signed.tgz' ...
Verified jais-5.0R4.0.tgz signed by PackageProductionRSA_2016
Adding jais...
Available space: 2619677 require: 1814
Installing AI-Scripts version: 5.0R4
chmod: /var/db/scripts/event/cron.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event2.slax: No such file or directory
chmod: /var/db/scripts/op/ais_bit.slax: No such file or directory
Saving package file in /var/db/scripts/commit/jais-5.0R4.0-signed.tgz ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
ln: ///etc/rc.d/ais: Read-only file system
Mounted jais package on /dev/md2...
Verified manifest signed by PackageProductionRSA_2016
Verified jais-5.0R4.0 signed by PackageProductionRSA_2016
Model: srx5600
Model: srx5600
Linking in Junos ES manifest file.
Creating srx5800/srx5600 trend data file.
Creating SRX intelligence attachments file.
```



```
Creating SRX events attachments file.
Creating AI-Scripts FIFO
Starting AI-Scripts FIFO handler
99423: old priority 0, new priority 20
99428: old priority 0, new priority 20
99429: old priority 0, new priority 20
99430: old priority 0, new priority 20
RSI parameters are now being set
BIOS validation parameter is now being set
BIOS interval parameter is now being set
JMB cleanup age is now being set
JMB Event file is now being set
JMB User Event file is now being set
PHDC collect parameter is now being set
PHDC duration parameter is now being set
PHDC commands file is now being set
JMB Progress Logging parameter is now being set
iJMB generation parameters are now being set
AI-Scripts platform support flag is now being set
Interval event commands file is now being set
Interval event enabled parameter is now being set
All node log collect parameter is now being set
Disk Warning Threshold is now being set
Disk Full Threshold is now being set
RSI Lite Enabled is now being set
chmod: /var/db/scripts/event/cron.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event2.slax: No such file or directory
chmod: /var/db/scripts/op/ais_bit.slax: No such file or directory
Removing any old files that need to be updated
Copying updated files
Restarting eventd ...
Event processing process started, pid 99730
Installation completed
Saving package file in /var/sw/pkg/jais-5.0R4.0-signed.tgz ...
Saving state for rollback ...
```


show system license (View)

Supported Platforms SRX Series, vSRX

Syntax

```
show system license
<installed | keys | status | usage>
```

Release Information Command introduced in Junos OS Release 9.5. Logical system status option added in Junos OS Release 11.2.

Description Display licenses and information about how licenses are used.

Options **none**—Display all license information.

installed—(Optional) Display installed licenses only.

keys—(Optional) Display a list of license keys. Use this information to verify that each expected license key is present.

status—(Optional) Display license status for a specified logical system or for all logical systems.

usage—(Optional) Display the state of licensed features.

Required Privilege Level view

Related Documentation

- *Verifying Junos OS License Installation*

List of Sample Output

show system license on page 377
show system license installed on page 377
show system license keys on page 378
show system license usage on page 378
show system license status logical-system all on page 378

Output Fields Table 30 on page 376 lists the output fields for the **show system license** command. Output fields are listed in the approximate order in which they appear.

Table 30: show system license Output Fields

Field Name	Field Description
Feature name	Name assigned to the configured feature. You use this information to verify that all the features for which you installed licenses are present.
Licenses used	Number of licenses used by the device. You use this information to verify that the number of licenses used matches the number configured. If a licensed feature is configured, the feature is considered used.

Table 30: show system license Output Fields *(continued)*

Field Name	Field Description
Licenses installed	Information about the installed license key: • **License identifier**—Identifier associated with a license key. • **License version**—Version of a license. The version indicates how the license is validated, the type of signature, and the signer of the license key. • **Valid for device**—Device that can use a license key. • **Features**—Feature associated with a license.
Licenses needed	Number of licenses required for features being used but not yet properly licensed.
Expiry	Time remaining in the grace period before a license is required for a feature being used.
Logical system license status	Displays whether a license is enabled for a logical system.

Sample Output

show system license

```
user@host> show system license

License usage:
                                 Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  av_key_kaspersky_engine               1            1           0    2012-03-30
 01:00:00 IST
  wf_key_surfcontrol_cpa                0            1           0    2012-03-30
 01:00:00 IST
  dynamic-vpn                           0            1           0    permanent
  ax411-wlan-ap                         0            2           0    permanent

Licenses installed:
  License identifier: JUNOS301998
  License version: 2
  Valid for device: AG4909AA0080
  Features:
    av_key_kaspersky_engine - Kaspersky AV
      date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST

  License identifier: JUNOS302000
  License version: 2
  Valid for device: AG4909AA0080
  Features:
    wf_key_surfcontrol_cpa - Web Filtering
      date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST
```

show system license installed

```
user@host> show system license installed

License identifier: JUNOS301998
  License version: 2
  Valid for device: AG4909AA0080
  Features:
```

```
    av_key_kaspersky_engine - Kaspersky AV
      date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST

  License identifier: JUNOS302000
  License version: 2
  Valid for device: AG4909AA0080
  Features:
    wf_key_surfcontrol_cpa - Web Filtering
      date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST
```

show system license keys

```
user@host> show system license keys

XXXXXXXXXX xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx
         xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx
         xxxxxx xxxxxx xxx
```

show system license usage

```
user@host> show system license usage

                                   Licenses     Licenses    Licenses  Expiry
Feature name                           used    installed      needed
  av_key_kaspersky_engine                 1            1           0    2012-03-30
 01:00:00 IST
  wf_key_surfcontrol_cpa                  0            1           0    2012-03-30
 01:00:00 IST
  dynamic-vpn                             0            1           0     permanent
  ax411-wlan-ap                           0            2           0     permanent
```

show system license status logical-system all

```
user@host> show system license status logical-system all
Logical system license status:

logical system name                  license status
root-logical-system                  enabled
LSYS0                                enabled
LSYS1                                enabled
LSYS2                                enabled
```


request system software in-service-upgrade (Maintenance)

Supported Platforms SRX1500, SRX300, SRX320, SRX340, SRX345, SRX5400, SRX550M, SRX5600, SRX5800

Syntax

request system software in-service-upgrade *image_name*
<no-copy>
<no-sync>
<no-tcp-syn-check>
<no-validate>
<reboot>
<unlink>

Release Information For SRX5400, SRX5600, and SRX5800 devices, command introduced in Junos OS Release 9.6 and support for **reboot** as a required parameter added in Junos OS Release 11.2R2. For SRX5400 devices, the command is introduced in Junos OS Release 12.1X46-D20. For SRX300, SRX320, SRX340, and SRX345 devices, command introduced in Junos OS Release 15.1X49-D40. For SRX1500 devices, command introduced in Junos OS Release 15.1X49-D50.

Description The in-service software upgrade (ISSU) feature allows a chassis cluster pair to be upgraded from supported Junos OS versions with a traffic impact similar to that of redundancy group failovers. Before upgrading, you must perform failovers so that all redundancy groups are active on only one device. We recommend that graceful restart for routing protocols be enabled before you initiate an ISSU.

For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, you must use the **no-sync** parameter to perform an in-band cluster upgrade (ICU). This allows a chassis cluster pair to be upgraded with a minimal service disruption of approximately 30 seconds.

For SRX1500, SRX4100, and SRX4200 devices, the **no-sync** parameter is not supported when using ISSU to upgrade. The **no-sync** option specifies that the state is not synchronized from the primary node to the secondary node.

For SRX1500 devices, the **no-tcp-syn-check** parameter is not supported when using ISSU to upgrade.

Options

- *image_name*—Location and name of the software upgrade package to be installed.
- **no-copy**—(Optional) Installs the software upgrade package but does not save the copies of package files.

  NOTE: This option is not supported on SRX1500 devices.

- **no-sync**—Stops the flow state from synchronizing when the old secondary node has booted with a new Junos OS image.

  This parameter applies to SRX300, SRX320, SRX340, SRX345, and SRX550M devices only. It is required for an ICU.

NOTE: This option is not supported on SRX1500 devices.

- **no-tcp-syn-check**—(Optional) Creates a window wherein the TCP SYN check for the incoming packets is disabled. The default value for the window is 7200 seconds (2 hours).

  This parameter applies to SRX300, SRX320, SRX340, SRX345, and SRX550M devices only.

NOTE: This option is not supported on SRX1500 devices.

- **no-validate**—(Optional) Disables the configuration validation step at installation. The system behavior is similar to that of the **request system software add** command.

  This parameter applies to SRX300, SRX320, SRX340, SRX345, and SRX550M devices only.

- **reboot**—Reboots each device in the chassis cluster pair after installation is completed.

  This parameter applies to SRX5400, SRX5600, and SRX5800 devices only. It is required for an ISSU. (The devices in a cluster are automatically rebooted following an ICU.)

NOTE: This option is not supported on SRX1500 devices.

- **unlink**—(Optional) Removes the software package after successful installation.

Required Privilege Level maintenance

Related Documentation

- request system software rollback (SRX Series) on page 384

List of Sample Output

request system software in-service-upgrade (High-End SRX Series Devices) on page 380
request system software in-service-upgrade (SRX300, SRX320, SRX340, SRX345, and SRX550M devices) on page 381

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

request system software in-service-upgrade (High-End SRX Series Devices)

```
user@host> request system software in-service-upgrade
/var/tmp/junos-srx1k3k-11.2R2.5-domestic.tgz no-copy reboot
Chassis ISSU Started
node0:
--------------------------------------------------------------------------
```



```
Chassis ISSU Started
ISSU: Validating Image
Inititating in-service-upgrade

node0:
--------------------------------------------------------------------------
Inititating in-service-upgrade
Checking compatibility with configuration
mgd: commit complete
Validation succeeded
ISSU: Preparing Backup RE
Finished upgrading secondary node node0
Rebooting Secondary Node

node0:
--------------------------------------------------------------------------
Shutdown NOW!
[pid 3257]
ISSU: Backup RE Prepare Done
Waiting for node0 to reboot.
node0 booted up.
Waiting for node0 to become secondary
node0 became secondary.
Waiting for node0 to be ready for failover
ISSU: Preparing Daemons
Secondary node0 ready for failover.
Failing over all redundancy-groups to node0
ISSU: Preparing for Switchover
Initiated failover for all the redundancy groups to node1
Waiting for node0 take over all redundancy groups

Exiting in-service-upgrade window

node0:
--------------------------------------------------------------------------
Exiting in-service-upgrade window
Exiting in-service-upgrade window
Chassis ISSU Aborted

node0:
--------------------------------------------------------------------------
Chassis ISSU Ended
ISSU completed successfully, rebooting...
Shutdown NOW!
[pid 4294]
```

Sample Output

request system software in-service-upgrade (SRX300, SRX320, SRX340, SRX345, and SRX550M devices)

user@host> **request system software in-service-upgrade /var/tmp/junos-srxsme-15.1I20160520_0757-domestic.tgz no-sync**

```
ISSU: Validating package
WARNING: in-service-upgrade shall reboot both the nodes
         in your cluster. Please ignore any subsequent
         reboot request message
ISSU: start downloading software package on secondary node
Pushing /var/tmp/junos-srxsme-15.1I20160520_0757-domestic.tgz to
node0:/var/tmp/junos-srxsme-15.1I20160520_0757-domestic.tgz
Formatting alternate root (/dev/da0s1a)...
```

```
/dev/da0s1a: 2510.1MB (5140780 sectors) block size 16384, fragment size 2048
        using 14 cylinder groups of 183.62MB, 11752 blks, 23552 inodes.
super-block backups (for fsck -b #) at:
32, 376096, 752160, 1128224, 1504288, 1880352, 2256416, 2632480, 3008544,
3384608, 3760672, 4136736, 4512800, 4888864
Installing package
'/altroot/cf/packages/install-tmp/junos-15.1I20160520_0757-domestic' ...
Verified junos-boot-srxsme-15.1I20160520_0757.tgz signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256
Verified junos-srxsme-15.1I20160520_0757-domestic signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256

WARNING:     The software that is being installed has limited support.
WARNING:     Run 'file show /etc/notices/unsupported.txt' for details.

Verified junos-boot-srxsme-15.1I20160520_0757.tgz signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256
Verified junos-srxsme-15.1I20160520_0757-domestic signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256
JUNOS 15.1I20160520_0757 will become active at next reboot
WARNING: A reboot is required to load this software correctly
WARNING:     Use the 'request system reboot' command
WARNING:         when software installation is complete
cp: cannot overwrite directory /altroot/cf/etc/ssh with non-directory /cf/etc/ssh
Saving state for rollback ...
ISSU: finished upgrading on secondary node node0
ISSU: start upgrading software package on primary node
Formatting alternate root (/dev/da0s1a)...
/dev/da0s1a: 2510.1MB (5140780 sectors) block size 16384, fragment size 2048
        using 14 cylinder groups of 183.62MB, 11752 blks, 23552 inodes.
super-block backups (for fsck -b #) at:
32, 376096, 752160, 1128224, 1504288, 1880352, 2256416, 2632480, 3008544,
3384608, 3760672, 4136736, 4512800, 4888864
Installing package
'/altroot/cf/packages/install-tmp/junos-15.1I20160520_0757-domestic' ...
Verified junos-boot-srxsme-15.1I20160520_0757.tgz signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256
Verified junos-srxsme-15.1I20160520_0757-domestic signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256

WARNING:     The software that is being installed has limited support.
WARNING:     Run 'file show /etc/notices/unsupported.txt' for details.

Verified junos-boot-srxsme-15.1I20160520_0757.tgz signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256
Verified junos-srxsme-15.1I20160520_0757-domestic signed by
PackageDevelopmentEc_2016 method ECDSA256+SHA256
JUNOS 15.1I20160520_0757 will become active at next reboot
WARNING: A reboot is required to load this software correctly
WARNING:     Use the 'request system reboot' command
WARNING:         when software installation is complete
cp: cannot overwrite directory /altroot/cf/etc/ssh with non-directory /cf/etc/ssh
Saving state for rollback ...
ISSU: failover all redundancy-groups 1...n to primary node
Successfully reset all redundancy-groups priority back to configured priority.
Successfully reset all redundancy-groups priority back to configured priority.
error: Command failed. None of the redundancy-groups has been failed over.
        Some redundancy-groups' priority on node1 are 0.
        e.g.: priority of redundancy-groups-1 on node1 is 0.
Use 'force' option at the end to ignore this check.
WARNING: Using force option may cause traffic loss.
```



```
ISSU: rebooting Secondary Node
Shutdown NOW!
ISSU: Waiting for secondary node node0 to reboot.
ISSU: node 0 went down
ISSU: Waiting for node 0 to come up
ISSU: node 0 came up
ISSU: secondary node node0 booted up.
ISSU: failover all redundancy-groups 1...n to remote node, before reboot.
Successfully reset all redundancy-groups priority back to configured priority.

Shutdown NOW!

