



---

Junos<sup>®</sup> OS

# Chassis Cluster Feature Guide for Branch SRX Series Devices

Release  
15.1X49-D70



---

Modified: 2016-11-21

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

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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

*Junos<sup>®</sup> OS Chassis Cluster Feature Guide for Branch SRX Series Devices*  
15.1X49-D70  
Copyright © 2016, Juniper Networks, Inc.  
All rights reserved.

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

#### YEAR 2000 NOTICE

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

#### END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <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

	About the Documentation . . . . .	xv
	Documentation and Release Notes . . . . .	xv
	Supported Platforms . . . . .	xv
	Using the Examples in This Manual . . . . .	xv
	Merging a Full Example . . . . .	xvi
	Merging a Snippet . . . . .	xvi
	Documentation Conventions . . . . .	xvii
	Documentation Feedback . . . . .	xix
	Requesting Technical Support . . . . .	xix
	Self-Help Online Tools and Resources . . . . .	xix
	Opening a Case with JTAC . . . . .	xx
<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Introduction to Chassis Cluster . . . . .</b>	<b>3</b>
	Chassis Cluster Overview . . . . .	3
	High Availability Using Chassis Clusters . . . . .	3
	How High Availability Is Achieved by Chassis Cluster . . . . .	3
	Chassis Cluster Active/Active and Active/Passive Modes . . . . .	4
	Chassis Cluster Functionality . . . . .	4
	IPv6 Clustering Support . . . . .	4
	Chassis Cluster Supported Features . . . . .	5
	Chassis Cluster Features Support . . . . .	21
	Chassis Cluster Limitations . . . . .	23
<b>Chapter 2</b>	<b>Understanding Chassis Cluster License Requirements . . . . .</b>	<b>27</b>
	Understanding Chassis Cluster Licensing Requirements . . . . .	27
	Installing Licenses on the Devices in a Chassis Cluster . . . . .	28
	Verifying Licenses for an SRX Series Device in a Chassis Cluster . . . . .	30
<b>Chapter 3</b>	<b>Planning Your Chassis Cluster Configuration . . . . .</b>	<b>33</b>
	Preparing Your Equipment for Chassis Cluster Formation . . . . .	33
	SRX Series Chassis Cluster Configuration Overview . . . . .	34
<b>Part 2</b>	<b>Setting Up Chassis Cluster in SRX Series Devices</b>	
<b>Chapter 4</b>	<b>Chassis Cluster Physical Setup . . . . .</b>	<b>41</b>
	Connecting SRX Series Devices to Create a Chassis Cluster . . . . .	41

<b>Chapter 5</b>	<b>Setting Up Chassis Cluster Identification . . . . .</b>	<b>45</b>
	Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming . . . . .	45
	Example: Setting the Chassis Cluster Node ID and Cluster ID for Branch SRX Series Devices . . . . .	49
<b>Chapter 6</b>	<b>Setting Up Chassis Cluster Management Interfaces . . . . .</b>	<b>51</b>
	Management Interface on an Active Chassis Cluster . . . . .	51
	Example: Configuring the Chassis Cluster Management Interface . . . . .	51
<b>Chapter 7</b>	<b>Setting Up Fabric Interfaces on a Chassis Cluster . . . . .</b>	<b>55</b>
	Understanding Chassis Cluster Fabric Interfaces . . . . .	55
	Understanding Chassis Cluster Fabric Links . . . . .	56
	Understanding Session RTOs . . . . .	57
	Understanding Data Forwarding . . . . .	57
	Understanding Fabric Data Link Failure and Recovery . . . . .	57
	Example: Configuring the Chassis Cluster Fabric Interfaces . . . . .	59
	Verifying Chassis Cluster Data Plane Interfaces . . . . .	61
	Verifying Chassis Cluster Data Plane Statistics . . . . .	61
	Clearing Chassis Cluster Data Plane Statistics . . . . .	62
<b>Chapter 8</b>	<b>Setting Up Control Plane Interfaces on a Chassis Cluster . . . . .</b>	<b>63</b>
	Understanding Chassis Cluster Control Plane and Control Links . . . . .	63
	Understanding the Chassis Cluster Control Plane . . . . .	63
	Understanding Chassis Cluster Control Links . . . . .	64
	Verifying Chassis Cluster Control Plane Statistics . . . . .	64
	Clearing Chassis Cluster Control Plane Statistics . . . . .	65
<b>Chapter 9</b>	<b>Setting Up Chassis Cluster Redundancy Groups . . . . .</b>	<b>67</b>
	Understanding Chassis Cluster Redundancy Groups . . . . .	67
	Understanding Chassis Cluster Redundancy Group 0: Routing Engines . . . . .	68
	Understanding Chassis Cluster Redundancy Groups 1 Through 128 . . . . .	69
	Example: Configuring Chassis Cluster Redundancy Groups . . . . .	71
<b>Chapter 10</b>	<b>Setting Up Chassis Cluster Redundant Ethernet Interfaces . . . . .</b>	<b>75</b>
	Understanding Chassis Cluster Redundant Ethernet Interfaces . . . . .	75
	Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses . . . . .	77
	Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster . . . . .	82
<b>Chapter 11</b>	<b>Configuring Chassis Cluster . . . . .</b>	<b>85</b>
	Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster . . . . .	85
	Verifying a Chassis Cluster Configuration . . . . .	97
	Verifying Chassis Cluster Statistics . . . . .	97
	Clearing Chassis Cluster Statistics . . . . .	99

<b>Part 3</b>	<b>Managing Chassis Cluster Operations</b>	
<b>Chapter 12</b>	<b>Monitoring Chassis Cluster</b>	<b>103</b>
	Understanding Chassis Cluster Redundancy Group Interface Monitoring	103
	Example: Configuring Chassis Cluster Interface Monitoring	104
	Understanding Chassis Cluster Redundancy Group IP Address Monitoring	131
	Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring	134
<b>Chapter 13</b>	<b>Managing Chassis Cluster Redundancy Group Failover</b>	<b>139</b>
	Understanding Chassis Cluster Redundancy Group Failover	139
	Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers	140
	Understanding Chassis Cluster Redundancy Group Manual Failover	141
	Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover	143
	Initiating a Chassis Cluster Manual Redundancy Group Failover	144
	Verifying Chassis Cluster Failover Status	146
	Clearing Chassis Cluster Failover Status	147
<b>Chapter 14</b>	<b>Configuring Chassis Cluster Dual Fabric Links to Increase Redundancy and Performance</b>	<b>149</b>
	Understanding Chassis Cluster Dual Fabric Links	149
	Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports	150
	Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports	152
<b>Chapter 15</b>	<b>Configuring Route Advertisement over Redundant Ethernet Interfaces in a Chassis Cluster</b>	<b>157</b>
	Understanding Conditional Route Advertising in a Chassis Cluster	157
	Example: Configuring Conditional Route Advertising in a Chassis Cluster	158
<b>Chapter 16</b>	<b>Configuring Redundant Ethernet LAG Interfaces for Increasing High Availability and Overall Throughput</b>	<b>163</b>
	Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups	163
	Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups	165
	Understanding Chassis Cluster Redundant Ethernet Interface LAG Failover	168
	Scenario 1: Monitored Interface Weight Is 255	169
	Scenario 2: Monitored Interface Weight Is 75	170
	Scenario 3: Monitored Interface Weight Is 100	170
	Understanding LACP on Chassis Clusters	171
	Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups	171
	Minimum Links	172
	Sub-LAGs	172
	Supporting Hitless Failover	173
	Managing Link Aggregation Control PDUs	173
	Example: Configuring LACP on Chassis Clusters	173
	Example: Configuring Chassis Cluster Minimum Links	176

<b>Chapter 17</b>	<b>Simplifying Chassis Cluster Management</b> . . . . .	<b>179</b>
	Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes . . . . .	179
	Verifying Chassis Cluster Configuration Synchronization Status . . . . .	180
	NTP Time Synchronization on SRX Series Devices . . . . .	181
	Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP . . . . .	181
<b>Part 4</b>	<b>Additional Chassis Cluster Configurations</b>	
<b>Chapter 18</b>	<b>Configuring Active/Passive Chassis Cluster Deployments</b> . . . . .	<b>191</b>
	Understanding Active/Passive Chassis Cluster Deployment . . . . .	191
	Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) . . . . .	192
	Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) . . . . .	203
	Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel . . . . .	205
	Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel . . . . .	206
	Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) . . . . .	221
<b>Chapter 19</b>	<b>Enabling Multicast Routing or Asymmetric Routing</b> . . . . .	<b>225</b>
	Understanding Multicast Routing on a Chassis Cluster . . . . .	225
	Understanding PIM Data Forwarding . . . . .	226
	Understanding Multicast and PIM Session Synchronization . . . . .	226
	Understanding Asymmetric Routing Chassis Cluster Deployment . . . . .	226
	Understanding Failures in the Trust Zone Redundant Ethernet Interface . . . . .	227
	Understanding Failures in the Untrust Zone Interfaces . . . . .	227
	Example: Configuring an Asymmetric Chassis Cluster Pair . . . . .	228
<b>Chapter 20</b>	<b>Configuring Chassis Cluster Layer 2 Ethernet Switching</b> . . . . .	<b>241</b>
	Layer 2 Ethernet Switching Capability in Chassis Cluster Mode . . . . .	241
	Understanding Layer 2 Ethernet Switching Capability in Chassis Cluster on SRX Series Devices . . . . .	241
	Understanding Chassis Cluster Failover and New Primary Election . . . . .	242
	Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode (CLI) . . . . .	243
	Example: Configuring IRB and VLAN with Members Across Two Nodes (CLI) . . . . .	244
	Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI) . . . . .	246
<b>Chapter 21</b>	<b>Configuring Media Access Control Security (MACsec)</b> . . . . .	<b>249</b>
	Understanding Media Access Control Security (MACsec) for SRX Series . . . . .	249
	How MACsec Works . . . . .	249
	Understanding Connectivity Associations and Secure Channels . . . . .	250
	Understanding Static Connectivity Association Key Security Mode . . . . .	250
	MACsec Considerations . . . . .	251
	Configuring Media Access Control Security (MACsec) . . . . .	251
	Configuring MACsec Using Static Connectivity Association Key Security Mode . . . . .	253
	Configuring Static CAK on the Chassis Cluster Control Port . . . . .	257

	Configuring Static CAK on the Chassis Cluster Fabric Port . . . . .	258
	Considerations for Configuring MACsec on Control Ports . . . . .	258
	Configuring MACsec on Control Ports . . . . .	258
	Configuring MACsec on Fabric Ports . . . . .	259
<b>Part 5</b>	<b>Upgrading or Disabling Chassis Cluster</b>	
<b>Chapter 22</b>	<b>Upgrading Both Devices Separately . . . . .</b>	<b>263</b>
	Upgrading Individual Devices in a Chassis Cluster Separately . . . . .	263
<b>Chapter 23</b>	<b>Upgrading Both Devices Using ICU . . . . .</b>	<b>265</b>
	Upgrading Devices in a Chassis Cluster Using ICU . . . . .	265
	Upgrading Both Devices in a Chassis Cluster Using ICU . . . . .	265
	Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster . . . . .	266
	Upgrading ICU Using a Build Available on an FTP Server . . . . .	267
	Aborting an Upgrade in a Chassis Cluster During an ICU . . . . .	268
<b>Chapter 24</b>	<b>Disabling Chassis Cluster . . . . .</b>	<b>271</b>
	Disabling Chassis Cluster . . . . .	271
<b>Part 6</b>	<b>Configuration Statements and Operational Commands</b>	
<b>Chapter 25</b>	<b>Configuration Statements . . . . .</b>	<b>275</b>
	apply-groups (Chassis Cluster) . . . . .	277
	cak . . . . .	278
	ckn . . . . .	279
	cluster (Chassis) . . . . .	280
	configuration-synchronize (Chassis Cluster) . . . . .	281
	connectivity-association . . . . .	282
	connectivity-association (MACsec Interfaces) . . . . .	283
	control-link-recovery . . . . .	283
	device-count (Chassis Cluster) . . . . .	284
	no-encryption (MACsec) . . . . .	285
	exclude-protocol . . . . .	286
	ethernet (Chassis Cluster) . . . . .	287
	fabric-options . . . . .	288
	gether-options (Chassis Cluster) . . . . .	289
	global-threshold . . . . .	290
	global-weight . . . . .	291
	gratuitous-arp-count . . . . .	292
	heartbeat-interval . . . . .	293
	heartbeat-threshold . . . . .	294
	hold-down-interval . . . . .	295
	include-sci . . . . .	296
	interface (Chassis Cluster) . . . . .	297
	interfaces (MACsec) . . . . .	298
	interface-monitor . . . . .	299
	ip-monitoring . . . . .	300
	key-server-priority (MACsec) . . . . .	301

lacp (Interfaces) . . . . .	302
link-protection (Chassis Cluster) . . . . .	303
macsec . . . . .	304
member-interfaces . . . . .	305
mka . . . . .	305
must-secure . . . . .	306
network-management . . . . .	307
node (Chassis Cluster) . . . . .	308
node (Chassis Cluster Redundancy Group) . . . . .	308
offset . . . . .	309
preempt (Chassis Cluster) . . . . .	310
pre-shared-key . . . . .	311
priority (Chassis Cluster) . . . . .	312
redundancy-group (Chassis Cluster) . . . . .	313
redundancy-interface-process . . . . .	314
redundant-ether-options . . . . .	315
redundant-parent (Interfaces) . . . . .	316
redundant-pseudo-interface-options . . . . .	316
replay-protect . . . . .	317
replay-window-size . . . . .	318
reth-count (Chassis Cluster) . . . . .	319
reth (Interfaces) . . . . .	320
retry-count (Chassis Cluster) . . . . .	325
retry-interval (Chassis Cluster) . . . . .	326
route-active-on . . . . .	326
security-mode . . . . .	327
traceoptions (Chassis Cluster) . . . . .	328
transmit-interval (MACsec) . . . . .	330
weight . . . . .	331
<b>Chapter 26</b>	
<b>Operational Commands . . . . .</b>	<b>333</b>
clear chassis cluster control-plane statistics . . . . .	335
clear chassis cluster data-plane statistics . . . . .	336
clear chassis cluster failover-count . . . . .	337
clear chassis cluster ip-monitoring failure-count . . . . .	339
clear chassis cluster ip-monitoring failure-count ip-address . . . . .	340
clear chassis cluster statistics . . . . .	341
request chassis cluster configuration-synchronize . . . . .	342
request chassis cluster failover node . . . . .	343
request chassis cluster failover redundancy-group . . . . .	344
request chassis cluster failover reset . . . . .	345
request chassis fpc . . . . .	346
request system reboot . . . . .	347
request system software in-service-upgrade (Maintenance) . . . . .	348
request system software rollback (SRX Series) . . . . .	353
set chassis cluster cluster-id node reboot . . . . .	354
show chassis cluster control-plane statistics . . . . .	355
show chassis cluster data-plane interfaces . . . . .	357
show chassis cluster data-plane statistics . . . . .	358