{primary:node1}
user@host>

*** FINAL System shutdown message from user@host ***

System going down IMMEDIATELY
```

request system software rollback (SRX Series)

Supported Platforms SRX Series, vSRX

Syntax

```
request system software rollback
<node-id>
```

Release Information Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 15.1X49-D50 for SRX1500 devices.

Description Revert to the software that was loaded at the last successful **request system software add** command. Example: **request system software rollback**.

Options *node-id*—Identification number of the chassis cluster node. It can be 0 or 1.

Required Privilege Level maintenance

Related Documentation

- *request system reboot*

set chassis cluster cluster-id node reboot

Supported Platforms SRX Series, vSRX

Syntax set chassis cluster cluster-id *cluster-id* node *node* reboot

Release Information Support for extended cluster identifiers (more than 15 identifiers) added in Junos OS Release 12.1X45-D10.

Description This operational mode command sets the chassis cluster identifier (ID) and node ID on each device, and reboots the devices to enable clustering. 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.

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.

NOTE: 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 or later and re-create a cluster with cluster IDs greater than 16. 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. If the cluster ID set is less than 16 and you roll back to a previous release, the system comes back with the previous setup.

Options **cluster-id *cluster-id*** —Identifies the cluster within the Layer 2 domain.

Range: 0 through 255

node *node* —Identifies a node within a cluster.

Range: 0 to 1

Required Privilege Level maintenance

Related Documentation

- *Example: Setting the Chassis Cluster Node ID and Cluster ID for Branch SRX Series Devices*
- Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices on page 56
- *Understanding the Interconnect Logical System and Logical Tunnel Interfaces*
- *Example: Configuring Logical Systems in an Active/Passive Chassis Cluster (Master Administrators Only)*

Output Fields When you enter this command, you are provided feedback on the status of your request.

show chassis cluster control-plane statistics

Supported Platforms	SRX Series, vSRX
Syntax	**show chassis cluster control-plane statistics**
Release Information	Command introduced in Junos OS Release 9.3. Output changed to support dual control ports in Junos OS Release 10.0.
Description	Display information about chassis cluster control plane statistics.
Required Privilege Level	view
Related Documentation	• clear chassis cluster control-plane statistics on page 359
List of Sample Output	show chassis cluster control-plane statistics on page 387 show chassis cluster control-plane statistics (SRX5000 line devices) on page 387
Output Fields	Table 31 on page 386 lists the output fields for the **show chassis cluster control-plane statistics** command. Output fields are listed in the approximate order in which they appear.

Table 31: show chassis cluster control-plane statistics Output Fields

Field Name	Field Description
Control link statistics	Statistics of the control link used by chassis cluster traffic. Statistics for **Control link 1** are displayed when you use dual control links (SRX5600 and SRX5800 devices only). • **Heartbeat packets sent**—Number of heartbeat messages sent on the control link. • **Heartbeat packets received**—Number of heartbeat messages received on the control link. • **Heartbeat packet errors**—Number of heartbeat packets received with errors on the control link.
Fabric link statistics	Statistics of the fabric link used by chassis cluster traffic. Statistics for **Child Link 1** are displayed when you use dual fabric links. • **Probes sent**—Number of probes sent on the fabric link. • **Probes received**—Number of probes received on the fabric link.
Switch fabric link statistics	Statistics of the switch fabric link used by chassis cluster traffic. • **Probe state**—State of the probe, **UP** or **DOWN**. • **Probes sent**—Number of probes sent. • **Probes received**—Number of probes received. • **Probe recv error** —Error in receiving probe. • **Probe send error**—Error in sending probe.

Sample Output

show chassis cluster control-plane statistics

```
user@host> show chassis cluster control-plane statistics
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 11646
        Heartbeat packets received: 8343
        Heartbeat packet errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 11644
        Probes received: 8266
   Switch fabric link statistics:
    Probe state : DOWN
    Probes sent: 8145
    Probes received: 8013
    Probe recv errors: 0
    Probe send errors: 0
```

Sample Output

show chassis cluster control-plane statistics (SRX5000 line devices)

```
user@host> show chassis cluster control-plane statistics
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258698
        Heartbeat packets received: 258693
        Heartbeat packet errors: 0
   Control link 1:
        Heartbeat packets sent: 258698
        Heartbeat packets received: 258693
        Heartbeat packet errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258690
        Probes received: 258690
    Child link 1
        Probes sent: 258505
        Probes received: 258505
```

show chassis cluster data-plane interfaces

Supported Platforms SRX Series, vSRX

Syntax **show chassis cluster data-plane interfaces**

Release Information Command introduced in Junos OS Release 10.2.

Description Display the status of the data plane interface (also known as a fabric interface) in a chassis cluster configuration.

Required Privilege Level view

Related Documentation

- cluster (Chassis) on page 318

List of Sample Output show chassis cluster data-plane interfaces on page 388

Output Fields Table 32 on page 388 lists the output fields for the **show chassis cluster data-plane interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 32: show chassis cluster data-plane interfaces Output Fields

Field Name	Field Description
fab0/fab1	Name of the logical fabric interface. • **Name**—Name of the physical Ethernet interface. • **Status**—State of the fabric interface: **up** or **down**.

Sample Output

show chassis cluster data-plane interfaces

```
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
```


show chassis cluster data-plane statistics

Supported Platforms SRX Series, vSRX

Syntax **show chassis cluster data-plane statistics**

Release Information Command introduced in Junos OS Release 9.3.

Description Display information about chassis cluster data plane statistics.

Required Privilege Level view

Related Documentation

- clear chassis cluster data-plane statistics on page 360

List of Sample Output show chassis cluster data-plane statistics on page 390

Output Fields Table 33 on page 389 lists the output fields for the **show chassis cluster data-plane statistics** command. Output fields are listed in the approximate order in which they appear.

Table 33: show chassis cluster data-plane statistics Output Fields

Field Name	Field Description
Services Synchronized	• **Service name**—Name of the service. • **Rtos sent**—Number of runtime objects (RTOs) sent. • **Rtos received**—Number of RTOs received. • **Translation context**—Messages synchronizing Network Address Translation (NAT) translation context. • **Incoming NAT**—Messages synchronizing incoming Network Address Translation (NAT) service. • **Resource manager**—Messages synchronizing resource manager groups and resources. • **Session create**—Messages synchronizing session creation. • **Session close**—Messages synchronizing session close. • **Session change**—Messages synchronizing session change. • **Gate create**—Messages synchronizing creation of pinholes (temporary openings in the firewall). • **Session ageout refresh request**—Messages synchronizing request session after age-out. • **Session ageout refresh reply**—Messages synchronizing reply session after age-out. • **IPsec VPN**—Messages synchronizing VPN session. • **Firewall user authentication**—Messages synchronizing firewall user authentication session. • **MGCP ALG**—Messages synchronizing MGCP ALG sessions. • **H323 ALG**—Messages synchronizing H.323 ALG sessions. • **SIP ALG**—Messages synchronizing SIP ALG sessions. • **SCCP ALG**—Messages synchronizing SCCP ALG sessions. • **PPTP ALG**—Messages synchronizing PPTP ALG sessions. • **RTSP ALG**—Messages synchronizing RTSP ALG sessions.

Sample Output

show chassis cluster data-plane statistics

```
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
```

show chassis cluster ethernet-switching interfaces

Supported Platforms SRX1500, SRX550M, vSRX

Syntax show chassis cluster ethernet-switching interfaces

Release Information Command introduced in Junos OS Release 11.1.

Description Display the status of the switch fabric interfaces (swfab) in a chassis cluster.

Required Privilege Level view

Related Documentation

- cluster (Chassis) on page 318
- *Ethernet Switching and Layer 2 Transparent Mode Feature Guide for Security Devices*

List of Sample Output show chassis cluster ethernet-switching interfaces on page 391

Output Fields Table 34 on page 391 lists the output fields for the **show chassis cluster ethernet-switching interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 34: show chassis cluster ethernet-switching interfaces Output Fields

Field Name	Field Description
swfab0/swfab1	Name of the switch fabric interface. • Name—Name of the physical interface. • Status—State of the swfab interface: up or down.

Sample Output

show chassis cluster ethernet-switching interfaces

```
user@host> show chassis cluster ethernet-switching interfaces
swfab0:
    Name                  Status
    ge-0/0/9              up
    ge-0/0/10             up
swfab1:
    Name                  Status
    ge-5/0/9              up
    ge-5/0/10             up
```

show chassis cluster ethernet-switching status

Supported Platforms SRX1500, SRX550M, vSRX

Syntax show chassis cluster ethernet-switching status

Release Information Command introduced in Junos OS Release 11.1.

Description Display the Ethernet switching status of the chassis cluster.

Required Privilege Level view

Related Documentation

- cluster (Chassis) on page 318
- *Ethernet Switching and Layer 2 Transparent Mode Feature Guide for Security Devices*

List of Sample Output show chassis cluster ethernet-switching status on page 393

Output Fields Table 35 on page 392 lists the output fields for the **show chassis cluster ethernet-switching status** command. Output fields are listed in the approximate order in which they appear.

Table 35: show chassis cluster ethernet-switching status Output Fields

Field Name	Field Description
Cluster ID	ID number (1-255) of a cluster. Setting a cluster ID to 0 is equivalent to disabling a cluster. More than 16 cluster IDs will work only if the fabric and control link interfaces are connected back-to-back. NOTE: 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. NOTE: 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.
Redundancy-Group	ID number (1-255) of a redundancy group in the chassis cluster.
Node name	Node (device) in the chassis cluster (**node0** or **node1**).
Priority	Assigned priority for the redundancy group on that node.

Table 35: show chassis cluster ethernet-switching status Output Fields *(continued)*

Field Name	Field Description
Status	State of the redundancy group (**Primary**, **Secondary**, **Lost**, or **Unavailable**). • **Primary**—Redundancy group is active and passing traffic. • **Secondary**—Redundancy group is passive and not passing traffic. • **Lost**—Node loses contact with the other node through the control link. Most likely to occur when both nodes are in a cluster and due to control link failure, one node cannot exchange heartbeats, or when the other node is rebooted. • **Unavailable**—Node has not received a single heartbeat over the control link from the other node since the other node booted up. Most likely to occur when one node boots up before the other node, or if only one node is present in the cluster.
Preempt	• **Yes**: Mastership can be preempted based on priority. • **No**: Change in priority will not preempt mastership.
Manual failover	• **Yes**: If mastership is set manually through the CLI. • **No**: Mastership is not set manually through the CLI.

Sample Output

show chassis cluster ethernet-switching status

```
user@host> show chassis cluster ethernet-switching status
Cluster ID: 10
Node                  Priority          Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
    node0                   1           primary        no       no
    node1                   0           lost           n/a      n/a

Switch fabric link statistics:
    Probe state : DOWN
    Probes sent: 8145
    Probes received: 8013
    Probe recv errors: 0
    Probe send errors: 0
```


show chassis cluster information

Supported Platforms SRX Series, vSRX

Syntax show chassis cluster information

Release Information Command introduced in Junos OS Release 12.1X47-D10.

Description Display chassis cluster messages. The messages indicate each node's health condition and details of the monitored failure.

Required Privilege Level view

Related Documentation
- show chassis cluster status on page 414

List of Sample Output show chassis cluster information on page 394
show chassis cluster information on page 395

Output Fields Table 36 on page 394 lists the output fields for the **show chassis cluster information** command. Output fields are listed in the approximate order in which they appear.

Table 36: show chassis cluster information Output Fields

Field Name	Field Description
Node	Node (device) in the chassis cluster (**node0** or **node1**).
Redundancy Group Information	• Redundancy Group—ID number (0 - 255) of a redundancy group in the cluster. • Current State—State of the redundancy group: **primary**, **secondary**, **hold**, or **secondary-hold**. • Weight—Relative importance of the redundancy group. • Time—Time when the redundancy group changed the state. • From—State of the redundancy group before the change. • To—State of the redundancy group after the change. • Reason—Reason for the change of state of the redundancy group.
Chassis cluster LED information	• Current LED color—Current color state of the LED. • Last LED change reason—Reason for change of state of the LED.

Sample Output

show chassis cluster information

```
user@host> show chassis cluster information

node0:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: primary, Weight: 255
```



```
        Time             From           To             Reason
        Mar 27 17:44:19 hold           secondary      Hold timer expired
        Mar 27 17:44:27 secondary      primary        Better priority (200/200)

    Redundancy Group 1 , Current State: primary, Weight: 255

        Time             From           To             Reason
        Mar 27 17:44:19 hold           secondary      Hold timer expired
        Mar 27 17:44:27 secondary      primary        Remote yield (0/0)

    Redundancy Group 2 , Current State: secondary, Weight: 255

        Time             From           To             Reason
        Mar 27 17:44:19 hold           secondary      Hold timer expired
        Mar 27 17:44:27 secondary      primary        Remote yield (0/0)
        Mar 27 17:50:24 primary        secondary-hold Preempt/yield(100/200)
        Mar 27 17:50:25 secondary-hold secondary      Ready to become secondary

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
        Mar 27 17:44:27 hold           secondary      Hold timer expired

    Redundancy Group 1 , Current State: secondary, Weight: 255

        Time             From           To             Reason
        Mar 27 17:44:27 hold           secondary      Hold timer expired
        Mar 27 17:50:23 secondary      primary        Remote yield (100/0)
        Mar 27 17:50:24 primary        secondary-hold Preempt/yield(100/200)
        Mar 27 17:50:25 secondary-hold secondary      Ready to become secondary

    Redundancy Group 2 , Current State: primary, Weight: 255

        Time             From           To             Reason
        Mar 27 17:44:27 hold           secondary      Hold timer expired
        Mar 27 17:50:23 secondary      primary        Remote yield (200/0)

Chassis cluster LED information:
    Current LED color: Green
    Last LED change reason: No failures
```

Sample Output

show chassis cluster information

`user@host>` **show chassis cluster information**

The following output is specific to monitoring abnormal (unhealthy) case.