---

show chassis cluster ethernet-switching interfaces . . . . .	360
show chassis cluster ethernet-switching status . . . . .	361
show chassis cluster information . . . . .	363
show chassis cluster information configuration-synchronization . . . . .	367
show chassis cluster interfaces . . . . .	369
show chassis cluster ip-monitoring status redundancy-group . . . . .	374
show chassis cluster statistics . . . . .	377
show chassis cluster status . . . . .	381
show chassis environment (Security) . . . . .	384
show chassis ethernet-switch . . . . .	388
show chassis fabric plane . . . . .	392
show chassis fabric plane-location . . . . .	398
show chassis fabric summary . . . . .	400
show chassis hardware (View) . . . . .	403
show chassis routing-engine (View) . . . . .	414
show configuration chassis cluster traceoptions . . . . .	417
show interfaces (Gigabit Ethernet) SRX device . . . . .	418
show security macsec connections . . . . .	432
show security macsec statistics for SRX device . . . . .	434
show security mka statistics . . . . .	438
show security mka sessions for SRX device . . . . .	440



# List of Figures

<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 3</b>	<b>Planning Your Chassis Cluster Configuration</b> . . . . .	<b>33</b>
	Figure 1: Chassis Cluster Flow Diagram . . . . .	35
<b>Part 2</b>	<b>Setting Up Chassis Cluster in SRX Series Devices</b>	
<b>Chapter 4</b>	<b>Chassis Cluster Physical Setup</b> . . . . .	<b>41</b>
	Figure 2: Connecting SRX Series Devices in a Cluster (SRX300 Devices) . . . . .	42
	Figure 3: Connecting SRX Series Devices in a Cluster (SRX320 Devices) . . . . .	42
	Figure 4: Connecting SRX Series Devices in a Cluster (SRX340 Devices) . . . . .	42
	Figure 5: Connecting SRX Series Devices in a Cluster (SRX345 Devices) . . . . .	42
	Figure 6: Connecting SRX Series Devices in a Cluster (SRX550M Devices) . . . . .	42
	Figure 7: Connecting SRX Series Devices in a Cluster (SRX1500 Devices) . . . . .	43
<b>Chapter 5</b>	<b>Setting Up Chassis Cluster Identification</b> . . . . .	<b>45</b>
	Figure 8: Slot Numbering in an SRX Series Chassis Cluster (SRX300 Devices) . . . . .	47
	Figure 9: Slot Numbering in an SRX Series Chassis Cluster (SRX320 Devices) . . . . .	48
	Figure 10: Slot Numbering in an SRX Series Chassis Cluster (SRX340 Devices) . . . . .	48
	Figure 11: Slot Numbering in an SRX Series Chassis Cluster (SRX345 Devices) . . . . .	48
	Figure 12: Slot Numbering in an SRX Series Chassis Cluster (SRX550M Devices) . . . . .	48
	Figure 13: Slot Numbering in an SRX Series Chassis Cluster (SRX1500 Devices) . . . . .	48
<b>Chapter 11</b>	<b>Configuring Chassis Cluster</b> . . . . .	<b>85</b>
	Figure 14: SRX Series for the Branch Topology Example . . . . .	87
<b>Part 3</b>	<b>Managing Chassis Cluster Operations</b>	
<b>Chapter 12</b>	<b>Monitoring Chassis Cluster</b> . . . . .	<b>103</b>
	Figure 15: SRX Series Chassis Cluster Interface Monitoring Topology Example . . . . .	106
<b>Chapter 15</b>	<b>Configuring Route Advertisement over Redundant Ethernet Interfaces in a Chassis Cluster</b> . . . . .	<b>157</b>
	Figure 16: Conditional Route Advertising . . . . .	158
	Figure 17: Conditional Route Advertising . . . . .	160
<b>Chapter 17</b>	<b>Simplifying Chassis Cluster Management</b> . . . . .	<b>179</b>
	Figure 18: Synchronizing Time From Peer Node Through Control Link . . . . .	182

<b>Part 4</b>	<b>Additional Chassis Cluster Configurations</b>	
<b>Chapter 18</b>	<b>Configuring Active/Passive Chassis Cluster Deployments . . . . .</b>	<b>191</b>
	Figure 19: Active/Passive Chassis Cluster Scenario . . . . .	192
	Figure 20: Active/Passive Chassis Cluster Topology . . . . .	193
	Figure 21: Active/Passive Chassis Cluster with IPsec Tunnel Scenario (SRX Series Devices) . . . . .	205
	Figure 22: Active/Passive Chassis Cluster with IPsec Tunnel Topology (SRX Series Devices) . . . . .	207
<b>Chapter 19</b>	<b>Enabling Multicast Routing or Asymmetric Routing . . . . .</b>	<b>225</b>
	Figure 23: Asymmetric Routing Chassis Cluster Scenario . . . . .	227
	Figure 24: Asymmetric Routing Chassis Cluster Topology . . . . .	229
<b>Chapter 20</b>	<b>Configuring Chassis Cluster Layer 2 Ethernet Switching . . . . .</b>	<b>241</b>
	Figure 25: Layer 2 Ethernet Switching Across Chassis Cluster Nodes . . . . .	242

# List of Tables

	<b>About the Documentation</b> . . . . .	<b>xv</b>
	Table 1: Notice Icons . . . . .	xvii
	Table 2: Text and Syntax Conventions . . . . .	xvii
<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Introduction to Chassis Cluster</b> . . . . .	<b>3</b>
	Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster . . . . .	5
	Table 4: Chassis Cluster Feature Support on Branch SRX Series Devices . . . . .	21
<b>Part 2</b>	<b>Setting Up Chassis Cluster in SRX Series Devices</b>	
<b>Chapter 5</b>	<b>Setting Up Chassis Cluster Identification</b> . . . . .	<b>45</b>
	Table 5: SRX Series Built-in Interfaces . . . . .	45
	Table 6: SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming . . . . .	46
<b>Chapter 9</b>	<b>Setting Up Chassis Cluster Redundancy Groups</b> . . . . .	<b>67</b>
	Table 7: Example of Redundancy Groups in a Chassis Cluster . . . . .	70
<b>Chapter 10</b>	<b>Setting Up Chassis Cluster Redundant Ethernet Interfaces</b> . . . . .	<b>75</b>
	Table 8: Maximum Number of Redundant Ethernet Interfaces Allowed . . . . .	76
<b>Chapter 11</b>	<b>Configuring Chassis Cluster</b> . . . . .	<b>85</b>
	Table 9: SRX Series Services Gateways Interface Renumbering . . . . .	87
	Table 10: SRX Series Services Gateways for the Branch Interface Settings . . . . .	88
<b>Part 4</b>	<b>Additional Chassis Cluster Configurations</b>	
<b>Chapter 18</b>	<b>Configuring Active/Passive Chassis Cluster Deployments</b> . . . . .	<b>191</b>
	Table 11: Group and Chassis Cluster Configuration Parameters . . . . .	194
	Table 12: Chassis Cluster Configuration Parameters . . . . .	194
	Table 13: Security Zone Configuration Parameters . . . . .	195
	Table 14: Security Policy Configuration Parameters . . . . .	195
	Table 15: Group and Chassis Cluster Configuration Parameters . . . . .	208
	Table 16: Chassis Cluster Configuration Parameters . . . . .	208
	Table 17: IKE Configuration Parameters . . . . .	209
	Table 18: IPsec Configuration Parameters . . . . .	209
	Table 19: Static Route Configuration Parameters . . . . .	210
	Table 20: Security Zone Configuration Parameters . . . . .	210
	Table 21: Security Policy Configuration Parameters . . . . .	210

<b>Chapter 19</b>	<b>Enabling Multicast Routing or Asymmetric Routing . . . . .</b>	<b>225</b>
	Table 22: Group and Chassis Cluster Configuration Parameters . . . . .	229
	Table 23: Chassis Cluster Configuration Parameters . . . . .	230
	Table 24: Security Zone Configuration Parameters . . . . .	230
	Table 25: Security Policy Configuration Parameters . . . . .	231
<b>Part 5</b>	<b>Upgrading or Disabling Chassis Cluster</b>	
<b>Chapter 23</b>	<b>Upgrading Both Devices Using ICU . . . . .</b>	<b>265</b>
	Table 26: request system software in-service-upgrade Output Fields . . . . .	269
<b>Part 6</b>	<b>Configuration Statements and Operational Commands</b>	
<b>Chapter 26</b>	<b>Operational Commands . . . . .</b>	<b>333</b>
	Table 27: show chassis cluster control-plane statistics Output Fields . . . . .	355
	Table 28: show chassis cluster data-plane interfaces Output Fields . . . . .	357
	Table 29: show chassis cluster data-plane statistics Output Fields . . . . .	358
	Table 30: show chassis cluster ethernet-switching interfaces Output Fields . . . . .	360
	Table 31: show chassis cluster ethernet-switching status Output Fields . . . . .	361
	Table 32: show chassis cluster information Output Fields . . . . .	363
	Table 33: show chassis cluster information configuration-synchronization Output Fields . . . . .	367
	Table 34: show chassis cluster interfaces Output Fields . . . . .	369
	Table 35: show chassis cluster ip-monitoring status Output Fields . . . . .	374
	Table 36: show chassis cluster ip-monitoring status redundancy group Reason Fields . . . . .	375
	Table 37: show chassis cluster statistics Output Fields . . . . .	377
	Table 38: show chassis cluster status Output Fields . . . . .	381
	Table 39: show chassis environment Output Fields . . . . .	384
	Table 40: show chassis ethernet-switch Output Fields . . . . .	388
	Table 41: show chassis fabric plane Output Fields . . . . .	392
	Table 42: show chassis fabric plane-location Output Fields . . . . .	398
	Table 43: show chassis fabric summary Output Fields . . . . .	400
	Table 44: show chassis hardware Output Fields . . . . .	403
	Table 45: show chassis routing-engine Output Fields . . . . .	414
	Table 46: show configuration chassis cluster traceoptions Output Fields . . . . .	417
	Table 47: show interfaces (Gigabit Ethernet) Output Fields . . . . .	419
	Table 48: Gigabit Ethernet IQ PIC Traffic and MAC Statistics by Interface Type . . . . .	428
	Table 49: show security macsec connections Output Fields . . . . .	432
	Table 50: show security macsec statistics Output Fields . . . . .	435
	Table 51: show security mka statistics Output Fields . . . . .	438
	Table 52: show security mka sessions Output Fields . . . . .	440

# About the Documentation

- Documentation and Release Notes on page xv
- Supported Platforms on page xv
- Using the Examples in This Manual on page xv
- Documentation Conventions on page xvii
- Documentation Feedback on page xix
- Requesting Technical Support on page xix