```
node0:
--------------------------------------------------------------------------
Redundancy Group Information:
```

```
    Redundancy Group 0 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:07:38 hold           secondary      Hold timer expired
        Apr  1 11:07:41 secondary      primary        Only node present
        Apr  1 11:29:20 primary        secondary-hold Manual failover
        Apr  1 11:34:20 secondary-hold secondary      Ready to become secondary

    Redundancy Group 1 , Current State: primary, Weight: 0

        Time            From           To             Reason
        Apr  1 11:07:38 hold           secondary      Hold timer expired
        Apr  1 11:07:41 secondary      primary        Only node present

    Redundancy Group 2 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:07:38 hold           secondary      Hold timer expired
        Apr  1 11:07:41 secondary      primary        Only node present

Chassis cluster LED information:
    Current LED color: Amber
    Last LED change reason: Monitored objects are down

Failure Information:

    IP Monitoring Failure Information:
    Redundancy Group 1, Monitoring Status: Failed
    IP Address        Status       Reason
    1.1.1.1           Unreachable  redundancy-group state unknown

node1:
--------------------------------------------------------------------------
Redundancy Group Information:

    Redundancy Group 0 , Current State: primary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:08:40 hold           secondary      Hold timer expired
        Apr  1 11:29:20 secondary      primary        Remote is in secondary hold


    Redundancy Group 1 , Current State: secondary, Weight: 0

        Time            From           To             Reason
        Apr  1 11:08:40 hold           secondary      Hold timer expired

    Redundancy Group 2 , Current State: secondary, Weight: 255

        Time            From           To             Reason
        Apr  1 11:08:40 hold           secondary      Hold timer expired

Chassis cluster LED information:
    Current LED color: Amber
    Last LED change reason: Monitored objects are down

Failure Information:

    IP Monitoring Failure Information:
    Redundancy Group 1, Monitoring Status: Failed
```



```
IP Address        Status       Reason
1.1.1.1           Unreachable  redundancy-group state unknown
```

show chassis cluster information configuration-synchronization

Supported Platforms SRX Series, vSRX

Syntax show chassis cluster information configuration-synchronization

Release Information Command introduced in Junos OS Release 12.1X47-D10.

Description Display chassis cluster messages. The messages indicate the redundancy mode, automatic synchronization status, and if automatic synchronization is enabled on the device.

Required Privilege Level view

Related Documentation

- Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 161
- NTP Time Synchronization on SRX Series Devices on page 163
- Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 163
- **request chassis cluster configuration-synchronize on page 367**

List of Sample Output **show chassis cluster information configuration-synchronization on page 398**

Output Fields Table 37 on page 398 lists the output fields for the **show chassis cluster information configuration-synchronization** command. Output fields are listed in the approximate order in which they appear.

Table 37: show chassis cluster information configuration-synchronization Output Fields

Field Name	Field Description
Node name	Node (device) in the chassis cluster (**node0** or **node1**).
Status	• Activation status—State of automatic configuration synchronization: **Enabled** or **Disabled**. • Last sync operation—Status of the last synchronization. • Last sync result—Result of the last synchronization. • Last sync mgd messages—Management daemon messages of the last synchronization.
Events	The timestamp of the event, the automatic configuration synchronization status, and the number of synchronization attempts.

Sample Output

show chassis cluster information configuration-synchronization

```
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:
     Feb 25 22:21:49.174 : 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
       Network security daemon: warning: You have enabled/disabled inet6 flow.
       Network security daemon: You must reboot the system for your change to
take effect.
      Network security daemon: If you have deployed a cluster, be sure to reboot
 all nodes.
       mgd: commit complete
   Events:
     Feb 25 23:02:33.467 : Auto-Sync: In progress. Attempt: 1
     Feb 25 23:03:13.200 : Auto-Sync: Succeeded. Attempt: 1
```

show chassis cluster information issu

Supported Platforms	SRX1500, SRX5400, SRX5600, SRX5800
Syntax	show chassis cluster information issu
Release Information	Command introduced in Junos OS Release 12.1X47-D10.
Description	Display chassis cluster messages. The messages indicate the progress of the in-service software upgrade (ISSU).
Required Privilege Level	view
Related Documentation	• show chassis cluster status on page 414
List of Sample Output	show chassis cluster information issu on page 400
Output Fields	Table 38 on page 400 lists the output fields for the **show chassis cluster information issu** command. Output fields are listed in the approximate order in which they appear.

Table 38: show chassis cluster information issu Output Fields

Field Name	Field Description
Node name	Node (device) in the chassis cluster (**node0** or **node1**).
CS Prereq	Status of all cold synchronization prerequisites: • if_state sync—Status of if_state synchronization. • fabric link—Status of fabric link synchronization. • policy data sync—Status of policy data synchronization. • cp ready—Status of the central point. • VPN data sync—Status of the VPN data synchronization.
CS RTO sync	Status of cold synchronization runtime objects.
CS postreq	Status of cold synchronization postrequirements.

Sample Output

show chassis cluster information issu

```
user@host> show chassis cluster information issu

node0:
--------------------------------------------------------------------------
Cold Synchronization Progress:
    CS Prereq               10 of 10 SPUs completed
       1. if_state sync          10 SPUs completed
       2. fabric link            10 SPUs completed
       3. policy data sync       10 SPUs completed
       4. cp ready               10 SPUs completed
```

```
      5. VPN data sync         10 SPUs completed
   CS RTO sync           10 of 10 SPUs completed
   CS Postreq            10 of 10 SPUs completed

node1:
--------------------------------------------------------------------------
Cold Synchronization Progress:
    CS Prereq              10 of 10 SPUs completed
       1. if_state sync        10 SPUs completed
       2. fabric link          10 SPUs completed
       3. policy data sync     10 SPUs completed
       4. cp ready             10 SPUs completed
       5. VPN data sync        10 SPUs completed
    CS RTO sync            10 of 10 SPUs completed
    CS Postreq             10 of 10 SPUs completed
```

show chassis cluster interfaces

Supported Platforms SRX Series, vSRX

Syntax **show chassis cluster interfaces**

Release Information Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0. Output changed to support control interfaces in Junos OS Release 11.2. Output changed to support redundant pseudo interfaces in Junos OS Release 12.1X44-D10. For high-end SRX Series devices, output changed to support the internal security association (SA) option in Junos OS Release 12.1X45-D10. Output changed to support MACsec status on control and fabric interfaces in Junos OS Release 15.1X49-D60.

Description Display the status of the control interface in a chassis cluster configuration.

Required Privilege Level view

Related Documentation

- cluster (Chassis) on page 318

List of Sample Output

show chassis cluster interfaces on page 403
show chassis cluster interfaces (SRX5000 line devices) on page 404
show chassis cluster interfaces on page 405
show chassis cluster interfaces(SRX5400, SRX5600, and SRX5800 devices with SRX5000 line SRX5K-SCB3 (SCB3) with enhanced midplanes and SRX5K-MPC3-100G10G (IOC3) or SRX5K-MPC3-40G10G (IOC3)) on page 405

Output Fields Table 39 on page 402 lists the output fields for the **show chassis cluster interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 39: show chassis cluster interfaces Output Fields

Field Name	Field Description
Control link status	State of the chassis cluster control interface: **up** or **down**.
Control interfaces	• **Index**—Index number of the chassis cluster control interface. • **Name**—Name of the chassis cluster control interface. • **Monitored-Status**—Monitored state of the interface: **up** or **down**. • **Internal SA**—State of the internal SA option on the chassis cluster control link: **enabled** or **disabled**. NOTE: This field is available only on high-end SRX Series devices. • **Security**—State of MACsec on chassis cluster control interfaces.
Fabric link status	State of the fabric interface: **up** or **down**.

Table 39: show chassis cluster interfaces Output Fields *(continued)*

Field Name	Field Description
Fabric interfaces	• **Name**—Name of the fabric interface. • **Child-interface**—Name of the child fabric interface. • **Status**—State of the interface: **up** or **down**. • **Security**—State of MACsec on chassis cluster fabric interfaces.
Redundant-ethernet Information	• **Name**—Name of the redundant Ethernet interface. • **Status**—State of the interface: **up** or **down**. • **Redundancy-group**—Identification number (1–255) of the redundancy group associated with the redundant Ethernet interface.
Redundant-pseudo-interface Information	• **Name**—Name of the redundant pseudointerface. • **Status**—State of the redundant pseudointerface: **up** or **down**. • **Redundancy-group**—Identification number (1–255) of the redundancy group associated with the redundant pseudointerface.
Interface Monitoring	• **Interface**—Name of the interface to be monitored. • **Weight**—Relative importance of the interface to redundancy group operation. • **Status**—State of the interface: **up** or **down**. • **Redundancy-group**—Identification number of the redundancy group associated with the interface.

Sample Output

show chassis cluster interfaces

```
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-0/1/0           Up       Disabled
    fab0
    fab1     ge-6/1/0           Up       Disabled
    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
```



```
    reth8         Down        Not configured
    reth9         Down        Not configured
    reth10        Down        Not configured
    reth11        Down        Not configured

Redundant-pseudo-interface Information:
    Name         Status       Redundancy-group
     lo0          Up                1

Interface Monitoring:
    Interface         Weight    Status    Redundancy-group
    ge-0/1/9          100       Up        0
    ge-0/1/9          100       Up
```

Sample Output

show chassis cluster interfaces (SRX5000 line devices)

```
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         Down               Disabled      Disabled
Fabric link status: Up

Fabric interfaces:
    Name    Child-interface    Status                    Security
                               (Physical/Monitored)
    fab0    xe-1/0/3           Up    / Down              Disabled
    fab0
    fab1    xe-7/0/3           Up    / Down              Disabled
    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
    reth8         Down        Not configured
    reth9         Down        Not configured
    reth10        Down        Not configured
    reth11        Down        Not configured

Redundant-pseudo-interface Information:
    Name         Status       Redundancy-group
     lo0          Up                1

Interface Monitoring:
    Interface         Weight    Status    Redundancy-group
    ge-0/1/9          100       Up        0
    ge-0/1/9          100       Up
```


Sample Output

show chassis cluster interfaces

user@host> **show chassis cluster interfaces**

The following output is specific to fabric monitoring failure:

```
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       Security
                               (Physical/Monitored)
    fab0    ge-0/0/2           Down / Down     Disabled
    fab0
    fab1    ge-9/0/2           Up   / Up       Disabled
    fab1

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    lo0          Up          0
```

Sample Output

show chassis cluster interfaces (SRX5400, SRX5600, and SRX5800 devices with SRX5000 line SRX5K-SCB3 (SCB3) with enhanced midplanes and SRX5K-MPC3-100G10G (IOC3) or SRX5K-MPC3-40G10G (IOC3))

user@host> **show chassis cluster interfaces**

The following output is specific to SRX5400, SRX5600, and SRX5800 devices in a chassis cluster cluster, when the PICs containing fabric links on the SRX5K-MPC3-40G10G (IOC3) are powered off to turn on alternate PICs. 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.

```
Control link status: Up

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

Fabric link status: Down

Fabric interfaces:
    Name    Child-interface    Status        Security
                               (Physical/Monitored)
    fab0     <<< fab child missing once PIC off lined    Disabled
    fab0
    fab1    xe-10/2/7          Up   / Down               Disabled
    fab1
```

```
Redundant-ethernet Information:
    Name         Status      Redundancy-group
    reth0        Up          Not configured
    reth1        Down        1

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    lo0          Up          0
```

show chassis cluster ip-monitoring status redundancy-group

Supported Platforms SRX Series, vSRX

Syntax

show chassis cluster ip-monitoring status
<redundancy-group *group-number*>

Release Information Command introduced in Junos OS Release 9.6. Support for global threshold, current threshold, and weight of each monitored IP address added in Junos OS Release 12.1X47-D10.

Description Display the status of all monitored IP addresses for a redundancy group.

Options

- none— Display the status of monitored IP addresses for all redundancy groups on the node.
- **redundancy-group *group-number*** — Display the status of monitored IP addresses under the specified redundancy group.

Required Privilege Level view

Related Documentation

- clear chassis cluster failover-count

List of Sample Output show chassis cluster ip-monitoring status on page 408
show chassis cluster ip-monitoring status redundancy-group on page 409

Output Fields Table 40 on page 407 lists the output fields for the **show chassis cluster ip-monitoring status** command.

Table 40: show chassis cluster ip-monitoring status Output Fields

Field Name	Field Description
Redundancy-group	ID number (0 - 255) of a redundancy group in the cluster.
Global threshold	Failover value for all IP addresses monitored by the redundancy group.
Current threshold	Value equal to the global threshold minus the total weight of the unreachable IP address.
IP Address	Monitored IP address in the redundancy group.
Status	Current reachability state of the monitored IP address. Values for this field are: **reachable**, **unreachable**, and **unknown**. The status is “unknown” if Packet Forwarding Engines (PFEs) are not yet up and running.
Failure count	Number of attempts to reach an IP address.
Reason	Explanation for the reported status. See Table 41 on page 408.

Table 40: show chassis cluster ip-monitoring status Output Fields *(continued)*

Field Name	Field Description
Weight	Combined weight (0 - 255) assigned to all monitored IP addresses. A higher weight value indicates greater importance.

Expanded reason output fields for unreachable IP addresses added in Junos OS Release 10.1. You might see any of the following reasons displayed.

Table 41: show chassis cluster ip-monitoring status redundancy group Reason Fields

Reason	Reason Description
No route to host	The router could not resolve the ARP, which is needed to send the ICMP packet to the host with the monitored IP address.
No auxiliary IP found	The redundant Ethernet interface does not have an auxiliary IP address configured.
Reth child not up	A child interface of a redundant Ethernet interface is down.
redundancy-group state unknown	Unable to obtain the state (primary, secondary, secondary-hold, disable) of a redundancy-group.
No reth child MAC address	Could not extract the MAC address of the redundant Ethernet child interface.
Secondary link not monitored	The secondary link might be down (the secondary child interface of a redundant Ethernet interface is either down or non-functional).
Unknown	The IP address has just been configured and the router still does not know the status of this IP. or Do not know the exact reason for the failure.

Sample Output

show chassis cluster ip-monitoring status

```
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
10.254.5.44                reachable       0            n/a       220
2.2.2.1                    reachable       0            n/a       100

node1:
--------------------------------------------------------------------------
```

```
Redundancy group: 1
Global threshold: 200
Current threshold: -120

IP address                 Status        Failure count  Reason     Weight
10.254.5.44                reachable       0            n/a        220
2.2.2.1                    reachable       0            n/a        100
```

Sample Output

show chassis cluster ip-monitoring status redundancy-group

```
user@host> show chassis cluster ip-monitoring status redundancy-group 1
node0:
--------------------------------------------------------------------------

Redundancy group: 1

IP address                 Status        Failure count  Reason
10.254.5.44                reachable       0            n/a
2.2.2.1                    reachable       0            n/a
1.1.1.5                    reachable       0            n/a
1.1.1.4                    reachable       0            n/a
1.1.1.1                    reachable       0            n/a

node1:
--------------------------------------------------------------------------