## 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]
```

- 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
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  user@host> <b>configure</b>

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

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host&gt; show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies guide names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS CLI User Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: <code>[edit]</code> <code>root@# set system domain-name <i>domain-name</i></code>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> <li>To configure a stub area, include the <code>stub</code> statement at the <code>[edit protocols ospf area area-id]</code> hierarchy level.</li> <li>The console port is labeled <code>CONSOLE</code>.</li> </ul>
< > (angle brackets)	Encloses optional keywords or variables.	<code>stub &lt;default-metric <i>metric</i>&gt;;</code>
(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.	<code>broadcast   multicast</code> <code>(<i>string1</i>   <i>string2</i>   <i>string3</i>)</code>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<code>rsvp { # Required for dynamic MPLS only</code>
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	<code>community name members [ <i>community-ids</i> ]</code>
Indentation and braces ( { } )	Identifies a level in the configuration hierarchy.	<code>[edit]</code> <code>routing-options {</code> <code>  static {</code> <code>    route default {</code> <code>      nexthop <i>address</i>;</code> <code>      retain;</code> <code>    }</code> <code>  }</code> <code>}</code>
:(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
<b>GUI Conventions</b>		
<b>Bold text like this</b>	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>

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 <b>Protocols&gt;Ospf</b> .

## 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](mailto: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

- [Introduction to Chassis Cluster on page 3](#)
- [Understanding Chassis Cluster License Requirements on page 27](#)
- [Planning Your Chassis Cluster Configuration on page 33](#)



## CHAPTER 1

# Introduction to Chassis Cluster

- [Chassis Cluster Overview on page 3](#)
- [Chassis Cluster Supported Features on page 5](#)
- [Chassis Cluster Limitations on page 23](#)

## Chassis Cluster Overview

---

**Supported Platforms** [SRX Series, vSRX](#)

- [High Availability Using Chassis Clusters on page 3](#)
- [How High Availability Is Achieved by Chassis Cluster on page 3](#)
- [Chassis Cluster Active/Active and Active/Passive Modes on page 4](#)
- [Chassis Cluster Functionality on page 4](#)
- [IPv6 Clustering Support on page 4](#)

## 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.

- 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.

- Related Documentation**
- [Preparing Your Equipment for Chassis Cluster Formation on page 33](#)
  - [Understanding Chassis Cluster Redundancy Groups on page 67](#)
  - [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 75](#)

## Chassis Cluster Supported Features

**Supported Platforms** [SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M](#)

[Table 3 on page 5](#) lists the features that are supported on branch SRX Series devices in a chassis cluster.

**Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster**

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Address Books and Address Sets</b>	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
<b>Administrator Authentication Support</b>	Local authentication	Yes	Yes	Yes	Yes
	RADIUS	Yes	Yes	Yes	Yes
	TACACS+	Yes	Yes	Yes	Yes
<b>Alarms</b>	Chassis alarms	Yes	Yes	Yes	Yes
	Interface alarms	Yes	Yes	Yes	Yes
	System alarms	Yes	Yes	Yes	Yes

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Application Identification</b> <sup>1</sup>	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 a Branch SRX Series Device 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 a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Attack Detection and Prevention (Screens)</b>	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
	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	

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
	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
<b>Chassis Management</b>	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	No	No	No	No
<b>Class of Service</b>	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	No	No	No	No
	Transmission queues	Yes	Yes	Yes	Yes

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>DHCP</b>	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 <sup>2</sup>	Yes	Yes	Yes	Yes
<b>Diagnostics Tools</b>	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
<b>Dynamic VPN</b>	Package dynamic VPN client <sup>3</sup>	–	–	–	–
<b>Ethernet Interfaces</b>	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	–	–	–	–
	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	No	No	No	No

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Ethernet Link Aggregation</b>	LACP/LAG cross IOC (inter-IOC)	–	–	–	–
	LACP (port priority) Layer 3 Mode	No	Yes	No	Yes
	LACP (port priority) Layer 2 Mode	No	Yes	No	Yes
	Layer 3 LAG on routed ports	Yes	Yes	Yes	Yes
	Static LAG (routing)	Yes	Yes	Yes	Yes
	Static LAG (switching)	Yes	Yes	Yes	Yes
	Switching mode	Yes	Yes	Yes	Yes
<b>File Management</b>	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/decryption of configuration files	Yes	Yes	Yes	Yes
	Management of account files	Yes	Yes	Yes	Yes
<b>Firewall Authentication</b>	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 a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Flow-Based and Packet-Based Processing</b>	Alarms and auditing	Yes	Yes	Yes	Yes
	End-to-end packet debugging	No	No	No	No
	Express Path support	No	No	No	No
	Flow-based processing	Yes	Yes	Yes	Yes
	Host bound fragmented traffic	No	No	No	No
	Network processor bundling	Yes	Yes	Yes	Yes
	Packet-based processing	No	No	No	No
	Selective stateless packet-based services	Yes	Yes	Yes	Yes
<b>GPRS</b>	GPRS (transparent mode and route mode)	No	No	No	No
<b>GTPv2</b>	IMSI prefix and APN filtering	No	No	No	No
	Message-length filtering	No	No	No	No
	Message-rate limiting	No	No	No	No
	Message-type filtering	No	No	No	No
	Packet sanity check	No	No	No	No
	Policy-based inspection	No	No	No	No
	Restart GTPv2 path	No	No	No	No
	Sequence-number and GTP-U validation	No	No	No	No
	Stateful inspection	No	No	No	No
	Traffic logging	No	No	No	No
	Tunnel cleanup	No	No	No	No

Table 3: Features Supported on a Branch SRX Series Device 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	No	No	No	No
	DSCP marking	No	No	No	No
	IDP and application identification	Yes	Yes	Yes	Yes
	IDP and UAC coordinated threat control	Yes	Yes	Yes	Yes
	IDP class-of-service action	No	No	No	No
	IDP inline tap mode	No	No	No	No
	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	No	No	No	No
	IPS rule base	Yes	Yes	Yes	Yes
	Jumbo frames	No	No	No	No
	Performance and capacity tuning for IDP	No	No	No	No
SNMP MIB for IDP monitoring	Yes	Yes	Yes	Yes	

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
IPsec	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	Yes	Yes	Yes	Yes
	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 a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Layer 2 Mode</b>	802.1x port-based network authentication	Yes	Yes	Yes	Yes
	Flexible Ethernet services	Yes	Yes	Yes	Yes
	IRB interface	Yes	Yes	Yes	Yes
	LLDP and LLDP-MED	Yes	Yes	Yes	Yes
	MAC limit (port security)	Yes	Yes	Yes	Yes
	Q-in-Q tunneling	No	No	No	No
	Spanning Tree Protocol	Yes	Yes	Yes	Yes
	VLAN retagging	Yes	Yes	Yes	Yes
	VLANs	Yes	Yes	Yes	Yes
<b>Multicast VPN</b>	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 a Branch SRX Series Device 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
	Source IP address translation	Yes	Yes	Yes	Yes
Static NAT	Yes	Yes	Yes	Yes	

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Network Operations and Troubleshooting Support</b>	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
<b>Packet Capture</b>	Packet capture	Yes	Yes	Yes	Yes
<b>Public Key Infrastructure</b>	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
<b>Remote Device Access</b>	Reverse Telnet	Yes	Yes	Yes	Yes
<b>RPM Probe</b>	RPM probe	Yes	Yes	Yes	Yes

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Routing</b>	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
Virtual Router Redundancy Protocol (VRRP)	Yes	Yes	Yes	Yes	
<b>Secure Web Access</b>	CAs	Yes	Yes	Yes	Yes
	HTTP	Yes	Yes	Yes	Yes
	HTTPS	Yes	Yes	Yes	Yes
<b>Security Policy</b>	Security policy	Yes	Yes	Yes	Yes

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (*continued*)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Security Zones</b>	Functional zone	Yes	Yes	Yes	Yes
	Security zone	Yes	Yes	Yes	Yes
<b>Session Logging</b>	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
<b>SMTP</b>	SMTP	Yes	Yes	Yes	Yes
<b>SNMP</b>	SNMP v1, v2, v3	No	No	No	No
<b>Stateless Firewall Filters</b>	Stateless firewall filters (ACLs)	No	No	No	No
<b>System Log Files</b>	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

Table 3: Features Supported on a Branch SRX Series Device in a Chassis Cluster (continued)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>Transparent Mode</b>	Bridge domain and transparent mode	No	No	No	No
	Class of service	No	No	No	No
<b>UTM</b>	Antispam	Yes	Yes	Yes	Yes
	Antivirus–Express	Yes	No	Yes	No
	Antivirus–Full	Yes	No	Yes	No
	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	Yes	Yes
	Web filtering–Juniper Networks local	Yes	Yes	Yes	Yes
	Web filtering–Surf-control	Yes	Yes	Yes	Yes
	Web filtering–Websense redirect	Yes	Yes	No	No
<b>Upgrading and Rebooting</b>	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	Yes	Yes	Yes	Yes
	ISSU	No	No	No	No
	WELF support	Yes	Yes	Yes	Yes

Table 3: Features Supported on a Branch SRX Series Device 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	No	No	No	No
	Network and Security Manager	Yes	Yes	Yes	Yes
	Session and Resource Control (SRC) application	No	No	No	No

<sup>1</sup> 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 will be lost. The application identification process will be applied on new sessions created on new primary node.

<sup>2</sup> DHCPv6 is supported on SRX Series devices running Junos OS Release 12.1 and later releases.

<sup>3</sup> Package Dynamic VPN client is supported on branch SRX Series devices until Junos OS Release 12.3X48.

## Chassis Cluster Features Support

Table 4 on page 21 lists the chassis cluster features that are supported on branch SRX Series devices.

Table 4: Chassis Cluster Feature Support on Branch SRX Series Devices

Features	Branch SRX Series
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
Chassis cluster formation	Yes
Encrypted control link	No

Table 4: Chassis Cluster Feature Support on Branch SRX Series Devices (*continued*)

Features	Branch SRX Series
Chassis clusters (active/backup and active/active)	Yes
Control link recovery	No
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)	No
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)	Yes
Junos OS flow-based routing functionality	Yes
LACP support for Layer 3	Yes
Layer 2 Ethernet switching capability	Yes
Layer 2 transparent mode LAG	Yes
Layer 3 LAG	Yes
Local interface support (non-reth)	Yes
Low-Impact ISSU	No

Table 4: Chassis Cluster Feature Support on Branch SRX Series Devices (*continued*)

Features	Branch SRX Series
Multicast in HA mode	Yes
Network Time Protocol (NTP) time synchronization in chassis cluster	Yes
Point-to-Point Protocol over Ethernet (PPPoE) over redundant Ethernet interface	Yes
Quality of service (QoS)	SRX550M
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
Redundant Ethernet interfaces	Yes
SPU monitoring	No
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 23](#)

## Chassis Cluster Limitations

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

The SRX Series devices have the following chassis cluster limitations:

### Chassis Cluster

- Group VPN is not supported.
- Unified ISSU is not supported.
- VRRP is not supported.
- Starting with Junos OS Release 12.1X45-D10 and later, sampling features such as flow monitoring, packet capture, and port mirroring are supported on reth interfaces.
  - On all SRX Series devices in a chassis cluster, flow monitoring for version 5 and version 8 is supported. However, flow monitoring for version 9 is not supported.

### 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.



**NOTE:** Flowd monitoring is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

---

### Installation and Upgrade

- For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the **reboot** parameter is not available, because the devices in a cluster are automatically rebooted following an in-band cluster upgrade (ICU).

### Interfaces

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

**Layer 2 Switching**

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

**MIBs**

- The Chassis Cluster MIB is not supported.

**Monitoring**

- The maximum number of monitoring IPs that can be configured per cluster is 64 for the branch SRX Series devices.
- On SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices, logs cannot be sent to NSM when logging is configured in the stream mode. Logs cannot be sent because the security log does not support configuration of the source IP address for the fxp0 interface and the security log destination in stream mode cannot be routed through the fxp0 interface. This implies that you cannot configure the security log server in the same subnet as the fxp0 interface and route the log server through the fxp0 interface.

**Related  
Documentation**

- [Preparing Your Equipment for Chassis Cluster Formation on page 33](#)



## CHAPTER 2

# Understanding Chassis Cluster License Requirements

- [Understanding Chassis Cluster Licensing Requirements on page 27](#)
- [Installing Licenses on the Devices in a Chassis Cluster on page 28](#)
- [Verifying Licenses for an SRX Series Device in a Chassis Cluster on page 30](#)

## 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 28](#)
- [Verifying Licenses for an SRX Series Device in a Chassis Cluster on page 30](#)

## 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” on page 49](#) or [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#).
- 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.



## 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:

```

Feature name	Licenses used	Licenses installed	Licenses needed	Expiry
logical-system	1	26	0	permanent
services-offload	0	1	0	permanent

```

License usage:
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:

```

Feature name	Licenses used	Licenses installed	Licenses needed	Expiry
idp-sig	0	1	0	permanent
logical-system	1	26	0	permanent
services-offload	0	1	0	permanent

```

License usage:
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 28](#)
  - [Understanding Chassis Cluster Licensing Requirements on page 27](#)

## CHAPTER 3

# Planning Your Chassis Cluster Configuration

- [Preparing Your Equipment for Chassis Cluster Formation on page 33](#)
- [SRX Series Chassis Cluster Configuration Overview on page 34](#)

## Preparing Your Equipment for Chassis Cluster Formation

---

**Supported Platforms** [SRX1500](#), [SRX300](#), [SRX320](#), [SRX340](#), [SRX345](#), [SRX550M](#), [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.

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

- SRX300, SRX320, SRX340, SRX345, and SRX550M: Although the devices must be of the same type, they can contain different Physical Interface Modules (PIMs).

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.

The following message is displayed when you try to set a cluster ID greater than 15, and when fabric and control link interfaces are not connected back-to-back or are not connected on 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](#)
- [Understanding Chassis Cluster Fabric Interfaces on page 55](#)

## SRX Series Chassis Cluster Configuration Overview

---

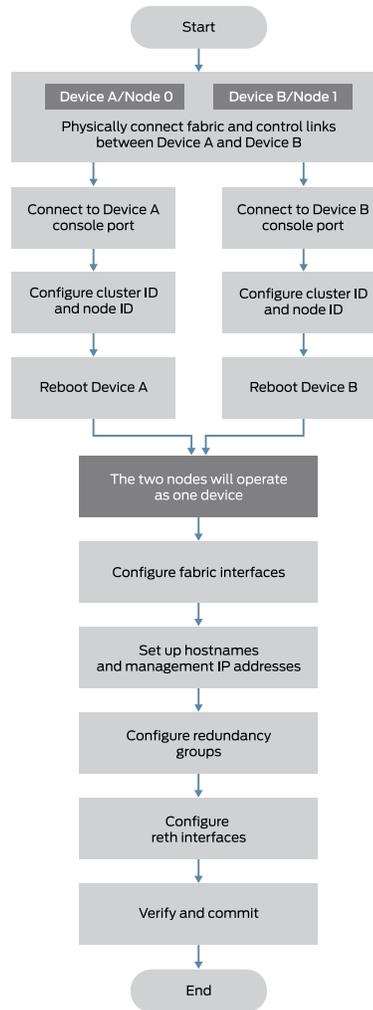
**Supported Platforms** [SRX1500](#), [SRX300](#), [SRX320](#), [SRX340](#), [SRX345](#), [SRX550M](#), [vSRX](#)

[Figure 1 on page 35](#) shows a chassis cluster flow diagram.

Note the following prerequisites for configuring a chassis cluster:

- On SRX Series branch devices, any existing configurations associated with interfaces that transform to the fxp0 management port and the control port should be removed. For more information, see [“Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming” on page 45](#).
- 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 SRX300, SRX320, SRX340, SRX345, and SRX550M chassis clusters, the placement and type of GPIMs, XGPIMs, XPIMs, and Mini-PIMs (as applicable) 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. For more information, see [“Connecting SRX Series Devices to Create a Chassis Cluster” on page 41](#).

- a. Create the fabric link between two nodes in a cluster by connecting any pair of Ethernet interfaces. For most SRX Series devices, the only requirement is that both interfaces be Gigabit Ethernet interfaces (or 10-Gigabit Ethernet interfaces). For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, connect a pair of Gigabit Ethernet interfaces. For SRX1500 devices, fabric child must be of a similar type.

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 149.

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. 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 for Branch SRX Series Devices”](#) on page 49.

4. 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.

5. Configure the management interfaces on the cluster. See [“Example: Configuring the Chassis Cluster Management Interface”](#) on page 51.

6. Configure the cluster with the CLI. See:

- a. [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster](#) on page 82
- b. [Example: Configuring the Chassis Cluster Fabric Interfaces](#) on page 59
- c. [Example: Configuring Chassis Cluster Redundancy Groups](#) on page 71
- d. [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses](#) on page 77
- e. [Example: Configuring Chassis Cluster Interface Monitoring](#) on page 104

7. Initiate manual failover. See [“Initiating a Chassis Cluster Manual Redundancy Group Failover”](#) on page 144.

8. Configure conditional route advertisement over redundant Ethernet interfaces. See [“Understanding Conditional Route Advertising in a Chassis Cluster”](#) on page 157.

9. Verify the configuration. See [“Verifying a Chassis Cluster Configuration”](#) on page 97.

**Related  
Documentation**

- [Connecting SRX Series Devices to Create a Chassis Cluster on page 41](#)
- [Example: Setting the Chassis Cluster Node ID and Cluster ID for Branch SRX Series Devices on page 49](#)
- [Example: Configuring the Chassis Cluster Management Interface on page 51](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 82](#)
- [Verifying a Chassis Cluster Configuration on page 97](#)



## PART 2

# Setting Up Chassis Cluster in SRX Series Devices

- [Chassis Cluster Physical Setup on page 41](#)
- [Setting Up Chassis Cluster Identification on page 45](#)
- [Setting Up Chassis Cluster Management Interfaces on page 51](#)
- [Setting Up Fabric Interfaces on a Chassis Cluster on page 55](#)
- [Setting Up Control Plane Interfaces on a Chassis Cluster on page 63](#)
- [Setting Up Chassis Cluster Redundancy Groups on page 67](#)
- [Setting Up Chassis Cluster Redundant Ethernet Interfaces on page 75](#)
- [Configuring Chassis Cluster on page 85](#)



# Chassis Cluster Physical Setup

- [Connecting SRX Series Devices to Create a Chassis Cluster on page 41](#)

## Connecting SRX Series Devices to Create a Chassis Cluster

---

**Supported Platforms** [SRX1500](#), [SRX300](#), [SRX320](#), [SRX340](#), [SRX345](#), [SRX550M](#)

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.

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

You must use the following ports to form the control link on the branch SRX Series devices:

- For SRX300 devices, connect the ge-0/0/1 on node 0 to the ge-1/0/1 on node 1.
- For SRX320 devices, connect the ge-0/0/1 on node 0 to the ge-3/0/1 on node 1.
- For SRX340 and SRX345 devices, connect the ge-0/0/1 on node 0 to the ge-5/0/1 on node 1.
- For SRX550M devices, connect the ge-0/0/1 on node 0 to the ge-9/0/1 on node 1.
- SRX1500 devices use dedicated control ports.

The fabric link connection must be any pair of either Gigabit Ethernet or 10-Gigabit Ethernet interfaces on all SRX Series devices.

To establish a fabric link:

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

[Figure 2 on page 42](#), [Figure 3 on page 42](#), [Figure 4 on page 42](#), [Figure 5 on page 42](#), [Figure 6 on page 42](#), and [Figure 7 on page 43](#) show pairs of SRX Series devices with the fabric links and control links connected.

Figure 2: Connecting SRX Series Devices in a Cluster (SRX300 Devices)

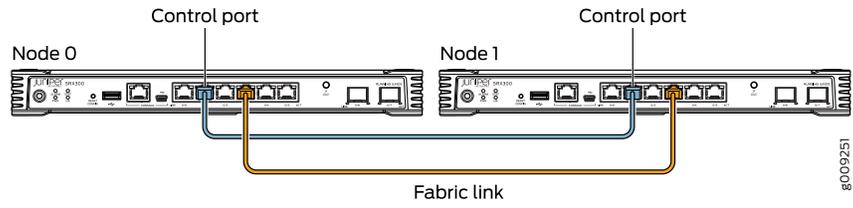


Figure 3: Connecting SRX Series Devices in a Cluster (SRX320 Devices)

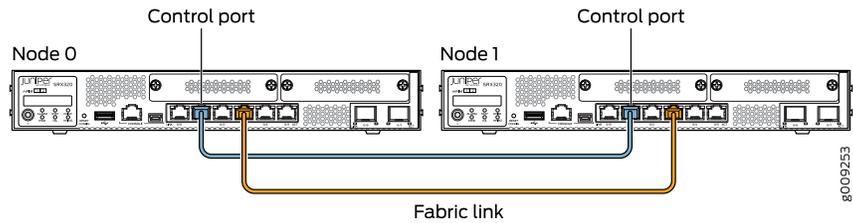


Figure 4: Connecting SRX Series Devices in a Cluster (SRX340 Devices)

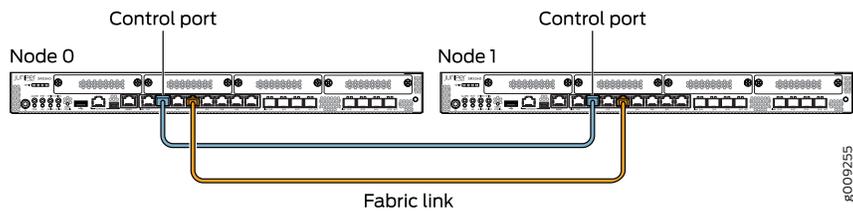


Figure 5: Connecting SRX Series Devices in a Cluster (SRX345 Devices)

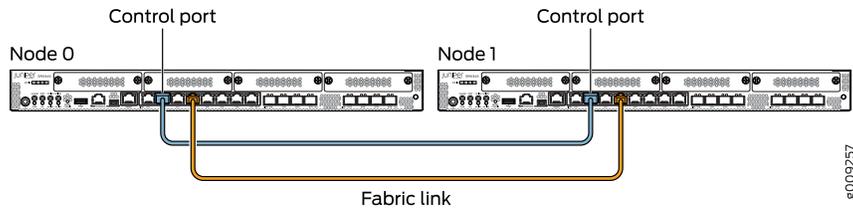


Figure 6: Connecting SRX Series Devices in a Cluster (SRX550M Devices)

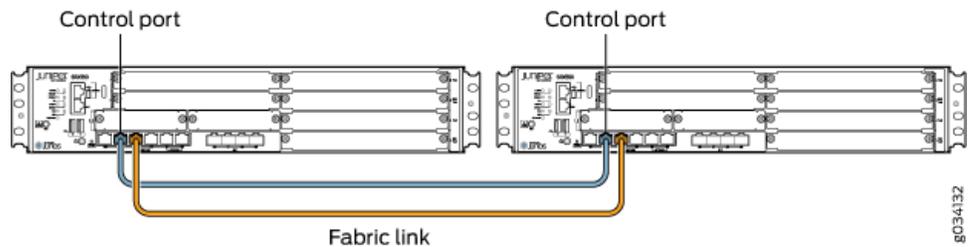
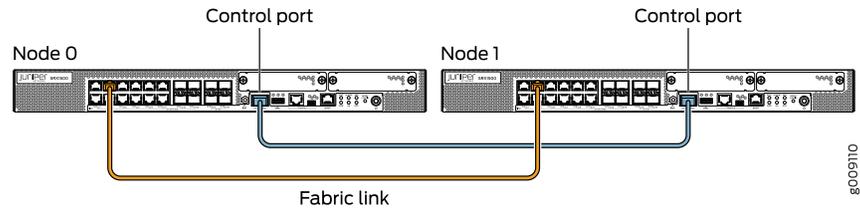


Figure 7: Connecting SRX Series Devices in a Cluster (SRX1500 Devices)



**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 34](#)
- [Example: Setting the Chassis Cluster Node ID and Cluster ID for Branch SRX Series Devices on page 49](#)
- [Example: Configuring the Chassis Cluster Management Interface on page 51](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 82](#)



# Setting Up Chassis Cluster Identification

- Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 45
- Example: Setting the Chassis Cluster Node ID and Cluster ID for Branch SRX Series Devices on page 49

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

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

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

**Table 5: SRX Series Built-in Interfaces**

Devices	Built-In Interfaces				
For SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Devices	ge-0/0/0	ge-0/0/1	ge-0/0/2	ge-0/0/3	...

SRX1500 devices have 16 GE interfaces and 4 XE ports.

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

For SRX300, SRX320, SRX340, and SRX345 devices, after you enable chassis clustering and reboot the system, the built-in interface named `ge-0/0/1` is repurposed as the control interface and is automatically renamed `fxp1`.

For SRX550M devices, control interfaces are dedicated Gigabit Ethernet ports.

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:

$$\text{cluster slot number} = (\text{node ID} * \text{maximum slots per node}) + \text{local slot number}$$

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.

[Table 6 on page 46](#) 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	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
1500	Node 0	1	0	fxp0	Dedicated Control port	Any Ethernet port
					em0	fab0
	Node 1	7	fxp0	Dedicated Control port	Any Ethernet port	
				em0	fab1	
550	Node 0	9 (PIM slots)	0 – 8	ge-0/0/0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1	9 – 17	ge-9/0/0	ge-9/0/1	Any Ethernet port	
				fxp0	fxp1	fab1
340 and 345	Node 0	5 (PIM slots)	0 – 4	fxp0	ge-0/0/1	Any Ethernet port
				fxp0	fxp1	fab0
	Node 1	5 – 9	fxp0	ge-5/0/1	Any Ethernet port	
				fxp0	fxp1	fab1

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

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

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 8 on page 47](#), [Figure 9 on page 48](#), [Figure 10 on page 48](#), [Figure 10 on page 48](#), [Figure 11 on page 48](#), [Figure 12 on page 48](#), and [Figure 13 on page 48](#).)

**Figure 8: Slot Numbering in an SRX Series Chassis Cluster (SRX300 Devices)**

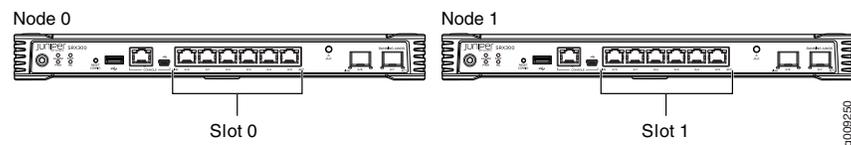


Figure 9: Slot Numbering in an SRX Series Chassis Cluster (SRX320 Devices)

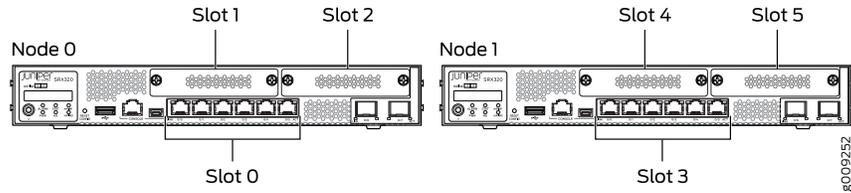


Figure 10: Slot Numbering in an SRX Series Chassis Cluster (SRX340 Devices)

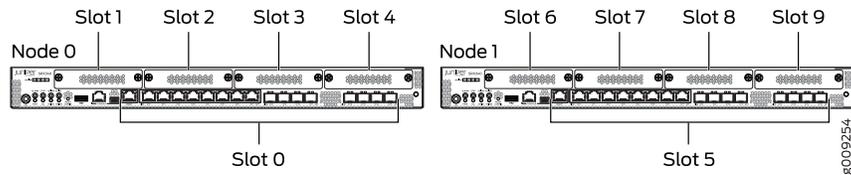


Figure 11: Slot Numbering in an SRX Series Chassis Cluster (SRX345 Devices)

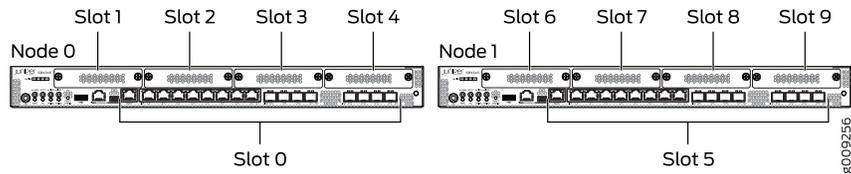


Figure 12: Slot Numbering in an SRX Series Chassis Cluster (SRX550M Devices)

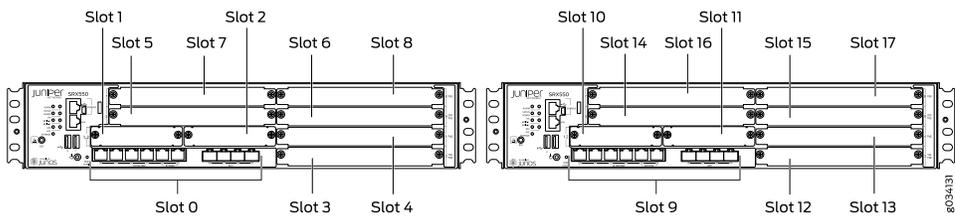
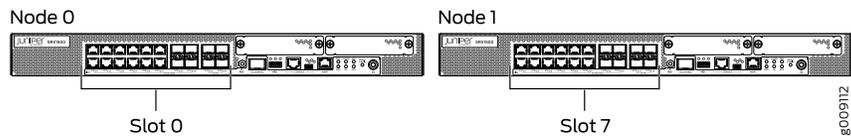


Figure 13: Slot Numbering in an SRX Series Chassis Cluster (SRX1500 Devices)



**Related Documentation**

- [Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster on page 85](#)

## Example: Setting the Chassis Cluster Node ID and Cluster ID for Branch SRX Series Devices

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, 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 on page 49](#)
- [Overview on page 49](#)
- [Configuration on page 49](#)
- [Verification on page 50](#)

### 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 179](#).

### 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.

## 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**
- [SRX Series Chassis Cluster Configuration Overview on page 34](#)
  - [Example: Configuring the Chassis Cluster Management Interface on page 51](#)
  - [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 82](#)

## CHAPTER 6

# Setting Up Chassis Cluster Management Interfaces

- [Management Interface on an Active Chassis Cluster on page 51](#)
- [Example: Configuring the Chassis Cluster Management Interface on page 51](#)

## Management Interface on an Active Chassis Cluster

---

**Supported Platforms** [SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX](#)

The fxp0 interfaces function like standard management interfaces on SRX Series devices and allow network access to each node in the cluster.

Most of SRX Series devices contain an fxp0 interface.

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

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

**Related Documentation**

- [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 45](#)

## 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 on page 52](#)
- [Overview on page 52](#)
- [Configuration on page 52](#)
- [Verification on page 54](#)

## 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”](#) on page 49 or [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#).

## 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

- |                              |  |
|------------------------------|--|
| <b>Purpose</b>               | Verify the chassis cluster management interface configuration.   |
| <b>Action</b>                | To verify the configuration is working properly, enter the <b>show config</b> command.   |
| <b>Related Documentation</b> | <ul style="list-style-type: none"> <li>• <a href="#">Management Interface on an Active Chassis Cluster for Branch SRX Series Devices on page 51</a></li> </ul> |

## CHAPTER 7

# Setting Up Fabric Interfaces on a Chassis Cluster

- [Understanding Chassis Cluster Fabric Interfaces on page 55](#)
- [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)
- [Verifying Chassis Cluster Data Plane Interfaces on page 61](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 61](#)
- [Clearing Chassis Cluster Data Plane Statistics on page 62](#)

## Understanding Chassis Cluster Fabric Interfaces

---

**Supported Platforms** [SRX1500](#), [SRX300](#), [SRX320](#), [SRX340](#), [SRX345](#), [SRX550M](#), [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 on page 56](#)
- [Understanding Session RTOs on page 57](#)
- [Understanding Data Forwarding on page 57](#)
- [Understanding Fabric Data Link Failure and Recovery on page 57](#)

## 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 (RGO) secondary node to move to a disabled state. (Resetting a device to the factory default configuration removes the fabric configuration and thereby causes the RGO secondary node to move to a disabled state.) After the fabric configuration is committed, do not reset either device to the factory default configuration.

---

For SRX1500, the fabric link can be any pair of Ethernet interfaces spanning the cluster; the fabric link can be any pair of Gigabit Ethernet interface or any pair of 10-Gigabit Ethernet interface. For SRX300, SRX320, SRX340, and SRX345 devices, the fabric link can be any pair of Gigabit Ethernet interfaces.

For SRX Series chassis clusters made up of SRX550M devices, SFP interfaces on Mini-PIMs cannot be used as the fabric link.

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 45.

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 should start exchanging probes. As soon as this happens, all the FPCs on the previously ineligible node will be reset. They then come to online state and rejoin the cluster.



**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.

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

- [Understanding Chassis Cluster Dual Fabric Links on page 149](#)
- [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)

- [Verifying Chassis Cluster Data Plane Interfaces on page 61](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 61](#)
- [Clearing Chassis Cluster Data Plane Statistics on page 62](#)
- [Preparing Your Equipment for Chassis Cluster Formation on page 33](#)

## 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 on page 59](#)
- [Overview on page 59](#)
- [Configuration on page 60](#)
- [Verification on page 60](#)

### 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*.

### 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](#)
  - [Verifying Chassis Cluster Data Plane Interfaces on page 61](#)

## 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 on page 55](#)
  - [Understanding Chassis Cluster Fabric Interfaces for High-End SRX Series](#)
  - [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)
  - [Verifying Chassis Cluster Data Plane Statistics on page 61](#)
  - [Clearing Chassis Cluster Data Plane Statistics on page 62](#)

## 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 on page 55](#)
  - [Understanding Chassis Cluster Fabric Interfaces for High-End SRX Series](#)
  - [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)
  - [Verifying Chassis Cluster Data Plane Interfaces on page 61](#)
  - [Clearing Chassis Cluster Data Plane Statistics on page 62](#)

## 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 on page 55](#)
  - [Understanding Chassis Cluster Fabric Interfaces for High-End SRX Series](#)
  - [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)
  - [Verifying Chassis Cluster Data Plane Statistics on page 61](#)
  - [Verifying Chassis Cluster Data Plane Interfaces on page 61](#)

## CHAPTER 8

# Setting Up Control Plane Interfaces on a Chassis Cluster

- [Understanding Chassis Cluster Control Plane and Control Links on page 63](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 64](#)
- [Clearing Chassis Cluster Control Plane Statistics on page 65](#)

## Understanding Chassis Cluster Control Plane and Control Links

---

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, 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 should 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 should 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 should be connected to control port 0 on node 1 and control port 1 should 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 liveness signals across the nodes.

SRX1500 devices use the dedicated control port.

For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the control link uses the ge-0/0/1 interface.

For details about port and interface usage for management, control, and fabric links, see [Table 6 on page 46](#).

### Related Documentation

- [Verifying Chassis Cluster Control Plane Statistics on page 64](#)
- [Clearing Chassis Cluster Control Plane Statistics on page 65](#)

## 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 65](#)

## 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 64](#)



# Setting Up Chassis Cluster Redundancy Groups

- [Understanding Chassis Cluster Redundancy Groups on page 67](#)
- [Example: Configuring Chassis Cluster Redundancy Groups on page 71](#)

## 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 preemption, see [preempt \(Chassis Cluster\)](#).

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



**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 71.)

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 [Table 8 on page 76](#)). 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 7 on page 70](#) 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 7: 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	<b>ge-1/0/0</b>	<b>ge-5/0/0</b>
		Node 1: 2	Redundant Ethernet interface 1	<b>ge-1/3/0</b>	<b>ge-5/3/0</b>
Redundancy group 2	Node 1	Node 0: 2	Redundant Ethernet interface 2	<b>ge-2/0/0</b>	<b>ge-6/0/0</b>
		Node 1: 254	Redundant Ethernet interface 3	<b>ge-2/3/0</b>	<b>ge-6/3/0</b>
Redundancy group 3	Node 0	Node 0: 254	Redundant Ethernet interface 4	<b>ge-3/0/0</b>	<b>ge-7/0/0</b>
		Node 1: 2	Redundant Ethernet interface 5	<b>ge-3/3/0</b>	<b>ge-7/3/0</b>

As the example for a chassis cluster in [Table 7 on page 70](#) 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 71](#)

## Example: Configuring Chassis Cluster Redundancy Groups

**Supported Platforms** [SRX Series, vSRX](#)

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

- [Requirements on page 71](#)
- [Overview on page 71](#)
- [Configuration on page 72](#)
- [Verification on page 73](#)

### 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” on page 49](#) or [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#).
2. Configure the chassis cluster management interface. See [“Example: Configuring the Chassis Cluster Management Interface” on page 51](#).
3. Configure the chassis cluster fabric. See [“Example: Configuring the Chassis Cluster Fabric Interfaces” on page 59](#).

### 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 67](#)



# Setting Up Chassis Cluster Redundant Ethernet Interfaces

- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 75
- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 77
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 82

## Understanding Chassis Cluster Redundant Ethernet Interfaces

---

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

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



**NOTE:** For SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a chassis cluster deployment is 1024.

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.



**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 8 on page 76](#). Note that

the number of redundant Ethernet interfaces configured determines the number of redundancy groups that can be configured.

**Table 8: Maximum Number of Redundant Ethernet Interfaces Allowed**

Device	Maximum Number of reth Interfaces
SRX300, SRX320, SRX340, SRX345	128
SRX550M	58
SRX1500	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.

Point-to-Point Protocol over Ethernet (PPPoE) over redundant Ethernet (reth) interface is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices in chassis cluster mode. This feature allows an existing PPPoE session to continue without starting a new PPPoE session in the event of a failover.



**NOTE:** On all branch SRX Series devices, the number of child interfaces per node is restricted to eight on the reth interface and the number of child interfaces per reth interface is restricted to eight.



**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 1.1.1/24;
    }
  }
}
```

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

```
interfaces {
  ge-2/0/2 {
    together-options {
      redundant-parent reth2;
    }
  }
  reth2 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet {
        address 1.1.1/24;
      }
    }
  }
}
```

**Related Documentation**

- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 77](#)
- [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 163](#)
- [Understanding Conditional Route Advertising in a Chassis Cluster on page 157](#)
- [Preparing Your Equipment for Chassis Cluster Formation on page 33](#)

## 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 on page 78](#)
- [Overview on page 78](#)
- [Configuration on page 78](#)
- [Verification on page 81](#)

## 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” on page 49](#) or [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#).
- 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 59](#).
- Understand how to set the chassis cluster node redundancy groups. See [“Example: Configuring Chassis Cluster Redundancy Groups” on page 71](#).

## 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 {
  gigeother-options {
    redundant-parent reth1;
  }
}
ge-7/0/0 {
  gigeother-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 81](#)
- [Verifying Chassis Cluster Control Links on page 82](#)

### 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 on page 75](#)

## 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 on page 83](#)
- [Overview on page 83](#)
- [Configuration on page 83](#)
- [Verification on page 83](#)

## 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](#)” on [page 49](#) or [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#).

## 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:

- Specify the number of redundant Ethernet interfaces:
 

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 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 77](#)



## CHAPTER 11

# Configuring Chassis Cluster

- [Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster on page 85](#)
- [Verifying a Chassis Cluster Configuration on page 97](#)
- [Verifying Chassis Cluster Statistics on page 97](#)
- [Clearing Chassis Cluster Statistics on page 99](#)

## Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster

---

**Supported Platforms** [SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX](#)

This example shows how to set up chassis clustering on an SRX Series for the branch device.

- [Requirements on page 85](#)
- [Overview on page 86](#)
- [Configuration on page 87](#)
- [Verification on page 93](#)

### Requirements

Before you begin:

- Physically connect the two devices and ensure that they are the same models. For example, on the SRX1500 Services Gateway, connect the dedicated control ports on node 0 and node 1.



**NOTE:** For SRX300, SRX320, SRX340, and SRX345 devices, connect `ge-0/0/1` on node 0 to `ge-0/0/1` on node 1.

- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:

- On node 0:

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

- On node 1:

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

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

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



**NOTE:**

After clustering occurs,

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



**NOTE:**

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

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

See “[Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming](#)” on page 45 for complete mapping of the SRX Series devices.

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

## Overview

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

The node 1 renumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. See [Table 9 on page 87](#) for interface renumbering on the SRX Series device.

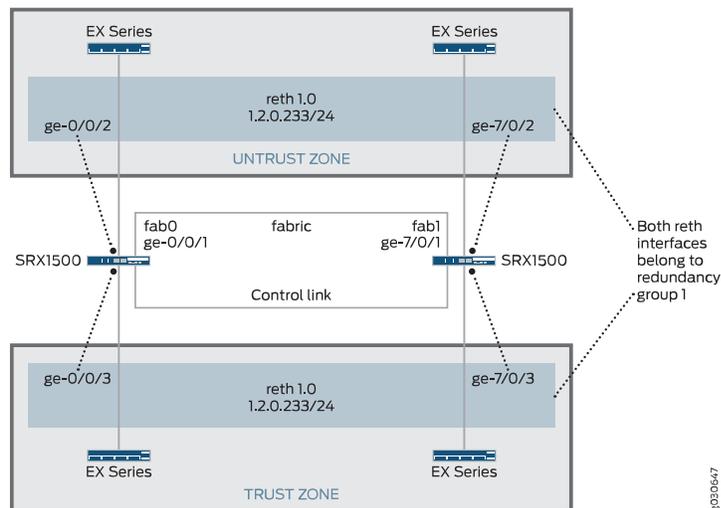
Table 9: SRX Series Services Gateways Interface Renumbering

SRX Series Services Gateway	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
SRX300	1	ge-0/0/0	ge-1/0/0
SRX320	3	ge-0/0/0	ge-3/0/0
SRX340	5	ge-0/0/0	ge-5/0/0
SRX345			
SRX550M	9	ge-0/0/0	ge-9/0/0
SRX1500	7	ge-0/0/0	ge-7/0/0

After clustering is enabled, the system creates fxp0, fxp1, and em0 interfaces. Depending on the device, the fxp0, fxp1, and em0 interfaces that are mapped to a physical interface are not user defined. However, the fab interface is user defined.

Figure 14 on page 87 shows the topology used in this example.

Figure 14: SRX Series for the Branch Topology Example



## Configuration

**CLI Quick Configuration** To quickly configure a chassis cluster on an SRX1500 Services Gateway, copy the following commands and paste them into the CLI:

On {primary:node0}

```
[edit]
set groups node0 system host-name srx1500-1
set groups node0 interfaces fxp0 unit 0 family inet address 192.16.35.46/24
set groups node1 system host-name srx1500-2
```

```

set groups node1 interfaces fxp0 unit 0 family inet address 192.16.35.47/24
set groups node0 system backup-router <backup next-hop from fxp0> destination
  <management network/mask>
set groups node1 system backup-router <backup next-hop from fxp0> destination
  <management network/mask>
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-2/0/1
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-7/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-7/0/2 weight 255
set chassis cluster reth-count 2
set interfaces ge-0/0/2 gigeother-options redundant-parent reth1
set interfaces ge-7/0/2 gigeother-options redundant-parent reth1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 1.2.0.233/24
set interfaces ge-0/0/3 gigeother-options redundant-parent reth0
set interfaces ge-7/0/3 gigeother-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.16.8.1/24
set security zones security-zone Untrust interfaces reth1.0
set security zones security-zone Trust interfaces reth0.0

```

If you are configuring a Branch SRX Series device, see [Table 10 on page 88](#) for command and interface settings for your device and substitute these commands into your CLI.

**Table 10: SRX Series Services Gateways for the Branch Interface Settings**

Command	SRX300	SRX320	SRX340 SRX345	SRX550M
set interfaces fab0 fabric-options member-interfaces	ge-0/0/2	ge-0/0/2	ge-0/0/2	ge-0/0/2
set interfaces fab1 fabric-options member-interfaces	ge-1/0/2	ge-3/0/2	ge-5/0/2	ge-9/0/2
set chassis cluster redundancy-group 1 interface-monitor	ge-0/0/3 weight 255	ge-0/0/3 weight 255	ge-0/0/3 weight 255	ge-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	ge-0/0/4 weight 255	ge-0/0/4 weight 255	ge-0/0/4 weight 255	ge-10/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	ge-1/0/3 weight 255	ge-3/0/3 weight 255	ge-5/0/3 weight 255	ge-1/0/1 weight 255

Table 10: SRX Series Services Gateways for the Branch Interface Settings (*continued*)

Command	SRX300	SRX320	SRX340 SRX345	SRX550M
set chassis cluster redundancy-group 1 interface-monitor	ge-1/0/4 weight 255	ge-3/0/4 weight 255	ge-5/0/4 weight 255	ge-10/0/1 weight 255
set interfaces	ge-0/0/3 gigether-options redundant-parent reth0	ge-0/0/3 gigether-options redundant-parent reth0	ge-0/0/3 gigether-options redundant-parent reth0	ge-1/0/0 gigether-options redundant-parent reth1
set interfaces	ge-0/0/4 gigether-options redundant-parent reth1	ge-0/0/4 gigether-options redundant-parent reth1	ge-0/0/4 gigether-options redundant-parent reth1	ge-10/0/0 gigether-options redundant-parent reth1
set interfaces	ge-1/0/3 gigether-options redundant-parent reth0	ge-3/0/3 gigether-options redundant-parent reth0	ge-5/0/3 gigether-options redundant-parent reth0	ge-1/0/1 gigether-options redundant-parent reth0
set interfaces	ge-1/0/4 gigether-options redundant-parent reth1	ge-3/0/4 gigether-options redundant-parent reth1	ge-5/0/4 gigether-options redundant-parent reth1	ge-10/0/1 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 the CLI User Guide*.

To configure a chassis cluster on an SRX Series for the branch device:



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

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

```

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

```

Set the default route and backup router for each node.

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

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

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

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

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

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

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

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



**NOTE:** We do not recommend Interface monitoring for redundancy group 0 because it causes the control plane to switch from one node to another node in case interface flap occurs.

```
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/3
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/3
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/2
weight 255
```



**NOTE:** Interface failover only occurs after the weight reaches 0.

5. Set up the redundant Ethernet (reth) interfaces and assign the redundant interface to a zone.

```
user@host# set chassis cluster reth-count 2
user@host# set interfaces ge-0/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-7/0/2 gigether-options redundant-parent reth1
```

```

user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 1.2.0.233/24
user@host# set interfaces ge-0/0/3 gigether-options redundant-parent reth0
user@host# set interfaces ge-7/0/3 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 10.16.8.1/24
user@host# set security zones security-zone Untrust interfaces reth1.0
user@host# set security zones security-zone Trust interfaces reth0.0

```

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

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

```

> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX1500-1;
      backup-router 10.100.22.1 destination 66.129.243.0/24;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.16.35.46/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name SRX1500-2;
      backup-router 10.100.21.1 destination 66.129.243.0/24;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 192.16.35.47/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
chassis {
  cluster {
    reth-count 2;
    redundancy-group 0 {
      node 0 priority 100;
      node 1 priority 1;
    }
    redundancy-group 1 {

```

```
        node 0 priority 100;
        node 1 priority 1;
        interface-monitor {
            ge-0/0/3 weight 255;
            ge-0/0/2 weight 255;
            ge-7/0/2 weight 255;
            ge-7/0/3 weight 255;
        }
    }
}
interfaces {
    ge-0/0/2 {
        gigger-options {
            redundant-parent reth1;
        }
        unit 0 {
            family inet {
                address 2.2.2.2/30;
            }
        }
    }
    ge-0/0/3 {
        gigger-options {
            redundant-parent reth0;
        }
    }
    ge-7/0/2 {
        gigger-options {
            redundant-parent reth1;
        }
    }
    ge-7/0/3 {
        gigger-options {
            redundant-parent reth0;
        }
    }
    fab0 {
        fabric-options {
            member-interfaces {
                ge-0/0/1;
            }
        }
    }
    fab1 {
        fabric-options {
            member-interfaces {
                ge-2/0/1;
            }
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.16.8.1/24;
            }
        }
    }
}
```

```

    reth1 {
      redundant-ether-options {
        redundancy-group 1;
      }
      unit 0 {
        family inet {
          address 1.2.0.233/24;
        }
      }
    }
  }
  ...
security {
  zones {
    security-zone Untrust {
      interfaces {
        reth1.0;
      }
    }
    security-zone Trust {
      interfaces {
        reth0.0;
      }
    }
  }
  policies {
    from-zone Trust to-zone Untrust {
      policy 1 {
        match {
          source-address any;
          destination-address any;
          application any;
        }
        then {
          permit;
        }
      }
    }
  }
}

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying Chassis Cluster Status on page 94](#)
- [Verifying Chassis Cluster Interfaces on page 94](#)
- [Verifying Chassis Cluster Statistics on page 94](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 95](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 95](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 96](#)
- [Troubleshooting with Logs on page 96](#)

### Verifying Chassis Cluster Status

**Purpose** Verify the chassis cluster status, failover status, and redundancy group information.

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

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

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

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

### Verifying Chassis Cluster Interfaces

**Purpose** Verify information about chassis cluster interfaces.

**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: em0

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

Interface Monitoring:
  Interface    Weight    Status    Redundancy-group
  ge-7/0/3     255      Up        1
  ge-7/0/2     255      Up        1
  ge-0/0/2     255      Up        1
  ge-0/0/3     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: 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
  PTP ALG               0          0
  RPC ALG               0          0
  RTSP ALG              0          0
  RAS ALG               0          0
  MAC address learning  0          0
  GPRS GTP              0          0

```

### Verifying Chassis Cluster Control Plane Statistics

- Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).
- Action** From operational mode, enter the `show chassis cluster control-plane statistics` command.
- ```

{primary:node0}
user@host> show chassis cluster control-plane statistics

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 2294
    Heartbeat packets received: 2298
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 2290
    Probes received: 615

```

### Verifying Chassis Cluster Data Plane Statistics

- Purpose** Verify information about the number of RTOs sent and received for services.
- Action** From operational mode, enter the `show chassis cluster data-plane statistics` command.
- ```

{primary:node0}
user@host> show chassis cluster data-plane statistics

Services Synchronized:

```

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### Verifying Chassis Cluster Redundancy Group Status

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the **chassis cluster status redundancy-group** command.