Redundancy group: 1

IP address                 Status        Failure count  Reason
10.254.5.44                reachable       0            n/a
2.2.2.1                    reachable       0            n/a
1.1.1.5                    reachable       0            n/a
1.1.1.4                    reachable       0            n/a
1.1.1.1                    reachable       0            n/a
```

show chassis cluster statistics

Supported Platforms	SRX Series, vSRX
Syntax	**show chassis cluster statistics**
Release Information	Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0.
Description	Display information about chassis cluster services and interfaces.
Required Privilege Level	view
Related Documentation	• clear chassis cluster statistics on page 365
List of Sample Output	show chassis cluster statistics on page 411 show chassis cluster statistics (SRX5000 line devices) on page 412 show chassis cluster statistics (SRX5000 line devices) on page 413
Output Fields	Table 42 on page 410 lists the output fields for the **show chassis cluster statistics** command. Output fields are listed in the approximate order in which they appear.

Table 42: show chassis cluster statistics Output Fields

Field Name	Field Description
Control link statistics	Statistics of the control link used by chassis cluster traffic. Statistics for **Control link 1** are displayed when you use dual control links (SRX5000 lines only). Note that the output for the SRX5000 lines will always show **Control link 0** and **Control link 1** statistics, even though only one control link is active or working. • **Heartbeat packets sent**—Number of heartbeat messages sent on the control link. • **Heartbeat packets received**—Number of heartbeat messages received on the control link. • **Heartbeat packet errors**—Number of heartbeat packets received with errors on the control link.
Fabric link statistics	Statistics of the fabric link used by chassis cluster traffic. Statistics for **Child Link 1** are displayed when you use dual fabric links. • **Probes sent**—Number of probes sent on the fabric link. • **Probes received**—Number of probes received on the fabric link.

Table 42: show chassis cluster statistics Output Fields *(continued)*

Field Name	Field Description
Services Synchronized	• **Service name**—Name of the service. • **Rtos sent**—Number of runtime objects (RTOs) sent. • **Rtos received**—Number of RTOs received. • **Translation context**—Messages synchronizing Network Address Translation (NAT) translation context. • **Incoming NAT**—Messages synchronizing incoming Network Address Translation (NAT) service. • **Resource manager**—Messages synchronizing resource manager groups and resources. • **Session create**—Messages synchronizing session creation. • **Session close**—Messages synchronizing session close. • **Session change**—Messages synchronizing session change. • **Gate create**—Messages synchronizing creation of pinholes (temporary openings in the firewall). • **Session ageout refresh request**—Messages synchronizing request session after age-out. • **Session ageout refresh reply**—Messages synchronizing reply session after age-out. • **IPsec VPN**—Messages synchronizing VPN session. • **Firewall user authentication**—Messages synchronizing firewall user authentication session. • **MGCP ALG**—Messages synchronizing MGCP ALG sessions. • **H323 ALG**—Messages synchronizing H.323 ALG sessions. • **SIP ALG**—Messages synchronizing SIP ALG sessions. • **SCCP ALG**—Messages synchronizing SCCP ALG sessions. • **PPTP ALG**—Messages synchronizing PPTP ALG sessions. • **RTSP ALG**—Messages synchronizing RTSP ALG sessions. • **MAC address learning**—Messages synchronizing MAC address learning.

Sample Output

show chassis cluster statistics

```
user@host> show chassis cluster statistics
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 798
        Heartbeat packets received: 784
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 793
        Probes received: 0
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
        MAC address learning                     0            0
```

Sample Output

show chassis cluster statistics (SRX5000 line devices)

```
user@host> show chassis cluster statistics
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
    Control link 1:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
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
```

Sample Output

show chassis cluster statistics (SRX5000 line devices)

```
user@host> show chassis cluster statistics
Control link statistics:
    Control link 0:
        Heartbeat packets sent: 82371
        Heartbeat packets received: 82321
        Heartbeat packets errors: 0
    Control link 1:
        Heartbeat packets sent: 0
        Heartbeat packets received: 0
        Heartbeat packets errors: 0
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
```

show chassis cluster status

Supported Platforms SRX Series, vSRX

Syntax **show chassis cluster status**
<redundancy-group *group-number* >

Release Information Command modified in Junos OS Release 9.2. Support for dual control ports added in Junos OS Release 10.0. Support for monitoring failures added in Junos OS Release 12.1X47-D10.

Description Display the failover status of a chassis cluster.

Options

- none—Display the status of all redundancy groups in the chassis cluster.
- **redundancy-group *group-number*** —(Optional) Display the status of the specified redundancy group.

Required Privilege Level view

Related Documentation

- redundancy-group (Chassis Cluster) on page 341
- clear chassis cluster failover-count on page 361
- request chassis cluster failover node on page 369
- request chassis cluster failover reset on page 370

List of Sample Output
show chassis cluster status on page 415
show chassis cluster status redundancy-group 1 on page 416

Output Fields Table 43 on page 414 lists the output fields for the **show chassis cluster status** command. Output fields are listed in the approximate order in which they appear.

Table 43: show chassis cluster status Output Fields

Field Name	Field Description
Cluster ID	ID number (1-15) of a cluster is applicable for releases upto 12.1X45-D10. ID number (1-255) is applicable for releases 12.1X45-D10 and later. Setting a cluster ID to 0 is equivalent to disabling a cluster.
Redundancy-Group	ID number (1-128) of a redundancy group in the chassis cluster.
Node name	Node (device) in the chassis cluster (**node0** or **node1**).
Priority	Assigned priority for the redundancy group on that node.

Table 43: show chassis cluster status Output Fields *(continued)*

Field Name	Field Description
Status	State of the redundancy group (**Primary**, **Secondary**, **Lost**, or **Unavailable**). • **Primary**—Redundancy group is active and passing traffic. • **Secondary**—Redundancy group is passive and not passing traffic. • **Lost**—Node loses contact with the other node through the control link. Most likely to occur when both nodes are in a cluster and due to control link failure, one node cannot exchange heartbeats, or when the other node is rebooted. • **Unavailable**—Node has not received a single heartbeat over the control link from the other node since the other node booted up. Most likely to occur when one node boots up before the other node, or if only one node is present in the cluster.
Preempt	• **Yes**: Mastership can be preempted based on priority. • **No**: Change in priority will not preempt the mastership.
Manual failover	• **Yes**: If the Mastership is set manually through the CLI with the **request chassis cluster failover node** or **request chassis cluster failover redundancy-group** command. This overrides **Priority** and **Preempt**. • **No**: Mastership is not set manually through the CLI.
Monitor-failures	• **None**: Cluster working properly. • **Monitor Failure code**: Cluster is not working properly and the respective failure code is displayed.

Sample Output

Displays chassis cluster status with all redundancy groups.

show chassis cluster status

```
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  200      primary        no      no       None
node1  1        secondary      no      no       None

Redundancy group: 1 , Failover count: 1
node0  101      primary        no      no       None
node1  1        secondary      no      no       None
```

Sample Output

Displays chassis cluster status with redundancy group 1 only.

show chassis cluster status redundancy-group 1

```
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  101      primary        no      no       None
node1  1        secondary      no      no       None
```

show chassis environment (Security)

Supported Platforms SRX Series, vSRX

Syntax show chassis environment

Release Information Command introduced in Junos OS Release 9.2.

Description Display environmental information about the services gateway chassis, including the temperature and information about the fans, power supplies, and Routing Engine.

Options **none**—Display environmental information about the device.

cb ***slot-number***—Display chassis environmental information for the Control Board.

fpc ***fpc-slot***—Display chassis environmental information for a specified Flexible PIC Concentrator.

fpm—Display chassis environmental information for the craft interface (FPM).

pem ***slot-number***—Display chassis environmental information for the specified Power Entry Module.

routing-engine ***slot-number***—Display chassis environmental information for the specified Routing Engine.

Required Privilege Level view

Related Documentation

- show chassis hardware (View) on page 439

List of Sample Output show chassis environment on page 417

Output Fields Table 44 on page 417 lists the output fields for the **show chassis environment** command. Output fields are listed in the approximate order in which they appear.

Table 44: show chassis environment Output Fields

Field Name	Field Description
Temp	Temperature of air flowing through the chassis in degrees Celsius (C) and Fahrenheit (F).
Fan	Fan status: **OK**, **Testing** (during initial power-on), **Failed**, or **Absent**.

Sample Output

show chassis environment

```
user@host> show chassis environment
user@host> show chassis environment
Class Item                          Status     Measurement
Temp  PEM 0                         OK         40 degrees C / 104 degrees F
```



```
PEM 1                          OK         40 degrees C / 104 degrees F
PEM 2                          OK         40 degrees C / 104 degrees F
PEM 3                          OK         45 degrees C / 113 degrees F
Routing Engine 0               OK         31 degrees C / 87 degrees F
Routing Engine 0 CPU           OK         27 degrees C / 80 degrees F
Routing Engine 1               Absent
Routing Engine 1 CPU           Absent
CB 0 Intake                    OK         28 degrees C / 82 degrees F
CB 0 Exhaust A                 OK         27 degrees C / 80 degrees F
CB 0 Exhaust B                 OK         29 degrees C / 84 degrees F
CB 0 ACBC                      OK         29 degrees C / 84 degrees F
CB 0 SF A                      OK         36 degrees C / 96 degrees F
CB 0 SF B                      OK         31 degrees C / 87 degrees F
CB 1 Intake                    OK         27 degrees C / 80 degrees F
CB 1 Exhaust A                 OK         26 degrees C / 78 degrees F
CB 1 Exhaust B                 OK         29 degrees C / 84 degrees F
CB 1 ACBC                      OK         27 degrees C / 80 degrees F
CB 1 SF A                      OK         36 degrees C / 96 degrees F
CB 1 SF B                      OK         31 degrees C / 87 degrees F
CB 2 Intake                    Absent
CB 2 Exhaust A                 Absent
CB 2 Exhaust B                 Absent
CB 2 ACBC                      Absent
CB 2 XF A                      Absent
CB 2 XF B                      Absent
FPC 0 Intake                   OK         47 degrees C / 116 degrees F
FPC 0 Exhaust A                OK         44 degrees C / 111 degrees F
FPC 0 Exhaust B                OK         52 degrees C / 125 degrees F
FPC 0 xlp0 TSen                OK         51 degrees C / 123 degrees F
FPC 0 xlp0 Chip                OK         46 degrees C / 114 degrees F
FPC 0 xlp1 TSen                OK         51 degrees C / 123 degrees F
FPC 0 xlp1 Chip                OK         47 degrees C / 116 degrees F
FPC 0 xlp2 TSen                OK         44 degrees C / 111 degrees F
FPC 0 xlp2 Chip                OK         42 degrees C / 107 degrees F
FPC 0 xlp3 TSen                OK         48 degrees C / 118 degrees F
FPC 0 xlp3 Chip                OK         43 degrees C / 109 degrees F
FPC 1 Intake                   OK         41 degrees C / 105 degrees F
FPC 1 Exhaust A                OK         41 degrees C / 105 degrees F
FPC 1 Exhaust B                OK         51 degrees C / 123 degrees F
FPC 1 LU TSen                  OK         46 degrees C / 114 degrees F
FPC 1 LU Chip                  OK         45 degrees C / 113 degrees F
FPC 1 XM TSen                  OK         46 degrees C / 114 degrees F
FPC 1 XM Chip                  OK         52 degrees C / 125 degrees F
FPC 1 xlp0 TSen                OK         49 degrees C / 120 degrees F
FPC 1 xlp0 Chip                OK         42 degrees C / 107 degrees F
FPC 1 xlp1 TSen                OK         49 degrees C / 120 degrees F
FPC 1 xlp1 Chip                OK         44 degrees C / 111 degrees F
FPC 1 xlp2 TSen                OK         38 degrees C / 100 degrees F
FPC 1 xlp2 Chip                OK         39 degrees C / 102 degrees F
FPC 1 xlp3 TSen                OK         44 degrees C / 111 degrees F
FPC 1 xlp3 Chip                OK         42 degrees C / 107 degrees F
FPC 2 Intake                   OK         29 degrees C / 84 degrees F
FPC 2 Exhaust A                OK         34 degrees C / 93 degrees F
FPC 2 Exhaust B                OK         40 degrees C / 104 degrees F
FPC 2 I3 0 TSensor             OK         42 degrees C / 107 degrees F
FPC 2 I3 0 Chip                OK         41 degrees C / 105 degrees F
FPC 2 I3 1 TSensor             OK         40 degrees C / 104 degrees F
FPC 2 I3 1 Chip                OK         39 degrees C / 102 degrees F
FPC 2 I3 2 TSensor             OK         38 degrees C / 100 degrees F
FPC 2 I3 2 Chip                OK         37 degrees C / 98 degrees F
FPC 2 I3 3 TSensor             OK         35 degrees C / 95 degrees F
```



```
                FPC 2 I3 3 Chip                OK         35 degrees C / 95 degrees F
                FPC 2 IA 0 TSensor             OK         45 degrees C / 113 degrees F
                FPC 2 IA 0 Chip                OK         42 degrees C / 107 degrees F
                FPC 2 IA 1 TSensor             OK         41 degrees C / 105 degrees F
                FPC 2 IA 1 Chip                OK         43 degrees C / 109 degrees F
                FPC 9 Intake                   OK         29 degrees C / 84 degrees F
                FPC 9 Exhaust A                OK         41 degrees C / 105 degrees F
                FPC 9 Exhaust B                OK         48 degrees C / 118 degrees F
                FPC 9 LU TSen                  OK         48 degrees C / 118 degrees F
                FPC 9 LU Chip                  OK         47 degrees C / 116 degrees F
                FPC 9 XM TSen                  OK         48 degrees C / 118 degrees F
                FPC 9 XM Chip                  OK         54 degrees C / 129 degrees F
                FPC 9 xlp0 TSen                OK         45 degrees C / 113 degrees F
                FPC 9 xlp0 Chip                OK         42 degrees C / 107 degrees F
                FPC 9 xlp1 TSen                OK         49 degrees C / 120 degrees F
                FPC 9 xlp1 Chip                OK         46 degrees C / 114 degrees F
                FPC 9 xlp2 TSen                OK         37 degrees C / 98 degrees F
                FPC 9 xlp2 Chip                OK         40 degrees C / 104 degrees F
                FPC 9 xlp3 TSen                OK         45 degrees C / 113 degrees F
                FPC 9 xlp3 Chip                OK         41 degrees C / 105 degrees F
                FPC 10 Intake                  OK         32 degrees C / 89 degrees F
                FPC 10 Exhaust A               OK         44 degrees C / 111 degrees F
                FPC 10 Exhaust B               OK         53 degrees C / 127 degrees F
                FPC 10 LU 0 TSen               OK         43 degrees C / 109 degrees F
                FPC 10 LU 0 Chip               OK         52 degrees C / 125 degrees F
                FPC 10 LU 1 TSen               OK         43 degrees C / 109 degrees F
                FPC 10 LU 1 Chip               OK         44 degrees C / 111 degrees F
                FPC 10 LU 2 TSen               OK         43 degrees C / 109 degrees F
                FPC 10 LU 2 Chip               OK         50 degrees C / 122 degrees F
                FPC 10 LU 3 TSen               OK         43 degrees C / 109 degrees F
                FPC 10 LU 3 Chip               OK         58 degrees C / 136 degrees F
                FPC 10 XM 0 TSen               OK         43 degrees C / 109 degrees F
                FPC 10 XM 0 Chip               OK         53 degrees C / 127 degrees F
                FPC 10 XF 0 TSen               OK         43 degrees C / 109 degrees F
                FPC 10 XF 0 Chip               OK         64 degrees C / 147 degrees F
                FPC 10 PLX Switch TSen         OK         43 degrees C / 109 degrees F
                FPC 10 PLX Switch Chip         OK         44 degrees C / 111 degrees F
                FPC 11 Intake                  OK         32 degrees C / 89 degrees F
                FPC 11 Exhaust A               OK         41 degrees C / 105 degrees F
                FPC 11 Exhaust B               OK         56 degrees C / 132 degrees F
                FPC 11 LU 0 TSen               OK         45 degrees C / 113 degrees F
                FPC 11 LU 0 Chip               OK         50 degrees C / 122 degrees F
                FPC 11 LU 1 TSen               OK         45 degrees C / 113 degrees F
                FPC 11 LU 1 Chip               OK         47 degrees C / 116 degrees F
                FPC 11 LU 2 TSen               OK         45 degrees C / 113 degrees F
                FPC 11 LU 2 Chip               OK         52 degrees C / 125 degrees F
                FPC 11 LU 3 TSen               OK         45 degrees C / 113 degrees F
                FPC 11 LU 3 Chip               OK         60 degrees C / 140 degrees F
                FPC 11 XM 0 TSen               OK         45 degrees C / 113 degrees F
                FPC 11 XM 0 Chip               OK         56 degrees C / 132 degrees F
                FPC 11 XF 0 TSen               OK         45 degrees C / 113 degrees F
                FPC 11 XF 0 Chip               OK         65 degrees C / 149 degrees F
                FPC 11 PLX Switch TSen         OK         45 degrees C / 113 degrees F
                FPC 11 PLX Switch Chip         OK         46 degrees C / 114 degrees F
          Fans  Top Fan Tray Temp              OK         34 degrees C / 93 degrees F
                Top Tray Fan 1                 OK         Spinning at normal speed
                Top Tray Fan 2                 OK         Spinning at normal speed
                Top Tray Fan 3                 OK         Spinning at normal speed
                Top Tray Fan 4                 OK         Spinning at normal speed
                Top Tray Fan 5                 OK         Spinning at normal speed
                Top Tray Fan 6                 OK         Spinning at normal speed
```