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

Redundancy group: 1, Failover count: 1
node0           100     primary no       no
node1           50      secondary no       no
```

### Troubleshooting with Logs

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these **show log** commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

- Related Documentation**
- [Understanding Chassis Cluster Redundancy Groups on page 67.](#)
  - [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 45](#)

## 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 97](#)
  - [Clearing Chassis Cluster Statistics on page 99](#)

## 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 97](#)
  - [Clearing Chassis Cluster Statistics on page 99](#)

---

## 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 97](#)
  - [Verifying Chassis Cluster Statistics on page 97](#)



## PART 3

# Managing Chassis Cluster Operations

- [Monitoring Chassis Cluster on page 103](#)
- [Managing Chassis Cluster Redundancy Group Failover on page 139](#)
- [Configuring Chassis Cluster Dual Fabric Links to Increase Redundancy and Performance on page 149](#)
- [Configuring Route Advertisement over Redundant Ethernet Interfaces in a Chassis Cluster on page 157](#)
- [Configuring Redundant Ethernet LAG Interfaces for Increasing High Availability and Overall Throughput on page 163](#)
- [Simplifying Chassis Cluster Management on page 179](#)



# Monitoring Chassis Cluster

- [Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 103](#)
- [Example: Configuring Chassis Cluster Interface Monitoring on page 104](#)
- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 131](#)
- [Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 134](#)

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

#### Related Documentation

- [Example: Configuring Chassis Cluster Interface Monitoring on page 104](#)
- [Understanding Chassis Cluster Redundancy Groups on page 67](#)
- [Example: Configuring Chassis Cluster Redundancy Groups on page 71](#)

## 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 104](#)
- [Overview on page 105](#)
- [Configuration on page 106](#)
- [Verification on page 110](#)

### Requirements

Before you begin, create a redundancy group. See [“Example: Configuring Chassis Cluster Redundancy Groups” on page 71](#).

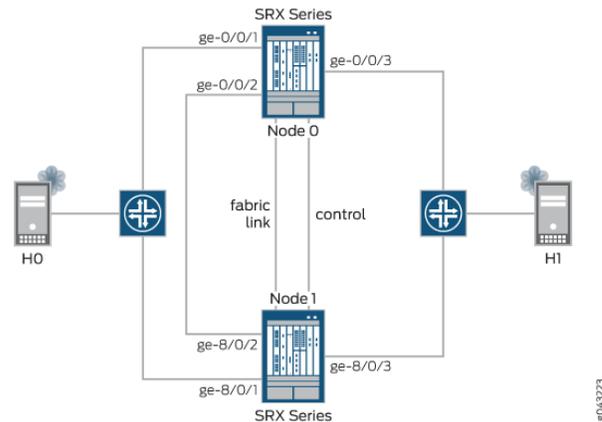
## Overview

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 106 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 Status on page 111](#)
- [Verifying Chassis Cluster Interfaces on page 111](#)
- [Verifying Chassis Cluster Information on page 112](#)
- [Verifying Interface ge-0/0/1 Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 on page 113](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 on page 114](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 on page 114](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 on page 115](#)
- [Verifying Interface ge-0/0/2 Is Disabled on page 117](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/2 on page 117](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/2 on page 118](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/2 on page 119](#)
- [Verifying Interface Status After Disabling ge-0/0/3 on page 120](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/3 on page 121](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/3 on page 121](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/3 on page 122](#)
- [Verifying That Interface ge-0/0/2 Is Enabled on page 124](#)
- [Verifying Chassis Cluster Status After Enabling Interface ge-0/0/2 on page 124](#)
- [Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/2 on page 125](#)
- [Verifying Chassis Cluster Information After Enabling Interface ge-0/0/2 on page 125](#)
- [Verifying Chassis Cluster RG2 Preempt on page 127](#)
- [Verifying Chassis Cluster Status After Preempting RG2 on page 127](#)
- [Verifying That Interface ge-0/0/3 Is Enabled on page 128](#)
- [Verifying Chassis Cluster Status After Enabling Interface ge-0/0/3 on page 128](#)

- [Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/3 on page 129](#)
- [Verifying Chassis Cluster Information After Enabling Interface ge-0/0/3 on page 130](#)

### 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;
gather-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    ge-0/0/0         Up / Up
  fab1    ge-8/0/0         Up / Up
  fab1    ge-8/0/0         Up / Up

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 RGI 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;
gether-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 71](#)
  - [Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 103](#)
  - [Understanding Chassis Cluster Redundancy Group IP Address Monitoring for Branch SRX Series Devices on page 131](#)
  - [Understanding Chassis Cluster Redundancy Group IP Address Monitoring for High-End SRX Series Devices](#)
  - [Understanding Chassis Cluster Redundancy Group Failover on page 139](#)
  - [Understanding Chassis Cluster Redundancy Groups on page 67](#)
  - [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for Branch SRX Series Devices on page 45](#)
  - [Understanding SRX Series Chassis Cluster Slot Numbering, Physical Port and Logical Interface Naming for High-End SRX Series Devices](#)

## Understanding Chassis Cluster Redundancy Group IP Address Monitoring

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, 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 should be monitored are router gateway addresses to ensure that valid traffic coming into the services gateway can be forwarded to the appropriate network router.



**NOTE:** Starting in Junos OS Release 12.1X46-D35, for all SRX Series devices, the *reth* interface supports proxy ARP.

---

One Services Processing Unit (SPU) or Packet Forwarding Engine (PFE) per node is designated to send Internet Control Message Protocol (ICMP) ping packets for the monitored IP addresses on the cluster. The primary PFE sends ping packets using Address Resolution Protocol (ARP) requests resolved by the Routing Engine (RE). The source for these pings is the redundant Ethernet interface MAC and IP addresses. The secondary PFE resolves ARP requests for the monitored IP address itself. The source for these pings is the physical child MAC address and a secondary IP address configured on the redundant Ethernet interface. For the ping reply to be received on the secondary interface, the I/O card (IOC), central PFE processor, or Flex IOC adds both the physical child MAC address and the redundant Ethernet interface MAC address to its MAC table. The secondary PFE responds with the physical child MAC address to ARP requests sent to the secondary IP address configured on the redundant Ethernet interface.

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.

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.



**NOTE:** Upstream device failure detection for the chassis cluster feature is supported on SRX300, SRX320, SRX340, SRX345, and SRX1500 devices.

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.



**NOTE:** Redundancy group IP address monitoring is not supported for IPv6 destinations.

#### Related Documentation

- [Understanding Chassis Cluster Redundancy Groups on page 67](#)
- [Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 103](#)

- [Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 134](#)
- [Understanding Chassis Cluster Redundancy Group Failover on page 139](#)

## 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 on page 134](#)
- [Overview on page 134](#)
- [Configuration on page 135](#)
- [Verification on page 136](#)

### 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 on page 131](#)
  - [Understanding Chassis Cluster Redundancy Group IP Address Monitoring for High-End SRX Series Devices](#)
  - [Understanding Chassis Cluster Redundancy Group Failover on page 139](#)



# Managing Chassis Cluster Redundancy Group Failover

- [Understanding Chassis Cluster Redundancy Group Failover on page 139](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 140](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover on page 141](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 143](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 144](#)
- [Verifying Chassis Cluster Failover Status on page 146](#)
- [Clearing Chassis Cluster Failover Status on page 147](#)

## 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 71](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover on page 141](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 143](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 144](#)

---

## 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 140](#)
- [Overview on page 141](#)
- [Configuration on page 141](#)

### Requirements

Before you begin:

- Understand redundancy group failover. See “[Understanding Chassis Cluster Redundancy Group Failover](#)” on page 139.
- Understand redundancy group manual failover. See “[Understanding Chassis Cluster Redundancy Group Manual Failover](#)” on page 141.

## 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 67](#)
  - [Example: Configuring Chassis Cluster Redundancy Groups on page 71](#)
  - [Understanding Chassis Cluster Redundancy Group Manual Failover on page 141](#)
  - [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 143](#)
  - [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 144](#)

## 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 preemption, see [preempt \(Chassis Cluster\)](#).

When you do a manual failover for redundancy group 0, the node in the primary state transitions to the secondary-hold state. The node stays in the secondary-hold state for the default or configured time (a minimum of 300 seconds) and then transitions to the secondary state.

State transitions in cases where one node is in the secondary-hold state and the other node reboots, or the control link connection or fabric link connection is lost to that node, are described as follows:

- Reboot case—The node in the secondary-hold state transitions to the primary state; the other node goes dead (inactive).
- Control link failure case—The node in the secondary-hold state transitions to the ineligible state and then to a disabled state; the other node transitions to the primary state.
- Fabric link failure case—The node in the secondary-hold state transitions directly to the ineligible state.



**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 139](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 144](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 140](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 143](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces for Branch SRX Series Devices on page 75](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces for High-End SRX Series Devices](#)

## 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 141](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 144](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 140](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces for Branch SRX Series Devices on page 75](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces for High-End SRX Series Devices](#)

## 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 71](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 77](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 140](#)



**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 141](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 140](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 143](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces for Branch SRX Series Devices on page 75](#)
- [Understanding Chassis Cluster Redundant Ethernet Interfaces for High-End SRX Series Devices](#)

## 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 144](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 82](#)
- [Verifying a Chassis Cluster Configuration on page 97](#)
- [Verifying Chassis Cluster Statistics on page 97](#)
- [Clearing Chassis Cluster Failover Status on page 147](#)

## 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 144](#)

- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster on page 82](#)
- [Verifying a Chassis Cluster Configuration on page 97](#)
- [Verifying Chassis Cluster Statistics on page 97](#)
- [Verifying Chassis Cluster Failover Status on page 146](#)

# Configuring Chassis Cluster Dual Fabric Links to Increase Redundancy and Performance

- [Understanding Chassis Cluster Dual Fabric Links on page 149](#)
- [Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports on page 150](#)
- [Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports on page 152](#)

## 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 should be the same type. For example, both should be Gigabit Ethernet interfaces or 10-Gigabit interfaces.



**NOTE:** SRX300, SRX320, SRX340, and SRX345 devices support Gigabit Ethernet interfaces only.

---

- Related Documentation**
- [Understanding Chassis Cluster Fabric Interfaces on page 55](#)
  - [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)
  - [Verifying Chassis Cluster Data Plane Interfaces on page 61](#)
  - [Verifying Chassis Cluster Data Plane Statistics on page 61](#)
  - [Clearing Chassis Cluster Data Plane Statistics on page 62](#)

---

## 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 on page 150](#)
- [Overview on page 150](#)
- [Configuration on page 151](#)
- [Verification on page 152](#)

### 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” on page 49](#) or [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#).

### 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 on page 149](#)
  - [Understanding Chassis Cluster Dual Fabric Links for High-End SRX Series](#)
  - [Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports on page 152](#)
  - [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)

## 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 on page 153](#)
- [Overview on page 153](#)

- [Configuration on page 153](#)
- [Verification on page 154](#)

## 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](#)” on [page 49](#) or [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#).

## 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 on page 149](#)
  - [Understanding Chassis Cluster Dual Fabric Links for High-End SRX Series](#)
  - [Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports on page 150](#)



# Configuring Route Advertisement over Redundant Ethernet Interfaces in a Chassis Cluster

- [Understanding Conditional Route Advertising in a Chassis Cluster on page 157](#)
- [Example: Configuring Conditional Route Advertising in a Chassis Cluster on page 158](#)

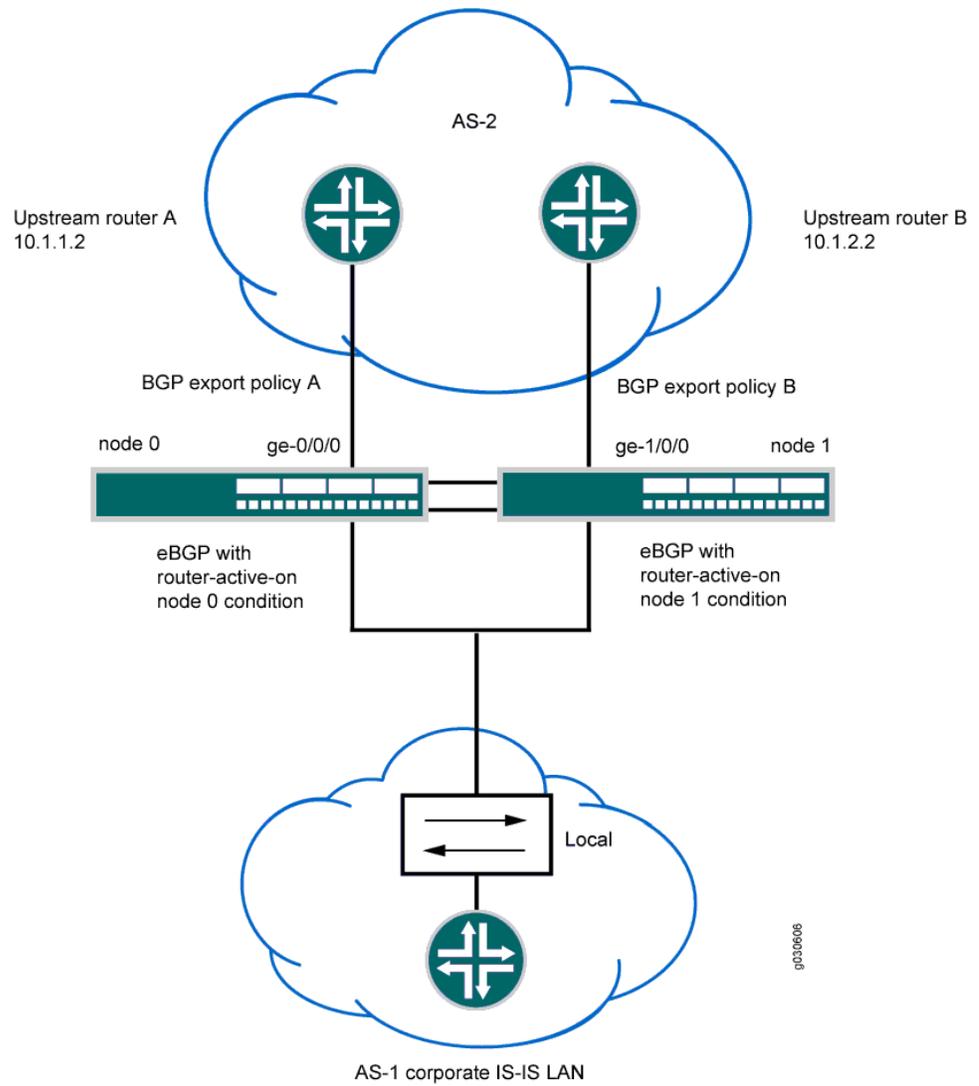
## Understanding Conditional Route Advertising in a Chassis Cluster

**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 16 on page 158](#) shows a typical scenario, with a redundant Ethernet interface connecting the corporate LAN, through a chassis cluster, to an external network segment.

Figure 16: Conditional Route Advertising



**Related Documentation**

- [Example: Configuring Conditional Route Advertising in a Chassis Cluster on page 158](#)
- [Verifying a Chassis Cluster Configuration on page 97](#)
- [Verifying Chassis Cluster Statistics on page 97](#)

**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 159](#)
- [Overview on page 159](#)
- [Configuration on page 161](#)

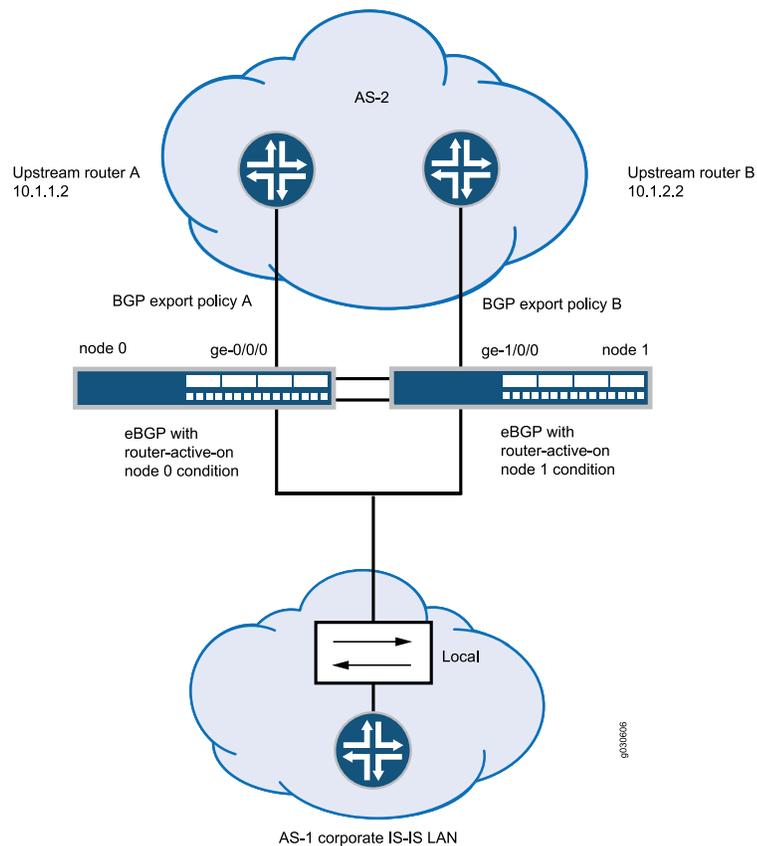
## Requirements

Before you begin, understand conditional route advertising in a chassis cluster. See [“Understanding Conditional Route Advertising in a Chassis Cluster” on page 157](#).

## Overview

As illustrated in [Figure 17 on page 160](#), 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 17: 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 157](#)
  - [Verifying a Chassis Cluster Configuration on page 97](#)
  - [Verifying Chassis Cluster Statistics on page 97](#)

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

- [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 163](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 165](#)
- [Understanding Chassis Cluster Redundant Ethernet Interface LAG Failover on page 168](#)
- [Understanding LACP on Chassis Clusters on page 171](#)
- [Example: Configuring LACP on Chassis Clusters on page 173](#)
- [Example: Configuring Chassis Cluster Minimum Links on page 176](#)

## Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, 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:** On SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the speed mode and link mode configuration is available for member interfaces of a reth 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 on page 75](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 165](#)
- [Example: Configuring Chassis Cluster Minimum Links on page 176](#)
- [Understanding Conditional Route Advertising in a Chassis Cluster on page 157](#)
- [Preparing Your Equipment for Chassis Cluster Formation on page 33](#)

## 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 on page 166](#)
- [Overview on page 166](#)
- [Configuration on page 166](#)
- [Verification on page 168](#)

## Requirements

Before you begin:

- Configure chassis cluster redundant interfaces. See “[Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses](#)” on page 77.
- Understand chassis cluster redundant Ethernet interface link aggregation groups. See “[Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for Branch SRX Series Devices](#)” on page 163 or *Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices*.

## 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 {
    gigger-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 on page 163](#)
  - [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices](#)
  - [Understanding Chassis Cluster Redundant Ethernet Interface LAG Failover on page 168](#)
  - [Understanding LACP on Chassis Clusters on page 171](#)
  - [Example: Configuring LACP on Chassis Clusters on page 173](#)
  - [Example: Configuring Chassis Cluster Minimum Links on page 176](#)

## 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 169](#)
- [Scenario 2: Monitored Interface Weight Is 75 on page 170](#)
- [Scenario 3: Monitored Interface Weight Is 100 on page 170](#)

## 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 on page 163](#)
- [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices](#)

- [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 165](#)
- [Understanding LACP on Chassis Clusters on page 171](#)
- [Example: Configuring LACP on Chassis Clusters on page 173](#)
- [Example: Configuring Chassis Cluster Minimum Links on page 176](#)

## 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 on page 171](#)
- [Sub-LAGs on page 172](#)
- [Supporting Hitless Failover on page 173](#)
- [Managing Link Aggregation Control PDUs on page 173](#)

### 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” on page 163](#) or [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices](#).

### 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 176](#).

## 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 173.

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 173](#)

---

## Example: Configuring LACP on Chassis Clusters

Supported Platforms [SRX Series](#)

This example shows how to configure LACP on chassis clusters.

- [Requirements on page 174](#)
- [Overview on page 174](#)
- [Configuration on page 174](#)
- [Verification on page 175](#)

## 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 171](#)
- [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 on page 176](#)
- [Overview on page 176](#)
- [Configuration on page 176](#)
- [Verification on page 177](#)

### Requirements

Before you begin:

- Configure redundant Ethernet interfaces. See “[Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses](#)” on page 77.
- Understand redundant Ethernet interface link aggregation groups. See “[Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups](#)” on page 165.

### 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
```

- 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
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 on page 163](#)
  - [Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups for High-End SRX Series Devices](#)
  - [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 165](#)
  - [Understanding Conditional Route Advertising in a Chassis Cluster on page 157](#)



# Simplifying Chassis Cluster Management

- [Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 179](#)
- [Verifying Chassis Cluster Configuration Synchronization Status on page 180](#)
- [NTP Time Synchronization on SRX Series Devices on page 181](#)
- [Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 181](#)

## 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 reenabling 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 180](#)
- [NTP Time Synchronization on SRX Series Devices on page 181](#)
- [Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 181](#)

## 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 179](#)
- [NTP Time Synchronization on SRX Series Devices on page 181](#)
- [Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 181](#)
- [show chassis cluster information configuration-synchronization on page 367](#)

## 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 181](#)

## 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 on page 182](#)
- [Overview on page 182](#)
- [Configuration on page 183](#)
- [Verification on page 183](#)

## 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](#).

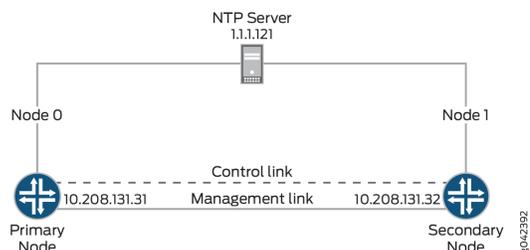
## Overview

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 18 on page 182](#) shows the time synchronization from the peer node using the control link.

**Figure 18: 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 183](#)
- [Results on page 183](#)

### 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 183](#)
- [Verifying the NTP Configuration on the Secondary Node on page 185](#)

### 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 181](#)
- [Verifying Chassis Cluster Configuration Synchronization Status on page 180](#)



## PART 4

# Additional Chassis Cluster Configurations

- [Configuring Active/Passive Chassis Cluster Deployments on page 191](#)
- [Enabling Multicast Routing or Asymmetric Routing on page 225](#)
- [Configuring Chassis Cluster Layer 2 Ethernet Switching on page 241](#)
- [Configuring Media Access Control Security \(MACsec\) on page 249](#)



# Configuring Active/Passive Chassis Cluster Deployments

- Understanding Active/Passive Chassis Cluster Deployment on page 191
- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 192
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 203
- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 205
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 206
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 221

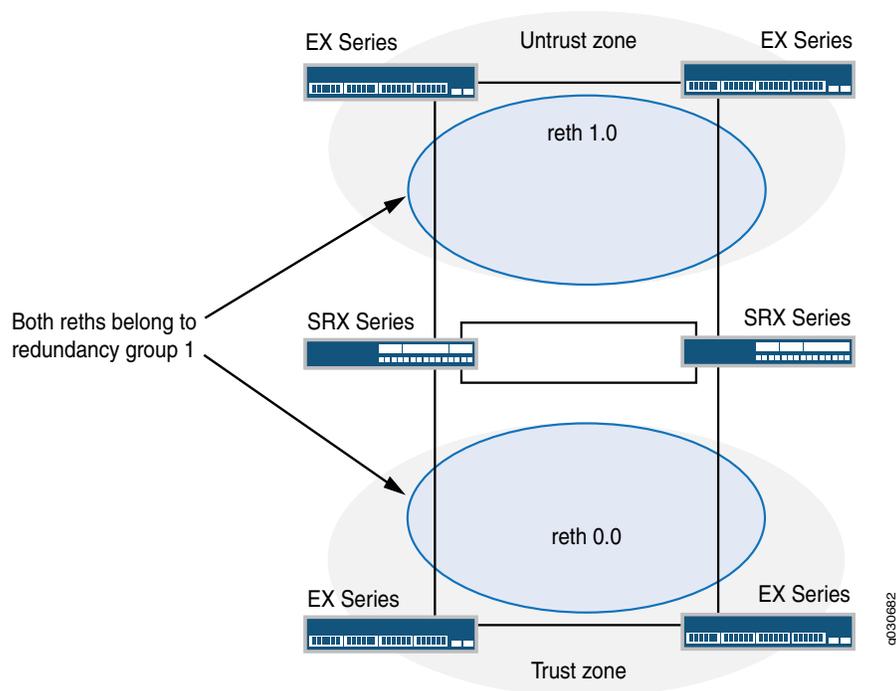
## Understanding Active/Passive Chassis Cluster Deployment

---

**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 19 on page 192](#)). When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 19: 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 192](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair \(J-Web\) on page 203](#)

## 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 192](#)
- [Overview on page 193](#)
- [Configuration on page 195](#)
- [Verification on page 200](#)

### 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
```

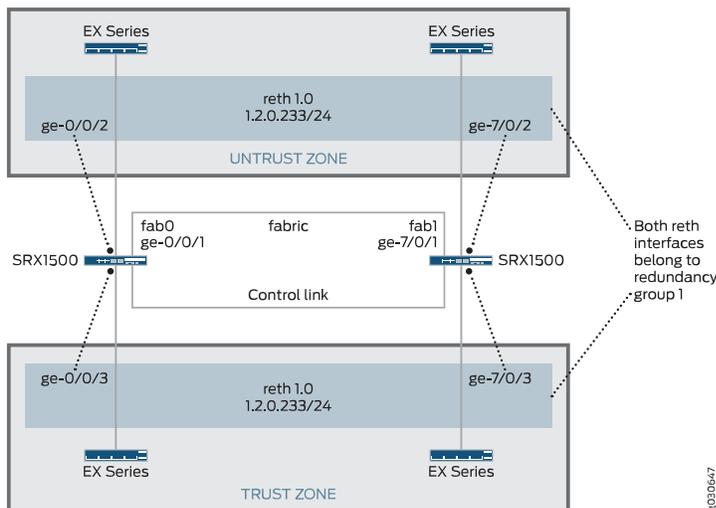
5. Connect to the other device using the console port and set the cluster ID and node number.

```
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 20 on page 193](#).) When a failure occurs, the backup device becomes master and controls all forwarding.

**Figure 20: 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 11 on page 194](#) through [Table 14 on page 195](#).

Table 11: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: srx1500-A</li> <li>• Interface: fxp0 <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.0.2.110/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: srx1500-B</li> <li>• Interface: fxp0 <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.0.2.111/24</li> </ul> </li> </ul>

Table 12: 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	<ul style="list-style-type: none"> <li>• Priority: <ul style="list-style-type: none"> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul> </li> </ul>
	1	<ul style="list-style-type: none"> <li>• Priority: <ul style="list-style-type: none"> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul> </li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>• ge-0/0/4</li> <li>• ge-7/0/4</li> <li>• ge-0/0/5</li> <li>• ge-7/0/5</li> </ul>
Number of redundant Ethernet interfaces	–	2
Interfaces	ge-0/0/4	Redundant parent: reth1
	ge-7/0/4	Redundant parent: reth1
	ge-0/0/5	Redundant parent: reth0
	ge-7/0/5	Redundant parent: reth0

Table 12: Chassis Cluster Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
	reth0	Redundancy group: 1
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>198.51.100.1/24</li> </ul>
	reth1	Redundancy group: 1
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>203.0.113.233/24</li> </ul>

Table 13: 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 14: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

## Configuration

**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 {
    ggether-options {
      redundant-parent reth0;
    }
  }
  ge-7/0/4 {
    ggether-options {
      redundant-parent reth0;
    }
  }
  ge-0/0/5 {
    ggether-options {
      redundant-parent reth1;
    }
  }
}
```

```
ge-7/0/5 {
  gigger-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 on page 200](#)
- [Verifying Chassis Cluster Interfaces on page 200](#)
- [Verifying Chassis Cluster Statistics on page 201](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 202](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 202](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 203](#)
- [Troubleshooting with Logs on page 203](#)

### 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 191](#)
  - [Example: Configuring an Active/Passive Chassis Cluster Pair \(J-Web\) on page 203](#)

## 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 192.
2. Configure the management interface. See Step 2 in “[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)” on page 192.
3. Configure the fabric interface. See Step 3 in “[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)” on page 192.
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 192](#) 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 192](#).

7. Configure the security policies. See Step 7 in “[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)” on page 192.
8. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

**Related Documentation**

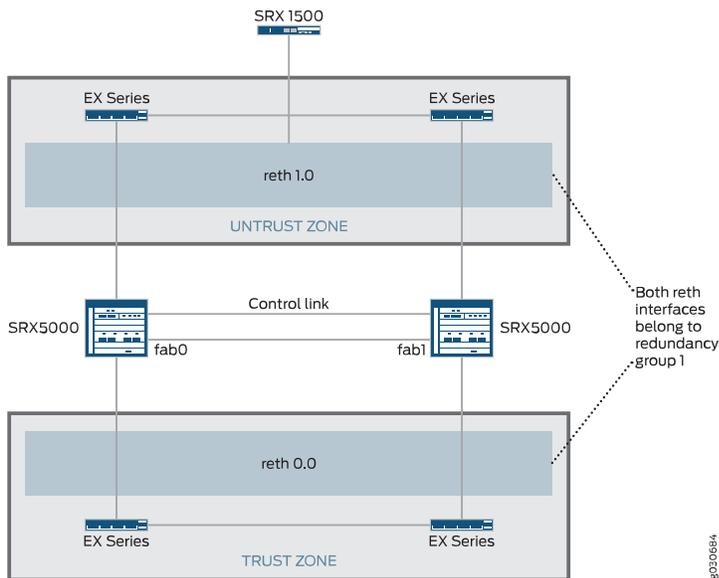
- [Understanding Active/Passive Chassis Cluster Deployment on page 191](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\) on page 192](#)

## 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 21 on page 205](#)). When a failure occurs, the backup device becomes master and controls all forwarding.