```
Top Tray Fan 7                  OK          Spinning at normal speed
Top Tray Fan 8                  OK          Spinning at normal speed
Top Tray Fan 9                  OK          Spinning at normal speed
Top Tray Fan 10                 OK          Spinning at normal speed
Top Tray Fan 11                 OK          Spinning at normal speed
Top Tray Fan 12                 OK          Spinning at normal speed
Bottom Fan Tray Temp            OK          31 degrees C / 87 degrees F
Bottom Tray Fan 1               OK          Spinning at normal speed
Bottom Tray Fan 2               OK          Spinning at normal speed
Bottom Tray Fan 3               OK          Spinning at normal speed
Bottom Tray Fan 4               OK          Spinning at normal speed
Bottom Tray Fan 5               OK          Spinning at normal speed
Bottom Tray Fan 6               OK          Spinning at normal speed
Bottom Tray Fan 7               OK          Spinning at normal speed
Bottom Tray Fan 8               OK          Spinning at normal speed
Bottom Tray Fan 9               OK          Spinning at normal speed
Bottom Tray Fan 10              OK          Spinning at normal speed
Bottom Tray Fan 11              OK          Spinning at normal speed
Bottom Tray Fan 12              OK          Spinning at normal speed
  OK
```

show chassis environment cb

Supported Platforms	SRX Series, vSRX
Syntax	show chassis environment cb <*slot*>
Release Information	Command introduced in Junos OS Release 9.2. Starting with Junos OS Release 12.1X47-D15, the SRX5K-SCBE (SCB2) is introduced and starting with Junos OS Release 15.1X49-D10, the SRX5K-SCB3 (SCB3) with enhanced midplanes is introduced.
Description	SRX Series devices display environmental information about the Control Boards (CBs).
Options	*slot*—(Optional) Display environmental information about the specified CB.
Required Privilege Level	view
Related Documentation	• request chassis cb on page 366
List of Sample Output	show chassis environment cb node 0 (SRX5600 devices with SRX5K-SCB3 (SCB3) and Enhanced Midplanes) on page 422 show chassis environment cb node 1(SRX5600 devices with SRX5K-SCB3 (SCB3) and Enhanced Midplanes) on page 422
Output Fields	Table 45 on page 421 lists the output fields for the **show chassis environment cb** command. Output fields are listed in the approximate order in which they appear.

Table 45: show chassis environment cb Output Fields

Field Name	Field Description
State	Status of the CB. If two CBs are installed and online, one is functioning as the master, and the other is the standby. • **Online**—CB is online and running. • **Offline**— CB is powered down.
Temperature	Temperature in Celsius (C) and Fahrenheit (F) of the air flowing past the CB. • **Temperature Intake**—Measures the temperature of the air intake to cool the power supplies. • **Temperature Exhaust**—Measures the temperature of the hot air exhaust.
Power	Power required and measured on the CB. The left column displays the required power, in volts. The right column displays the measured power, in millivolts.
BUS Revision	Revision level of the generic bus device.
FPGA Revision	Revision level of the field-programmable gate array (FPGA).

Table 45: show chassis environment cb Output Fields *(continued)*

Field Name	Field Description
PMBus device	Enhanced SCB on SRX Series devices allows the system to save power by supplying only the amount of voltage that is required. Configurable PMBus devices are used to provide the voltage for each individual device. There is one PMBus device for each XF ASIC so that the output can be customized to each device. The following PMBus device information is displayed for devices with Enhanced MX SCB: • **Expected voltage** • **Measured voltage** • **Measured current** • **Calculated power**

Sample Output

show chassis environment cb node 0 (SRX5600 devices with SRX5K-SCB3 (SCB3) and Enhanced Midplanes)

```
user@host> show chassis environment cb node 0
node0:
--------------------------------------------------------------------------
--------------------------------------------------------------------------
CB 0 status:
  State                      Online Master
  Temperature                34 degrees C / 93 degrees F
  Power 1
    1.0 V                       1002
    1.2 V                       1198
    1.5 V                       1501
    1.8 V                       1801
    2.5 V                       2507
    3.3 V                       3300
    5.0 V                       5014
    5.0 V RE                    4982
    12.0 V                     11988
    12.0 V RE                  11930
  Power 2
    4.6 V bias MidPlane         4801
    11.3 V bias PEM            11292
    11.3 V bias FPD            11272
    11.3 V bias POE 0          11214
    11.3 V bias POE 1          11253
  Bus Revision               96
  FPGA Revision              16
  PMBus             Expected   Measured   Measured   Calculated
  device            voltage    voltage    current    power
    XF ASIC A        1033 mV    1033 mV   15500 mA    16011 mW
    XF ASIC B        1034 mV    1033 mV   15000 mA    15495 mW
```

show chassis environment cb node 1(SRX5600 devices with SRX5K-SCB3 (SCB3) and Enhanced Midplanes)

```
user@host> show chassis environment cb node 1
node1:
--------------------------------------------------------------------------
CB 0 status:
  State                      Online Master
  Temperature                35 degrees C / 95 degrees F
  Power 1
```



```
  1.0 V                      1002
  1.2 V                      1198
  1.5 V                      1504
  1.8 V                      1801
  2.5 V                      2507
  3.3 V                      3325
  5.0 V                      5014
  5.0 V RE                   4943
  12.0 V                    12007
  12.0 V RE                 12007
Power 2
  4.6 V bias MidPlane        4814
  11.3 V bias PEM           11272
  11.3 V bias FPD           11330
  11.3 V bias POE 0         11176
  11.3 V bias POE 1         11292
Bus Revision              96
FPGA Revision             16
PMBus             Expected   Measured   Measured  Calculated
device            voltage    voltage    current   power
  XF ASIC A         958 mV     959 mV   13500 mA   12946 mW
  XF ASIC B        1033 mV    1031 mV   16500 mA   17011 mW
```


show chassis ethernet-switch

Supported Platforms SRX Series, vSRX

Syntax show chassis ethernet-switch

Release Information Command introduced in Junos OS Release 9.2.

Description SRX Series devices display information about the ports on the Control Board (CB) Ethernet switch.

Required Privilege Level view

Related Documentation
- cluster (Chassis) on page 318

List of Sample Output show chassis ethernet-switch on page 424

Output Fields Table 46 on page 424 lists the output fields for the **show chassis ethernet-switch** command. Output fields are listed in the approximate order in which they appear.

Table 46: show chassis ethernet-switch Output Fields

Field Name	Field Description
Link is good on port n connected to device or **Link is good on Fast Ethernet port n connected to device**	Information about the link between each port on the CB's Ethernet switch and one of the following devices: • FPC0 (Flexible PIC Concentrator 0) through FPC7 • Local controller • Routing Engine • Other Routing Engine (on a system with two Routing Engines) • SPMB (Switch Processor Mezzanine Board)
Speed is	Speed at which the Ethernet link is running.
Duplex is	Duplex type of the Ethernet link: **full** or **half**.
Autonegotiate is Enabled (or Disabled)	By default, built-in Fast Ethernet ports on a PIC autonegotiate whether to operate at 10 Mbps or 100 Mbps. All other interfaces automatically choose the correct speed based on the PIC type and whether the PIC is configured to operate in multiplexed mode.

Sample Output

show chassis ethernet-switch

```
user@host> show chassis ethernet-switch
node0:
--------------------------------------------------------------------------
Displaying summary for switch 0
Link is good on GE port 0 connected to device: FPC0
  Speed is 1000Mb
  Duplex is full
```



```
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 1 connected to device: FPC1
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 2 connected to device: FPC2
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 3 connected to device: FPC3
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 4 connected to device: FPC4
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is down on GE port 5 connected to device: FPC5

Link is down on GE port 6 connected to device: FPC6

Link is good on GE port 7 connected to device: FPC7
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 8 connected to device: FPC8
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 9 connected to device: FPC9
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is down on GE port 10 connected to device: FPC10

Link is down on GE port 11 connected to device: FPC11
```

```
Link is good on GE port 12 connected to device: Other RE
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 13 connected to device: RE-GigE
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is down on GE port 14 connected to device: Debug-GigE

node1:
--------------------------------------------------------------------------
Displaying summary for switch 0
Link is good on GE port 0 connected to device: FPC0
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 1 connected to device: FPC1
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 2 connected to device: FPC2
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 3 connected to device: FPC3
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 4 connected to device: FPC4
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is down on GE port 5 connected to device: FPC5

Link is down on GE port 6 connected to device: FPC6

Link is good on GE port 7 connected to device: FPC7
  Speed is 1000Mb
  Duplex is full
```



```
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 8 connected to device: FPC8
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 9 connected to device: FPC9
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is down on GE port 10 connected to device: FPC10

Link is down on GE port 11 connected to device: FPC11

Link is good on GE port 12 connected to device: Other RE
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is good on GE port 13 connected to device: RE-GigE
  Speed is 1000Mb
  Duplex is full
  Autonegotiate is Enabled
  Flow Control TX is Disabled
  Flow Control RX is Disabled

Link is down on GE port 14 connected to device: Debug-GigE
```

show chassis fabric plane

Supported Platforms SRX Series, vSRX

Syntax show chassis fabric plane

Release Information Command introduced in Junos OS Release 9.2.

Description Show state of fabric management plane.

Required Privilege Level view

Related Documentation
- show chassis fabric plane-location on page 434

List of Sample Output show chassis fabric plane(SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4) on page 429

Output Fields Table 47 on page 428 lists the output fields for the **show chassis fabric plane** command. Output fields are listed in the approximate order in which they appear.

Table 47: show chassis fabric plane Output Fields

Field Name	Field Description	Level of output
Plane	Number of the plane.	none

Plane state	State of each plane:
	• **ACTIVE**—SIB is operational and running.
	• **FAULTY**— SIB is in alarmed state where the SIB's plane is not operational for the following reasons:
	  ◦ On-board fabric ASIC is not operational.
	  ◦ Fiber-optic connector faults.
	  ◦ FPC connector faults.
	  ◦ SIB midplane connector faults.
		none

FPC	Slot number of each Flexible PIC Concentrator (FPC).	none
PFE	Slot number of each Packet Forwarding Engine and the state of the links to the FPC: • **Links ok**: Link between SIB and FPC is active. • **Link error**: Link between SIB and FPC is not operational. • **Unused**: No FPC is present.	none

Table 47: show chassis fabric plane Output Fields *(continued)*

Field Name	Field Description	Level of output

State	State of the fabric plane:
	• **Online**: Fabric plane is operational and running and links on the SIB are operational.
	• **Offline**: Fabric plane state is **Offline** because the plane does not have four or more F2S and one F13 online.
	• **Empty**: Fabric plane state is **Empty** if all SIBs in the plane are absent.
	• **Spare**: Fabric plane is redundant and can be operational if the operational fabric plane encounters an error.
	• **Check**: Fabric plane is in alarmed state due to the following reason and the cause of the error must be resolved:
	  ◦ One or more SIBs (belonging to the fabric plane) in the **Online** or **Spare** states has transitioned to the **Check** state. **Check** state of the SIB can be caused by link errors or destination errors.
	• **Fault**: Fabric plane is in alarmed state if one or more SIBs belonging to the plane are in the **Fault** state. A SIB can be in the **Fault** state because of the following reasons:
	  ◦ On-board fabric ASIC is not operational.
	  ◦ Fiber-optic connector faults.
	  ◦ FPC connector faults.
	  ◦ SIB midplane connector faults.
	  ◦ Link errors have exceeded the threshold.
		none

Sample Output

show chassis fabric plane (SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4)

```
user@host> show chassis fabric plane
node0:
--------------------------------------------------------------------------
Fabric management PLANE state
Plane 0
  Plane state: ACTIVE
      FPC 0
          PFE 0 :Links ok
      FPC 2
          PFE 0 :Links ok
      FPC 3
          PFE 0 :Links ok
      FPC 4
          PFE 0 :Links ok
      FPC 7
          PFE 0 :Links ok
      FPC 8
          PFE 0 :Links ok
      FPC 9
          PFE 0 :Links ok
      FPC 10
          PFE 0 :Links ok
```

```
Plane 1
  Plane state: ACTIVE
      FPC 0
          PFE 0 :Links ok
      FPC 2
          PFE 0 :Links ok
      FPC 3
          PFE 0 :Links ok
      FPC 4
          PFE 0 :Links ok
      FPC 7
          PFE 0 :Links ok
      FPC 8
          PFE 0 :Links ok
      FPC 9
          PFE 0 :Links ok
      FPC 10
          PFE 0 :Links ok
Plane 2
  Plane state: ACTIVE
      FPC 0
          PFE 0 :Links ok
      FPC 2
          PFE 0 :Links ok
      FPC 3
          PFE 0 :Links ok
      FPC 4
          PFE 0 :Links ok
      FPC 7
          PFE 0 :Links ok
      FPC 8
          PFE 0 :Links ok
      FPC 9
          PFE 0 :Links ok
      FPC 10
          PFE 0 :Links ok
Plane 3
  Plane state: ACTIVE
      FPC 0
          PFE 0 :Links ok
      FPC 2
          PFE 0 :Links ok
      FPC 3
          PFE 0 :Links ok
      FPC 4
          PFE 0 :Links ok
      FPC 7
          PFE 0 :Links ok
      FPC 8
          PFE 0 :Links ok
      FPC 9
          PFE 0 :Links ok
      FPC 10
          PFE 0 :Links ok
Plane 4
  Plane state: SPARE
      FPC 0
          PFE 0 :Links ok
      FPC 2
          PFE 0 :Links ok
      FPC 3
```



```
            PFE 0 :Links ok
        FPC 4
            PFE 0 :Links ok
        FPC 7
            PFE 0 :Links ok
        FPC 8
            PFE 0 :Links ok
        FPC 9
            PFE 0 :Links ok
        FPC 10
            PFE 0 :Links ok
Plane 5
  Plane state: SPARE
        FPC 0
            PFE 0 :Links ok
        FPC 2
            PFE 0 :Links ok
        FPC 3
            PFE 0 :Links ok
        FPC 4
            PFE 0 :Links ok
        FPC 7
            PFE 0 :Links ok
        FPC 8
            PFE 0 :Links ok
        FPC 9
            PFE 0 :Links ok
        FPC 10
            PFE 0 :Links ok

node1:
--------------------------------------------------------------------------
Fabric management PLANE state
Plane 0
  Plane state: ACTIVE
        FPC 0
            PFE 0 :Links ok
        FPC 1
            PFE 0 :Links ok
        FPC 2
            PFE 0 :Links ok
        FPC 3
            PFE 0 :Links ok
        FPC 4
            PFE 0 :Links ok
        FPC 7
            PFE 0 :Links ok
        FPC 8
            PFE 0 :Links ok
        FPC 10
            PFE 0 :Links ok
Plane 1
  Plane state: ACTIVE
        FPC 0
            PFE 0 :Links ok
        FPC 1
            PFE 0 :Links ok
        FPC 2
            PFE 0 :Links ok
        FPC 3
            PFE 0 :Links ok
```

```
        FPC 4
            PFE 0 :Links ok
        FPC 7
            PFE 0 :Links ok
        FPC 8
            PFE 0 :Links ok
        FPC 10
            PFE 0 :Links ok
Plane 2
  Plane state: ACTIVE
        FPC 0
            PFE 0 :Links ok
        FPC 1
            PFE 0 :Links ok
        FPC 2
            PFE 0 :Links ok
        FPC 3
            PFE 0 :Links ok
        FPC 4
            PFE 0 :Links ok
        FPC 7
            PFE 0 :Links ok
        FPC 8
            PFE 0 :Links ok
        FPC 10
            PFE 0 :Links ok
Plane 3
  Plane state: ACTIVE
        FPC 0
            PFE 0 :Links ok
        FPC 1
            PFE 0 :Links ok
        FPC 2
            PFE 0 :Links ok
        FPC 3
            PFE 0 :Links ok
        FPC 4
            PFE 0 :Links ok
        FPC 7
            PFE 0 :Links ok
        FPC 8
            PFE 0 :Links ok
        FPC 10
            PFE 0 :Links ok
Plane 4
  Plane state: SPARE
        FPC 0
            PFE 0 :Links ok
        FPC 1
            PFE 0 :Links ok
        FPC 2
            PFE 0 :Links ok
        FPC 3
            PFE 0 :Links ok
        FPC 4
            PFE 0 :Links ok
        FPC 7
            PFE 0 :Links ok
        FPC 8
            PFE 0 :Links ok
        FPC 10
```



```
          PFE 0 :Links ok
Plane 5
  Plane state: SPARE
      FPC 0
          PFE 0 :Links ok
      FPC 1
          PFE 0 :Links ok
      FPC 2
          PFE 0 :Links ok
      FPC 3
          PFE 0 :Links ok
      FPC 4
          PFE 0 :Links ok
      FPC 7
          PFE 0 :Links ok
      FPC 8
          PFE 0 :Links ok
      FPC 10
          PFE 0 :Links ok
```

show chassis fabric plane-location

Supported Platforms SRX Series, vSRX

Syntax show chassis fabric plane-location

Release Information Command introduced in Junos OS Release 9.2.

Description Show fabric plane location.

Required Privilege Level view

Related Documentation

- show chassis fabric plane on page 428

List of Sample Output show chassis fabric plane-location(SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4) on page 434

Output Fields Table 48 on page 434 lists the output fields for the **show chassis fabric plane-location** command. Output fields are listed in the approximate order in which they appear.

Table 48: show chassis fabric plane-location Output Fields

Field Name	Field Description
Plane *n*	Plane number.
Control Board *n*	Control Board number.

Sample Output

show chassis fabric plane-location (SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4)

```
user@host> show chassis fabric plane-location
node0:
--------------------------------------------------------------------------
------------Fabric Plane Locations-------------
Plane 0                       Control Board 0
Plane 1                       Control Board 0
Plane 2                       Control Board 1
Plane 3                       Control Board 1
Plane 4                       Control Board 2
Plane 5                       Control Board 2

node1:
--------------------------------------------------------------------------
------------Fabric Plane Locations-------------
Plane 0                       Control Board 0
Plane 1                       Control Board 0
Plane 2                       Control Board 1
Plane 3                       Control Board 1
Plane 4                       Control Board 2
Plane 5                       Control Board 2
```


show chassis fabric summary

Supported Platforms SRX Series, vSRX

Syntax show chassis fabric summary

Release Information Command introduced in Junos OS Release 9.2.

Description Show summary fabric management state.

Options This command has no options.

Required Privilege Level view

Related Documentation

- show chassis fabric plane on page 428
- show chassis fabric plane-location on page 434

List of Sample Output show chassis fabric summary(SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4) on page 437

Output Fields Table 49 on page 436 lists the output fields for the **show chassis fabric summary** command. Output fields are listed in the approximate order in which they appear.

Table 49: show chassis fabric summary Output Fields

Field Name	Field Description
Plane	Plane number.

Table 49: show chassis fabric summary Output Fields *(continued)*

Field Name	Field Description

State	State of the SIB or FPC:
	• **Online**—Switch Interface Board (SIB) is operational and running.
	• **Empty**—SIB is powered down.
	• **Check**—SIB is in the **Check** state because of the following reasons:
	  ◦ SIB is not inserted properly.
	  ◦ Some destination errors are detected on the SIB. In this case, the Packet Forwarding Engine stops using the SIB to send traffic to the affected destination Packet Forwarding Engine.
	  ◦ Some link errors are detected on the channel between the SIB and a Packet Forwarding Engine. Link errors can be detected at initialization time or runtime:
	    ▪ Link errors caused by a link training failure at initialization time—The Packet Forwarding Engine does not use the SIB to send traffic. The **show chassis fabric fpcs** command shows **Plane disabled** as status for this link.
	    ▪ Link errors caused by CRC errors detected at runtime—The Packet Forwarding Engine continues to use the SIB to send traffic. The **show chassis fabric fpcs** command shows **Link error** as the status for this link.
	For information about link and destination errors, issue the **show chassis fabric fpcs** commands.
	• **Spare**—SIB is redundant and will move to active state if one of the working SIBs fails.

Errors	Indicates whether there is any error on the SIB. • **None**—No errors • **Link Errors**—Fabric link errors were found on the SIB RX link. • **Cell drops**—Fabric cell drops were found on the SIB ASIC. • **Link, Cell drops**—Both link errors and cell drops were detected on at least one of the FPC's fabric links. NOTE: The **Errors** column is empty only when the FPC or SIB is offline.
Uptime	Elapsed time the plane has been online.

Sample Output

show chassis fabric summary (SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4)

```
user@host> show chassis fabric summary
node0:
--------------------------------------------------------------------------
Plane   State    Uptime
 0      Online   14 minutes, 10 seconds
 1      Online   14 minutes, 5 seconds
 2      Online   14 minutes
 3      Online   13 minutes, 55 seconds
```



```
 4      Spare    13 minutes, 50 seconds
 5      Spare    13 minutes, 44 seconds

node1:
--------------------------------------------------------------------------
Plane   State    Uptime
 0      Online   14 minutes, 7 seconds
 1      Online   14 minutes, 2 seconds
 2      Online   13 minutes, 57 seconds
 3      Online   13 minutes, 51 seconds
 4      Spare    13 minutes, 46 seconds
 5      Spare    13 minutes, 41 seconds
```


show chassis hardware (View)

Supported Platforms SRX Series

Syntax

```
show chassis hardware
<clei-models | detail | extensive | models | node ( node-id | all | local | primary)>
```

Release Information Command introduced in Junos OS Release 9.2. Command modified in Junos OS Release 9.2 to include **node** option.

Description Display chassis hardware information.

Options

- **clei-models**—(Optional) Display Common Language Equipment Identifier Code (CLEI) barcode and model number for orderable field-replaceable units (FRUs).
- **detail | extensive**—(Optional) Display the specified level of output.
- **models**—(Optional) Display model numbers and part numbers for orderable FRUs.
- **node**—(Optional) For chassis cluster configurations, display chassis hardware information on a specific node (device) in the cluster.
  - ***node-id*** —Identification number of the node. It can be 0 or 1.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.

Required Privilege Level view

Related Documentation

- *Juniper Networks Devices Processing Overview*
- *Interface Naming Conventions*

Output Fields Table 50 on page 439 lists the output fields for the **show chassis hardware** command. Output fields are listed in the approximate order in which they appear.

Table 50: show chassis hardware Output Fields

Field Name	Field Description
Item	Chassis component—Information about the backplane; power supplies; fan trays; Routing Engine; each Physical Interface Module (PIM)—reported as FPC and PIC—and each fan, blower, and impeller.
Version	Revision level of the chassis component.
Part Number	Part number for the chassis component.
Serial Number	Serial number of the chassis component. The serial number of the backplane is also the serial number of the device chassis. Use this serial number when you need to contact Juniper Networks Customer Support about the device chassis.

Table 50: show chassis hardware Output Fields *(continued)*

Field Name	Field Description
Assb ID or Assembly ID	Identification number that describes the FRU hardware.
FRU model number	Model number of FRU hardware component.
CLEI code	Common Language Equipment Identifier code. This value is displayed only for hardware components that use ID EEPROM format v2. This value is not displayed for components that use ID EEPROM format v1.
EEPROM Version	ID EEPROM version used by hardware component: 0x01 (version 1) or 0x02 (version 2).

Table 50: show chassis hardware Output Fields *(continued)*

Field Name	Field Description
Description	Brief description of the hardware item: • Type of power supply. • Switch Control Board (SCB) Starting with Junos OS Release 12.1X47-D15, the SRX5K-SCBE (SCB2) is introduced. ◦ There are three SCB slots in SRX5800 devices. The third slot can be used for an SCB or an FPC. When an SRX5K-SCB was used , the third SCB slot was used as an FPC. SCB redundancy is provided in chassis cluster mode. ◦ With an SCB2, a third SCB is supported. If a third SCB is plugged in, it provides intra-chassis fabric redundancy. ◦ The Ethernet switch in the SCB2 provides the Ethernet connectivity among all the FPCs and the Routing Engine. The Routing Engine uses this connectivity to distribute forwarding and routing tables to the FPCs. The FPCs use this connectivity to send exception packets to the Routing Engine. ◦ Fabric connects all FPCs in the data plane. The Fabric Manager executes on the Routing Engine and controls the fabric system in the chassis. Packet Forwarding Engines on the FPC and fabric planes on the SCB are connected through HSL2 channels. ◦ SCB2 supports HSL2 with both 3.11 Gbps and 6.22 Gbps (SerDes) link speed and various HSL2 modes. When an FPC is brought online, the link speed and HSL2 mode are determined by the type of FPC. Starting with Junos OS Release 15.1X49-D10, the SRX5K-SCB3 (SCB3) with enhanced midplanes is introduced. ◦ All existing SCB software that is supported by SCB2 is supported on SCB3. ◦ SRX5K-RE-1800X4 (RE2). Mixed Routing Engine use is not supported. ◦ SCB3 works with the SRX5K-MPC (IOC2), SRX5K-MPC3-100G10G (IOC3), SRX5K-MPC3-40G10G (IOC3), and SRX5K-SPC-4-15-320 (SPC2) with current midplanes and the new enhanced midplanes. ◦ Mixed SCB use is not supported. If an SCB2 and an SCB3 are used, the system will only power on the master Routing Engine's SCB and will power off the other SCBs. Only the SCB in slot 0 is powered on and a system log is generated. ◦ SCB3 supports up to 400 Gbps per slot with old midplanes and up to 500 Gbps per slot with new midplanes. ◦ SCB3 supports fabric intra-chassis redundancy. ◦ SCB3 supports the same chassis cluster function as the SRX5K-SCB (SCB1) and the SRX5K-SCBE (SCB2), except for in-service software upgrade (ISSU) and in-service hardware upgrade (ISHU). ◦ SCB3 has a second external Ethernet port. ◦ Fabric bandwidth increasing mode is not supported.

Table 50: show chassis hardware Output Fields *(continued)*

<table>
<tr><th>Field Name</th><th>Field Description</th></tr>
<tr><td></td><td>

Type of Flexible PIC Concentrator (FPC), Physical Interface Card (PIC), Modular Interface Cards (MICs), and PIMs.
IOCs
Starting with Junos OS Release 15.1X49-D10, the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) are introduced.

IOC3 has two types of IOC3 MPCs, which have different built-in MICs: the 24x10GE + 6x40GE MPC and the 2x100GE + 4x10GE MPC.
IOC3 supports SCB3 and SRX5000 line backplane and enhanced backplane.
IOC3 can only work with SRX5000 line SCB2 and SCB3. If an SRX5000 line SCB is detected, IOC3 is offline, an FPC misconfiguration alarm is raised, and a system log message is generated.
IOC3 interoperates with SCB2 and SCB3.
IOC3 interoperates with the SRX5K-SPC-4-15-320 (SPC2) and the SRX5K-MPC (IOC2).
The maximum power consumption for one IOC3 is 645W. An enhanced power module must be used.
The IOC3 does not support the following command to set a PIC to go offline or online:
request chassis pic fpc-slot <fpc-slot> pic-slot <pic-slot> <offline | online> .
IOC3 supports 240 Gbps of throughput with the enhanced SRX5000 line backplane.
Chassis cluster functions the same as for the SRX5000 line IOC2.
IOC3 supports intra-chassis and inter-chassis fabric redundancy mode.
IOC3 supports ISSU and ISHU in chassis cluster mode.
IOC3 supports intra-FPC and and Inter-FPC Express Path (previously known as services offloading) with IPv4.
NAT of IPv4 and IPv6 in normal mode and IPv4 for Express Path mode.
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.

NOTE: Fabric bandwidth increasing mode is not supported on IOC3.
SRX Clustering Module (SCM)
Fan tray
For hosts, the Routing Engine type.

Starting with Junos OS Release 12.1X47-D15, the SRX5K-RE-1800X4 (RE2) Routing Engine is introduced.
The RE2 has an Intel Quad core Xeon processor, 16 GB of DRAM, and a 128-GB solid-state drive (SSD).
The number 1800 refers to the speed of the processor (1.8 GHz). The maximum required power for this Routing Engine is 90W.
NOTE: The RE2 provides significantly better performance than the previously used Routing Engine, even with a single core.

</td></tr>
</table>

show chassis hardware

show chassis hardware

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                CM0715AK0021      SRX1500
Midplane         REV 08   750-058562   ACMA4255          SRX1500
CB 0             REV 08   711-053838   ACMA7529          CPU Board SRX700E
Routing Engine 0          BUILTIN      BUILTIN           SRX Routing Engine
FPC 0            REV 07   711-053832   ACMA3311          FEB
  PIC 0                   BUILTIN      BUILTIN           12x1G-T-4x1G-SFP-4x10G
    Xcvr 12      REV 01   740-014132   61521013          SFP-T
    Xcvr 13      REV 02   740-013111   A281604           SFP-T
    Xcvr 14      REV 02   740-011613   NRN30NV           SFP-SX
    Xcvr 15      REV 02   740-011613   NRN2PWV           SFP-SX
    Xcvr 16      REV 01   740-021308   AJA17B5           SFP+-10G-SR
    Xcvr 17      REV 01   740-021308   MSP056B           SFP+-10G-SR
    Xcvr 18      REV 01   740-031980   AS920WJ           SFP+-10G-SR
    Xcvr 19      REV 01   740-031980   AS92W5N           SFP+-10G-SR
Power Supply 0   REV 01   740-055217   1EDP42500JZ       PS 400W 90-264V AC in
Fan Tray 0                                              SRX1500 0, Front to Back
 Airflow - AFO
Fan Tray 1                                              SRX1500 1, Front to Back
 Airflow - AFO
Fan Tray 2                                              SRX1500 2, Front to Back
 Airflow - AFO
Fan Tray 3                                              SRX1500 3, Front to Back
 Airflow - AFO
```

show chassis hardware (SRX5600 and SRX5800 devices for SRX5K-MPC)

```
user@host> show chassis hardware
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                JN12170EAAGA      SRX 5800
Midplane         REV 01   710-041799   ACAX3849          SRX 5800 Backplane
FPM Board        REV 01   710-024632   CAAX7297          Front Panel Display
PDM              Rev 03   740-013110   QCS170250DU       Power Distribution Modu
le
PEM 0            Rev 03   740-034724   QCS17020203F      PS 4.1kW; 200-240V AC i
n
PEM 1            Rev 03   740-034724   QCS17020203C      PS 4.1kW; 200-240V AC i
n
PEM 2            Rev 04   740-034724   QCS17100200A      PS 4.1kW; 200-240V AC i
n
PEM 3            Rev 03   740-034724   QCS17080200M      PS 4.1kW; 200-240V AC i
n
Routing Engine 0 REV 11   740-023530   9012047437        SRX5k RE-13-20
CB 0             REV 09   710-024802   CAAX7202          SRX5k SCB
CB 1             REV 09   710-024802   CAAX7157          SRX5k SCB
FPC 0            REV 07   750-044175   CAAD0791          SRX5k SPC II
  CPU                     BUILTIN      BUILTIN           SRX5k DPC PPC
  PIC 0                   BUILTIN      BUILTIN           SPU Cp
  PIC 1                   BUILTIN      BUILTIN           SPU Flow
  PIC 2                   BUILTIN      BUILTIN           SPU Flow
  PIC 3                   BUILTIN      BUILTIN           SPU Flow
FPC 1            REV 07   750-044175   CAAD0751          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 2            REV 28    750-020751   CAAW1817           SRX5k DPC 4X 10GE
  CPU            REV 04    710-024633   CAAZ5269           SRX5k DPC PMB
  PIC 0                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
    Xcvr 0       REV 02    740-014289   T10A00404          XFP-10G-SR
  PIC 1                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
  PIC 2                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
  PIC 3                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
FPC 6            REV 02    750-044175   ZY2552             SRX5k SPC II
  CPU                      BUILTIN      BUILTIN            SRX5k DPC PPC
FPC 9            REV 10    750-044175   CAAP5932           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 10           REV 22    750-043157   ZH8192             SRX5k IOC II    CPU
      REV 08    711-043360    YX3879             SRX5k MPC PMB
  MIC 0          REV 01    750-049488   YZ2084             10x 10GE SFP+
    PIC 0                  BUILTIN      BUILTIN            10x 10GE SFP+
      Xcvr 0     REV 01    740-031980   AMB0HG3            SFP+-10G-SR
      Xcvr 1     REV 01    740-031980   AM20B6F            SFP+-10G-SR
  MIC 1          REV 19    750-049486   CAAH3504           1x 100GE CFP
    PIC 2                  BUILTIN      BUILTIN            1x 100GE CFP
      Xcvr 0     REV 01    740-035329   X000D375           CFP-100G-SR10
FPC 11           REV 07.04.07 750-043157 CAAJ8771          SRX5k IOC II    CPU
      REV 08    711-043360    CAAJ3881           SRX5k MPC PMB
  MIC 0          REV 19    750-049486   CAAH0979           1x 100GE CFP
    PIC 0                  BUILTIN      BUILTIN            1x 100GE CFP
      Xcvr 0     REV 01    740-035329   UP1020Z            CFP-100G-SR10
  MIC 1          REV 08    750-049487   CAAM1160           2x 40GE QSFP+
    PIC 2                  BUILTIN      BUILTIN            2x 40GE QSFP+
      Xcvr 0     REV 01    740-032986   QB151094           QSFP+-40G-SR4
      Xcvr 1     REV 01    740-032986   QB160509           QSFP+-40G-SR4
Fan Tray 0       REV 04    740-035409   ACAE0875           Enhanced Fan Tray
Fan Tray 1       REV 04    740-035409   ACAE0876           Enhanced Fan Tray
```

show chassis hardware (with 20-Gigabit Ethernet MIC with SFP)

```
user@host> show chassis hardware
Hardware inventory:
Item             Version   Part number  Serial number      Description
Chassis                                 JN108DA5AAGA       SRX 5800
Midplane         REV 02    710-013698   TR0037             SRX 5600 Midplane
FPM Board        REV 02    710-014974   JY4635             Front Panel Display
PDM             Rev 02    740-013110    QCS10465005       Power Distribution Module
PEM 0            Rev 03    740-023514   QCS11154040        PS 1.7kW; 200-240VAC in
PEM 2            Rev 02    740-023514   QCS10504014        PS 1.7kW; 200-240VAC in
Routing Engine 0 REV 05    740-015113   1000681023         RE-S-1300
CB 0             REV 05    710-013385   JY4775             SRX5k SCB
FPC 1            REV 17    750-020751   WZ6349             SRX5k DPC 4X 10GE
  CPU            REV 02    710-024633   WZ0718             SRX5k DPC PMB
  PIC 0                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
    Xcvr 0                 NON-JNPR     C724XM088          XFP-10G-SR
  PIC 1                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
    Xcvr 0       REV 02    740-011571   C831XJ08S          XFP-10G-SR
  PIC 2                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
  PIC 3                    BUILTIN      BUILTIN            1x 10GE(LAN/WAN) RichQ
FPC 3            REV 22    750-043157   ZH8189             SRX5k IOC II
```

```
  CPU           REV 06   711-043360   YX3912          SRX5k MPC PMB
  MIC 0         REV 01   750-055732   CACF9115        20x 1GE(LAN) SFP
    PIC 0                BUILTIN      BUILTIN         10x 1GE(LAN) SFP
      Xcvr 2    REV 02   740-013111   B358549         SFP-T
      Xcvr 9    REV 02   740-011613   PNB1FQS         SFP-SX
    PIC 1                BUILTIN      BUILTIN         10x 1GE(LAN) SFP
      Xcvr 9    REV 02   740-011613   PNB1FFF         SFP-SX
FPC 5           REV 01   750-027945   JW9665          SRX5k FIOC
  CPU
FPC 8           REV 08   750-023996   XA7234          SRX5k SPC
  CPU           REV 02   710-024633   XA1599          SRX5k DPC PMB
  PIC 0                  BUILTIN      BUILTIN         SPU Cp-Flow
  PIC 1                  BUILTIN      BUILTIN         SPU Flow
Fan Tray 0      REV 03   740-014971   TP0902          Fan Tray
Fan Tray 1      REV 01   740-014971   TP0121          Fan Tray
```

show chassis hardware (SRX5600 and SRX5800 devices with SRX5000 line SRX5K-SCBE [SCB2] and SRX5K-RE-1800X4 [RE2])

```
user@host> show chassis hardware
node0:
--------------------------------------------------------------------------
Hardware inventory:
Item            Version  Part number  Serial number   Description
Chassis                               JN1251EA1AGB    SRX5600
Midplane       REV 01   760-063936   ACRE2657        Enhanced SRX5600 Midplane
FPM Board       REV 01   710-024631   CABY3551        Front Panel Display
PEM 0           Rev 03   740-034701   QCS13380901P    PS 1.4-2.6kW; 90-264V
AC in
PEM 1           Rev 03   740-034701   QCS133809019    PS 1.4-2.6kW; 90-264V
AC in
Routing Engine 0 REV 02  740-056658   9009210105      SRX5k RE-1800X4
Routing Engine 1 REV 02  740-056658   9013115551      SRX5k RE-1800X4
CB 0            REV 01   750-062257   CADW3663        SRX5k SCB3
CB 1            REV 01   750-062257   CADZ3263        SRX5k SCB3
FPC 0           REV 18   750-054877   CABG6043        SRX5k SPC II
  CPU                    BUILTIN      BUILTIN         SRX5k DPC PPC
  PIC 0                  BUILTIN      BUILTIN         SPU Cp
  PIC 1                  BUILTIN      BUILTIN         SPU Flow
  PIC 2                  BUILTIN      BUILTIN         SPU Flow
  PIC 3                  BUILTIN      BUILTIN         SPU Flow
FPC 1           REV 01   750-062243   CAEE5918        SRX5k IOC3 24XGE+6XLG
  CPU           REV 02   711-062244   CADX8509        RMPC PMB
  PIC 0                  BUILTIN      BUILTIN         12x 10GE SFP+
    Xcvr 0      REV 01   740-031980   273363A01891    SFP+-10G-SR
    Xcvr 1      REV 01   740-031980   273363A01915    SFP+-10G-SR
    Xcvr 2      REV 01   740-031980   ANA0BK6         SFP+-10G-SR
    Xcvr 3      REV 01   740-031980   AP407GA         SFP+-10G-SR
    Xcvr 9      REV 01   740-021308   MUC20G1         SFP+-10G-SR
  PIC 1                  BUILTIN      BUILTIN         12x 10GE SFP+
  PIC 2                  BUILTIN      BUILTIN         3x 40GE QSFP+
  PIC 3                  BUILTIN      BUILTIN         3x 40GE QSFP+
  WAN MEZZ      REV 15   750-049136   CAEE5845        MPC5E 24XGE OTN Mezz
FPC 3           REV 11   750-043157   CACL7452        SRX5k IOC II
  CPU           REV 04   711-043360   CACP1977        SRX5k MPC PMB
  MIC 0         REV 04   750-049488   CABL4759        10x 10GE SFP+
    PIC 0                BUILTIN      BUILTIN         10x 10GE SFP+
      Xcvr 0    REV 01   740-021308   CF36KM0SY       SFP+-10G-SR
      Xcvr 1    REV 01   740-021308   MUC0MF2         SFP+-10G-SR
      Xcvr 2    REV 01   740-021308   CF36KM01S       SFP+-10G-SR
      Xcvr 3    REV 01   740-021308   MUC229N         SFP+-10G-SR
```

```
FPC 5            REV 07   750-044175   CAAD0764          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
Fan Tray                                                 Enhanced Fan Tray

node1:
--------------------------------------------------------------------------
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                               JN124FE77AGB      SRX5600
Midplane        REV 01   760-063936   ACRE2970         Enhanced SRX5600 Midplane
FPM Board        REV 01   710-024631   CABY3552          Front Panel Display
PEM 0            Rev 03   740-034701   QCS133809028      PS 1.4-2.6kW; 90-264V
AC in
PEM 1            Rev 03   740-034701   QCS133809027      PS 1.4-2.6kW; 90-264V
AC in
Routing Engine 0 REV 02   740-056658   9009218294        SRX5k RE-1800X4
Routing Engine 1 REV 02   740-056658   9013104758        SRX5k RE-1800X4
CB 0             REV 01   750-062257   CAEB8180          SRX5k SCB3
CB 1             REV 01   750-062257   CADZ3334          SRX5k SCB3
FPC 0            REV 18   750-054877   CACJ9834          SRX5k SPC II
  CPU                     BUILTIN      BUILTIN           SRX5k DPC PPC
  PIC 0                   BUILTIN      BUILTIN           SPU Cp
  PIC 1                   BUILTIN      BUILTIN           SPU Flow
  PIC 2                   BUILTIN      BUILTIN           SPU Flow
  PIC 3                   BUILTIN      BUILTIN           SPU Flow
FPC 1            REV 01   750-062243   CAEB0981          SRX5k IOC3 24XGE+6XLG
  CPU            REV 02   711-062244   CAEA4644          RMPC PMB
  PIC 0                   BUILTIN      BUILTIN           12x 10GE SFP+
    Xcvr 0       REV 01   740-031980   AP41BLH           SFP+-10G-SR
    Xcvr 1       REV 01   740-031980   AQ400SL           SFP+-10G-SR
    Xcvr 2       REV 01   740-031980   AP422LJ           SFP+-10G-SR
    Xcvr 3       REV 01   740-021308   AMG0RBT           SFP+-10G-SR
    Xcvr 9       REV 01   740-021308   MUC2FRG           SFP+-10G-SR
  PIC 1                   BUILTIN      BUILTIN           12x 10GE SFP+
  PIC 2                   BUILTIN      BUILTIN           3x 40GE QSFP+
  PIC 3                   BUILTIN      BUILTIN           3x 40GE QSFP+
  WAN MEZZ       REV 15   750-049136   CAEA4837          MPC5E 24XGE OTN Mezz
FPC 3            REV 11   750-043157   CACA8784          SRX5k IOC II
  CPU            REV 04   711-043360   CACA8820          SRX5k MPC PMB
  MIC 0          REV 05   750-049488   CADF0521          10x 10GE SFP+
    PIC 0                 BUILTIN      BUILTIN           10x 10GE SFP+
      Xcvr 0     REV 01   740-030658   AD1130A00PV       SFP+-10G-USR
      Xcvr 1     REV 01   740-031980   AN40MVV           SFP+-10G-SR
      Xcvr 2     REV 01   740-021308   CF36KM37B         SFP+-10G-SR
      Xcvr 3     REV 01   740-021308   AD153830DSZ       SFP+-10G-SR
  MIC 1          REV 01   750-049487   CABB5961          2x 40GE QSFP+
    PIC 2                 BUILTIN      BUILTIN           2x 40GE QSFP+
      Xcvr 1     REV 01   740-032986   QB160513          QSFP+-40G-SR4
FPC 5            REV 02   750-044175   ZY2569            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
Fan Tray                                                 Enhanced Fan Tray
```

show chassis hardware

(SRX5400, SRX5600, and SRX5800 devices with SRX5000 line SRX5K-SCB3 [SCB3] with enhanced midplanes and SRX5K-MPC3-100G10G [IOC3] or SRX5K-MPC3-40G10G [IOC3])

```
user@host> show chassis hardware
node0:
--------------------------------------------------------------------------
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                JN1250870AGB      SRX5600
Midplane        REV 01   760-063936   ACRE2578         Enhanced SRX5600 Midplane


FPM Board        REV 02   710-017254   KD9027            Front Panel Display
PEM 0            Rev 03   740-034701   QCS13090900T      PS 1.4-2.6kW; 90-264V A

                                     C in
PEM 1            Rev 03   740-034701   QCS13090904T      PS 1.4-2.6kW; 90-264V A

                                     C in
Routing Engine 0 REV 01   740-056658   9009196496        SRX5k RE-1800X4
CB 0             REV 01   750-062257   CAEC2501          SRX5k SCB3
FPC 0            REV 10   750-056758   CADC8067          SRX5k SPC II
  CPU                     BUILTIN      BUILTIN           SRX5k DPC PPC
  PIC 0                   BUILTIN      BUILTIN           SPU Cp
  PIC 1                   BUILTIN      BUILTIN           SPU Flow
  PIC 2                   BUILTIN      BUILTIN           SPU Flow
  PIC 3                   BUILTIN      BUILTIN           SPU Flow
FPC 2            REV 01   750-062243   CAEE5924          SRX5k IOC3 24XGE+6XLG
  CPU            REV 01   711-062244   CAEB4890          SRX5k IOC3 PMB
  PIC 0                   BUILTIN      BUILTIN           12x 10GE SFP+
  PIC 1                   BUILTIN      BUILTIN           12x 10GE SFP+
  PIC 2                   BUILTIN      BUILTIN           3x 40GE QSFP+
    Xcvr 0       REV 01   740-038623   MOC13156230449    QSFP+-40G-CU1M
    Xcvr 2       REV 01   740-038623   MOC13156230449    QSFP+-40G-CU1M
  PIC 3                   BUILTIN      BUILTIN           3x 40GE QSFP+
  WAN MEZZ       REV 01   750-062682   CAEE5817          24x 10GE SFP+ Mezz
FPC 4            REV 11   750-043157   CACY1595          SRX5k IOC II
  CPU            REV 04   711-043360   CACZ8879          SRX5k MPC PMB
  MIC 1          REV 04   750-049488   CACM6062          10x 10GE SFP+
    PIC 2                 BUILTIN      BUILTIN           10x 10GE SFP+
      Xcvr 7     REV 01   740-021308   AD1439301TU       SFP+-10G-SR
      Xcvr 8     REV 01   740-021308   AD1439301SD       SFP+-10G-SR
      Xcvr 9     REV 01   740-021308   AD1439301TS       SFP+-10G-SR
FPC 5            REV 05   750-044175   ZZ1371            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
Fan Tray                                                 Enhanced Fan Tray

node1:
--------------------------------------------------------------------------
Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                JN124FEC0AGB      SRX5600
Midplane        REV 01   760-063936   ACRE2946         Enhanced SRX5600 Midplane
FPM Board        test     710-017254   test              Front Panel Display
PEM 0            Rev 01   740-038514   QCS114111003      DC 2.6kW Power Entry
Module
PEM 1            Rev 01   740-038514   QCS12031100J      DC 2.6kW Power Entry
```

```
Module
Routing Engine 0 REV 01   740-056658   9009186342       SRX5k RE-1800X4
CB 0             REV 01   750-062257   CAEB8178         SRX5k SCB3
FPC 0            REV 07   750-044175   CAAD0769         SRX5k SPC II
  CPU                     BUILTIN      BUILTIN          SRX5k DPC PPC
  PIC 0                   BUILTIN      BUILTIN          SPU Cp
  PIC 1                   BUILTIN      BUILTIN          SPU Flow
  PIC 2                   BUILTIN      BUILTIN          SPU Flow
  PIC 3                   BUILTIN      BUILTIN          SPU Flow
FPC 4            REV 11   750-043157   CACY1592         SRX5k IOC II
  CPU            REV 04   711-043360   CACZ8831         SRX5k MPC PMB
  MIC 1          REV 04   750-049488   CACN0239         10x 10GE SFP+
    PIC 2                 BUILTIN      BUILTIN          10x 10GE SFP+
      Xcvr 7     REV 01   740-031980   ARN23HW          SFP+-10G-SR
      Xcvr 8     REV 01   740-031980   ARN2FVW          SFP+-10G-SR
      Xcvr 9     REV 01   740-031980   ARN2YVM          SFP+-10G-SR
FPC 5            REV 10   750-056758   CADA8736         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
Fan Tray                                                Enhanced Fan Tray
```

show chassis hardware (SRX4200)

user@host> show chassis hardware

```
Hardware inventory:
Item             Version  Part number  Serial number    Description
Chassis                                DK2816AR0020     SRX4200
Mainboard        REV 01   650-071675   16061032317      SRX4200
Routing Engine 0          BUILTIN      BUILTIN          SRX Routing Engine
FPC 0                     BUILTIN      BUILTIN          FEB
  PIC 0                   BUILTIN      BUILTIN          8x10G-SFP
    Xcvr 0       REV 01   740-038153   MOC11511530020   SFP+-10G-CU3M
    Xcvr 1       REV 01   740-038153   MOC11511530020   SFP+-10G-CU3M
    Xcvr 2       REV 01   740-038153   MOC11511530020   SFP+-10G-CU3M
    Xcvr 3       REV 01   740-038153   MOC11511530020   SFP+-10G-CU3M
    Xcvr 4       REV 01   740-021308   04DZ06A00364     SFP+-10G-SR
    Xcvr 5       REV 01   740-031980   233363A03066     SFP+-10G-SR
    Xcvr 6       REV 01   740-021308   AL70SWE          SFP+-10G-SR
    Xcvr 7       REV 01   740-031980   ALN0N6C          SFP+-10G-SR
    Xcvr 8       REV 01   740-030076   APF16220018NK1   SFP+-10G-CU1M
Power Supply 0   REV 04   740-041741   1GA26241849      JPSU-650W-AC-AFO
Power Supply 1   REV 04   740-041741   1GA26241846      JPSU-650W-AC-AFO
Fan Tray 0                                            SRX4200 0, Front to Back
 Airflow - AFO
Fan Tray 1                                            SRX4200 1, Front to Back
 Airflow - AFO
Fan Tray 2                                            SRX4200 2, Front to Back
 Airflow - AFO
Fan Tray 3                                            SRX4200 3, Front to Back
 Airflow - AFO
```

show chassis hardware clei-models

show chassis hardware clei-models

(SRX5600 and SRX5800 devices with SRX5000 line SRX5K-SCBE [SCB2] and SRX5K-RE-1800X4 [RE2])

```
user@host> show chassis hardware clei-models node 1
node1:
--------------------------------------------------------------------------
Hardware inventory:
Item             Version  Part number  CLEI code         FRU model number
Midplane         REV 01   710-024803                     SRX5800-BP-A
FPM Board        REV 01   710-024632                     SRX5800-CRAFT-A
PEM 0            Rev 04   740-034724                     SRX5800-PWR-4100-AC
PEM 1            Rev 05   740-034724                     SRX5800-PWR-4100-AC
Routing Engine 0 REV 01   740-056658   COUCATTBAA        SRX5K-RE-1800X4
CB 0             REV 01   750-056587   COUCATSBAA        SRX5K-SCBE
CB 1             REV 01   750-056587   COUCATSBAA        SRX5K-SCBE
CB 2             REV 01   750-056587   COUCATSBAA        SRX5K-SCBE
FPC 0            REV 18   750-054877   COUCATLBAA        SRX5K-SPC-4-15-320
  CPU                     BUILTIN
FPC 1            REV 18   750-054877   COUCATLBAA        SRX5K-SPC-4-15-320
  CPU                     BUILTIN
FPC 2            REV 18   750-054877   COUCATLBAA        SRX5K-SPC-4-15-320
  CPU                     BUILTIN
FPC 3            REV 11   750-043157   COUIBCWBAA        SRX5K-MPC
  MIC 0          REV 05   750-049486   COUIBCYBAA        SRX-MIC-1X100G-CFP
  MIC 1          REV 04   750-049488   COUIBCXBAA        SRX-MIC-10XG-SFPP
FPC 4            REV 18   750-054877   COUCATLBAA        SRX5K-SPC-4-15-320
  CPU                     BUILTIN
FPC 7            REV 18   750-054877   COUCATLBAA        SRX5K-SPC-4-15-320
  CPU                     BUILTIN
FPC 8            REV 11   750-043157   COUIBCWBAA        SRX5K-MPC
  MIC 0          REV 05   750-049486   COUIBCYBAA        SRX-MIC-1X100G-CFP
FPC 9            REV 18   750-054877   COUCATLBAA        SRX5K-SPC-4-15-320
  CPU                     BUILTIN
FPC 10           REV 18   750-054877   COUCATLBAA        SRX5K-SPC-4-15-320
  CPU                     BUILTIN
Fan Tray 0       REV 04   740-035409                     SRX5800-HC-FAN
Fan Tray 1       REV 04   740-035409                     SRX5800-HC-FAN
```

show chassis routing-engine (View)

Supported Platforms SRX Series, vSRX

Syntax show chassis routing-engine

Release Information Command introduced in Junos OS Release 9.5.

Description Display the Routing Engine status of the chassis cluster.

Required Privilege Level view

Related Documentation

- cluster (Chassis) on page 318
- *request system snapshot (SRX Series)*

List of Sample Output

show chassis routing-engine (Sample 1 - SRX550M) on page 451
show chassis routing-engine (Sample 2- vSRX) on page 451

Output Fields Table 51 on page 450 lists the output fields for the **show chassis routing-engine** command. Output fields are listed in the approximate order in which they appear.

Table 51: show chassis routing-engine Output Fields

Field Name	Field Description
Temperature	Routing Engine temperature. (Not available for vSRX deployments.)
CPU temperature	CPU temperature. (Not available for vSRX deployments.)
Total memory	Total memory available on the system.
Control plane memory	Memory available for the control plane.
Data plane memory	Memory reserved for data plane processing.
CPU utilization	Current CPU utilization statistics on the control plane core.
User	Current CPU utilization in user mode on the control plane core.
Background	Current CPU utilization in nice mode on the control plane core.
Kernel	Current CPU utilization in kernel mode on the control plane core.
Interrupt	Current CPU utilization in interrupt mode on the control plane core.
Idle	Current CPU utilization in idle mode on the control plane core.
Model	Routing Engine model.

Table 51: show chassis routing-engine Output Fields *(continued)*

Field Name	Field Description
Start time	Routing Engine start time.
Uptime	Length of time the Routing Engine has been up (running) since the last start.
Last reboot reason	Reason for the last reboot of the Routing Engine.
Load averages	The average number of threads waiting in the run queue or currently executing over 1-, 5-, and 15-minute periods.

Sample Output

show chassis routing-engine (Sample 1 - SRX550M)

```
user@host> show chassis routing-engine
Routing Engine status:
    Temperature                 38 degrees C / 100 degrees F
    CPU temperature             36 degrees C / 96 degrees F
    Total memory               512 MB Max   435 MB used ( 85 percent)
      Control plane memory     344 MB Max   296 MB used ( 86 percent)
      Data plane memory        168 MB Max   138 MB used ( 82 percent)
    CPU utilization:
      User                       8 percent
      Background                 0 percent
      Kernel                     4 percent
      Interrupt                  0 percent
      Idle                      88 percent
    Model                          RE-SRX5500-LOWMEM
    Serial ID                      AAAP8652
    Start time                     2009-09-21 00:04:54 PDT
    Uptime                         52 minutes, 47 seconds
    Last reboot reason             0x200:chassis control reset
    Load averages:                 1 minute   5 minute  15 minute
                                       0.12       0.15       0.10
```

Sample Output

show chassis routing-engine (Sample 2- vSRX)

```
user@host> show chassis routing-engine
Routing Engine status:
    Total memory              1024 MB Max   358 MB used ( 35 percent)
      Control plane memory    1024 MB Max   358 MB used ( 35 percent)
    5 sec CPU utilization:
      User                       2 percent
      Background                 0 percent
      Kernel                     4 percent
      Interrupt                  6 percent
      Idle                      88 percent
    Model                          VSRX RE
    Start time                     2015-03-03 07:04:18 UTC
    Uptime                         2 days, 11 hours, 51 minutes, 11 seconds
    Last reboot reason             Router rebooted after a normal shutdown.
    Load averages:                 1 minute   5 minute  15 minute
                                       0.07       0.04       0.06
```

show configuration chassis cluster traceoptions

Supported Platforms SRX Series, vSRX

Syntax show configuration chassis cluster traceoptions

Release Information Command introduced in Junos OS Release 12.1.

Description Display tracing options for the chassis cluster redundancy process.

Required Privilege Level view

Related Documentation

- cluster (Chassis) on page 318
- traceoptions (Chassis Cluster) on page 353

List of Sample Output show configuration chassis cluster traceoptions on page 453

Output Fields Table 52 on page 453 lists the output fields for the **show configuration chassis cluster traceoptions** command. Output fields are listed in the approximate order in which they appear.

Table 52: show configuration chassis cluster traceoptions Output Fields

Field Name	Field Description
file	Name of the file that receives the output of the tracing operation.
size	Size of each trace file.
files	Maximum number of trace files.

Sample Output

show configuration chassis cluster traceoptions

```
user@host> show configuration chassis cluster traceoptions
file chassis size 10k files 300;
level all;
```


show security internal-security-association

Supported Platforms

Syntax **show security internal-security-association**

Release Information Command introduced in Junos OS Release 12.1X47-D10.

Description Provide secure login by enabling the internal security association in a chassis cluster configuration.

Required Privilege Level view

Related Documentation
- *Chassis Cluster Feature Guide for Branch SRX Series Devices*

List of Sample Output show security internal-security-association on page 454

Output Fields Table 53 on page 454 lists the output fields for the **show security internal-security-association** command. Output fields are listed in the approximate order in which they appear.

Table 53: show security internal-security-association Output Fields

Field Name	Field Description
Internal SA Status	State of the internal SA option on the chassis cluster control link: **enabled** or **disabled**.
Iked Encryption Status	State of the iked encryption.

Sample Output

show security internal-security-association

```
user@host>show security internal-security-association

node0:
--------------------------------------------------------------------------
Internal SA Status : Enabled
Iked Encryption Status : Enabled

node1:
--------------------------------------------------------------------------
Internal SA Status : Enabled
Iked Encryption Status : Enabled
```