**Figure 21: 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 191](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 206](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel \(J-Web\) on page 221](#)

## 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 on page 206](#)
- [Overview on page 207](#)
- [Configuration on page 211](#)
- [Verification on page 218](#)

### 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.

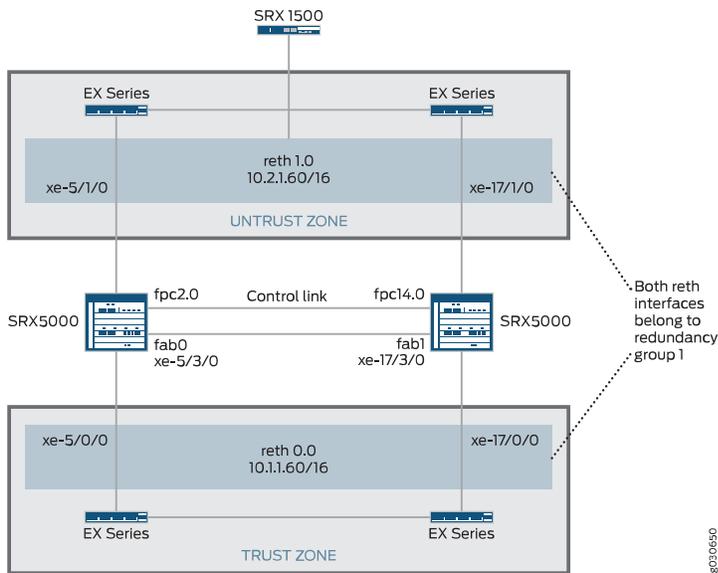
- Get two SRX5000 models with identical hardware configurations, one SRX1500 edge router, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control 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 22 on page 207](#).) When a failure occurs, the backup device becomes master and controls all forwarding.

**Figure 22: 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 15 on page 208](#) through [Table 21 on page 210](#).

Table 15: Group and Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: SRX5800-1</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 172.19.100.50/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: SRX5800-2</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 172.19.100.51/24</li> </ul> </li> </ul>

Table 16: Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: xe-5/3/0
	fab1	Interface: xe-17/3/0
Number of redundant Ethernet interfaces	–	2
Heartbeat interval	–	1000
Heartbeat threshold	–	3
Redundancy group	0	<ul style="list-style-type: none"> <li>• Priority:               <ul style="list-style-type: none"> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul> </li> </ul>
	1	<ul style="list-style-type: none"> <li>• Priority:               <ul style="list-style-type: none"> <li>• Node 0: 254</li> <li>• Node 1: 1</li> </ul> </li> <li>•</li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>• xe-5/0/0</li> <li>• xe-5/1/0</li> <li>• xe-17/0/0</li> <li>• xe-17/1/0</li> </ul>
Interfaces	xe-5/1/0	Redundant parent: reth1
	xe-5/1/0	Redundant parent: reth1
	xe-5/0/0	Redundant parent: reth0

Table 16: Chassis Cluster Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
	xe-17/0/0	Redundant parent: reth0
	reth0	Redundancy group: 1
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.1.1.60/16</li> </ul>
	reth1	Redundancy group: 1
		<ul style="list-style-type: none"> <li>Multipoint</li> <li>Unit 0</li> <li>10.10.1.1/30</li> </ul>
	st0	
		<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.10.1.1/30</li> </ul>

Table 17: IKE Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	-
Policy	preShared	<ul style="list-style-type: none"> <li>Mode: main</li> <li>Proposal reference: proposal-set standard</li> <li>IKE Phase 1 policy authentication method: pre-shared-key ascii-text</li> </ul>
Gateway	SRX1500-1	<ul style="list-style-type: none"> <li>IKE policy reference: perShared</li> <li>External interface: reth0.0</li> <li>Gateway address: 10.1.1.90</li> </ul> <p><b>NOTE:</b> 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.</p> <p>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.</p> <p>On all high-end SRX Series devices, the lo0 logical interface cannot be configured with RGO if used as an IKE gateway external interface.</p>

Table 18: IPsec Configuration Parameters

Feature	Name	Configuration Parameters
Proposal	proposal-set standard	-

Table 18: IPsec Configuration Parameters (*continued*)

Feature	Name	Configuration Parameters
Policy	std	–
VPN	SRX1500-1	<ul style="list-style-type: none"> <li>• IKE gateway reference: SRX1500-1</li> <li>• IPsec policy reference: std</li> <li>• Bind to interface: st0.0</li> <li>• VPN monitoring: vpn-monitor optimized</li> <li>• Tunnels established: establish-tunnels immediately</li> </ul> <p><b>NOTE:</b> The manual VPN name and the site-to-site gateway name cannot be the same.</p>

Table 19: 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 20: Security Zone Configuration Parameters

Name	Configuration Parameters
trust	<ul style="list-style-type: none"> <li>• All system services are allowed.</li> <li>• All protocols are allowed.</li> <li>• The reth0.0 interface is bound to this zone.</li> </ul>
untrust	<ul style="list-style-type: none"> <li>• All system services are allowed.</li> <li>• All protocols are allowed.</li> <li>• The reth1.0 interface is bound to this zone.</li> </ul>
vpn	<ul style="list-style-type: none"> <li>• All system services are allowed.</li> <li>• All protocols are allowed.</li> <li>• The st0.0 interface is bound to this zone.</li> </ul>

Table 21: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>• Match criteria: <ul style="list-style-type: none"> <li>• source-address any</li> <li>• destination-address any</li> <li>• application any</li> </ul> </li> <li>• Action: permit</li> </ul>

Table 21: Security Policy Configuration Parameters (*continued*)

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the vpn zone.	vpn-any	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

## Configuration

**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 "${node}"
set interfaces fab0 fabric-options member-interfaces xe-5/3/0
set interfaces fab1 fabric-options member-interfaces xe-17/3/0
set chassis cluster reth-count 2
set chassis cluster heartbeat-interval 1000
set chassis cluster heartbeat-threshold 3
set chassis cluster node 0
set chassis cluster node 1
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 254
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 interface-monitor xe-5/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-5/1/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-17/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-17/1/0 weight 255
set interfaces xe-5/1/0 gige-ether-options redundant-parent reth1
set interfaces xe-17/1/0 gige-ether-options redundant-parent reth1
set interfaces xe-5/0/0 gige-ether-options redundant-parent reth0
set interfaces xe-17/0/0 gige-ether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.60/16
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.1.60/16
set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
set security ike policy preShared mode main
set security ike policy preShared proposal-set standard
set security ike policy preShared pre-shared-key ascii-text "$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 on page 218](#)
- [Verifying Chassis Cluster Interfaces on page 218](#)
- [Verifying Chassis Cluster Statistics on page 219](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 219](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 220](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 220](#)
- [Troubleshooting with Logs on page 221](#)

### 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
  PTP 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-panel 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 191](#)
  - [Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 205](#)
  - [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel \(J-Web\) on page 221](#)

## 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 206.
2. Configure the management interface. See Step 2 in “[Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel](#)” on page 206.
3. Configure the fabric interface. See Step 3 in “[Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel](#)” on page 206.
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 206.
6. Configure the IPsec configuration. See Step 6 in [“Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel”](#) on page 206.
  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 206.
  9. Configure the security policies. See Step 9 in [“Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel”](#) on page 206.
  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 205
- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel](#) on page 206



# Enabling Multicast Routing or Asymmetric Routing

- [Understanding Multicast Routing on a Chassis Cluster on page 225](#)
- [Understanding Asymmetric Routing Chassis Cluster Deployment on page 226](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair on page 228](#)

## 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 226](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair on page 228](#)

## Understanding Asymmetric Routing Chassis Cluster Deployment

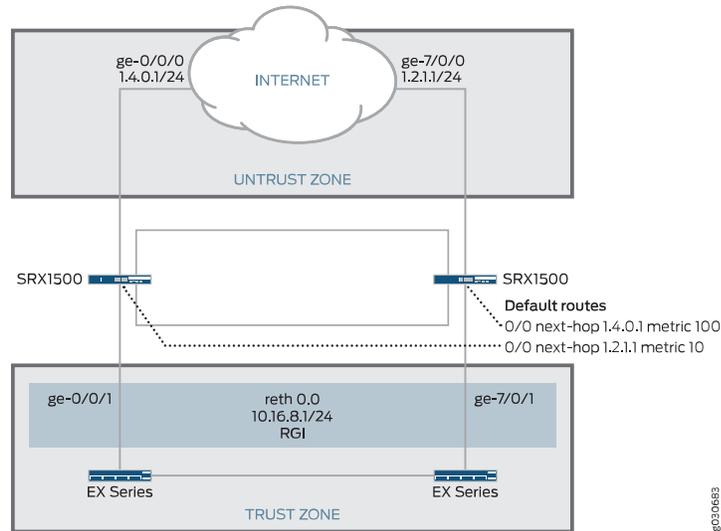
**Supported Platforms** SRX Series, vSRX

In this case, chassis cluster makes use of its asymmetric routing capability (see [Figure 23 on page 227](#)). 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 23: 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 227](#)
- [Understanding Failures in the Untrust Zone Interfaces on page 227](#)

### 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, sessions are only active on node 0.

A failure in interface ge-0/0/1 triggers a failover of the redundancy group, causing interface ge-7/0/1 in node 1 to become active. After the failover, traffic arrives at node 1. After session lookup, the traffic is sent to node 0 because the session is active on this node. Node 0 then processes the traffic and forwards it to the Internet. The return traffic follows a similar process. The traffic arrives at node 0 and gets processed for security purposes—for example, antispam scanning, antivirus scanning, and application of security policies—on node 0 because the session is anchored to node 0. The packet is then sent to node 1 through the fabric interface for egress processing and eventual transmission out of node 1 through interface ge-7/0/1.

### Understanding Failures in the Untrust Zone Interfaces

In this case, sessions are migrated from node to node. Under normal operating conditions, traffic is processed by only node 0. A failure of interface ge-0/0/0 on node 0 causes a change in the routing table, so that it now points to interface ge-7/0/0 in node 1. After

the failure, sessions in node 0 become inactive, and the passive sessions in node 1 become active. Traffic arriving from the trust zone is still received on interface ge-0/0/1, but is forwarded to node 1 for processing. After traffic is processed in node 1, it is forwarded to the Internet through interface ge-7/0/0.

In this chassis cluster configuration, redundancy group 1 is used to control the redundant Ethernet interface connected to the trust zone. As configured in this scenario, redundancy group 1 fails over only if interface ge-0/0/1 or ge-7/0/1 fails, but not if the interfaces connected to the Internet fail. Optionally, the configuration could be modified to permit redundancy group 1 to monitor all interfaces connected to the Internet and fail over if an Internet link were to fail. So, for example, the configuration can allow redundancy group 1 to monitor ge-0/0/0 and make ge-7/0/1 active for reth0 if the ge-0/0/0 Internet link fails. (This option is not described in the following configuration examples.)

#### Related Documentation

- [Understanding Multicast Routing on a Chassis Cluster on page 225](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair on page 228](#)

## 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 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. 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.
 

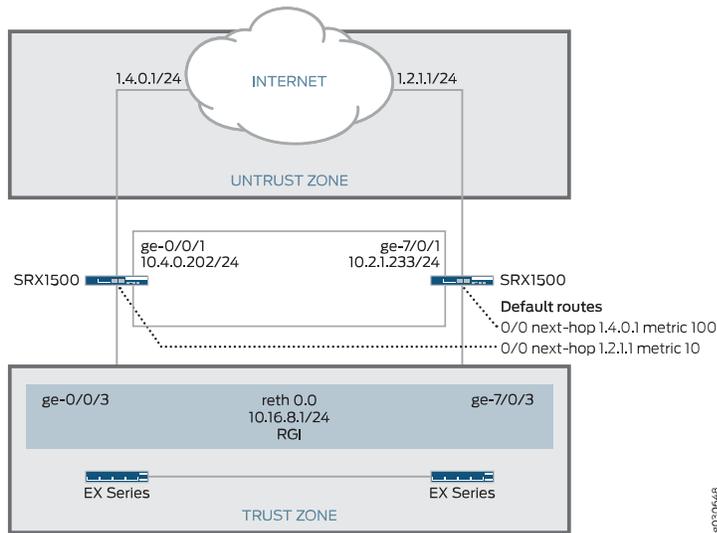
```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```
3. Connect to the other device using the console port.
  - a. Set the cluster ID and node number.

```
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 24 on page 229](#), 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 24: 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 22 on page 229](#) through [Table 25 on page 231](#).

**Table 22: Group and Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Groups	node0	<ul style="list-style-type: none"> <li>• Hostname: srxseries-1</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.100.50/24</li> </ul> </li> </ul>
	node1	<ul style="list-style-type: none"> <li>• Hostname: srxseries-2</li> <li>• Interface: fxp0               <ul style="list-style-type: none"> <li>• Unit 0</li> <li>• 192.168.100.51/24</li> </ul> </li> </ul>

Table 23: Chassis Cluster Configuration Parameters

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: ge-0/0/7
	fab1	Interface: ge-7/0/7
Heartbeat interval	–	1000
Heartbeat threshold	–	3
Redundancy group	1	<ul style="list-style-type: none"> <li>Priority: <ul style="list-style-type: none"> <li>Node 0: 100</li> <li>Node 1: 1</li> </ul> </li> </ul>
		Interface monitoring <ul style="list-style-type: none"> <li>ge-0/0/3</li> <li>ge-7/0/3</li> </ul>
Number of redundant Ethernet interfaces	–	1
Interfaces	ge-0/0/1	<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.4.0.202/24</li> </ul>
	ge-7/0/1	<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.2.1.233/24</li> </ul>
	ge-0/0/3	<ul style="list-style-type: none"> <li>Redundant parent: reth0</li> </ul>
	ge-7/0/3	<ul style="list-style-type: none"> <li>Redundant parent: reth0</li> </ul>
	reth0	<ul style="list-style-type: none"> <li>Unit 0</li> <li>10.16.8.1/24</li> </ul>

Table 24: 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 25: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>

## Configuration

**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 {
        gigger-options {
            redundant-parent reth0;
        }
    }
    ge-7/0/3 {
        gigger-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 {
        gigger-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 on page 236](#)
- [Verifying Chassis Cluster Interfaces on page 236](#)
- [Verifying Chassis Cluster Statistics on page 236](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 237](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 238](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 238](#)
- [Troubleshooting with Logs on page 238](#)

### 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 225](#)
  - [Understanding Asymmetric Routing Chassis Cluster Deployment on page 226](#)



# Configuring Chassis Cluster Layer 2 Ethernet Switching

- Layer 2 Ethernet Switching Capability in Chassis Cluster Mode on page 241
- Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode (CLI) on page 243
- Example: Configuring IRB and VLAN with Members Across Two Nodes (CLI) on page 244
- Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI) on page 246

## Layer 2 Ethernet Switching Capability in Chassis Cluster Mode

---

**Supported Platforms** SRX550M, vSRX

- Understanding Layer 2 Ethernet Switching Capability in Chassis Cluster on SRX Series Devices on page 241
- Understanding Chassis Cluster Failover and New Primary Election on page 242

## Understanding Layer 2 Ethernet Switching Capability in Chassis Cluster on SRX Series Devices

Ethernet ports support various Layer 2 features such as Spanning Tree Protocols (xSTP), DOT1X, Link Aggregation (LAG), Internet Group Membership Protocol (IGMP), GARP VLAN Registration Protocol (GVRP), Link Layer Discovery Protocol (LLDP), and snooping. The enhanced feature extends Layer 2 switching capability to devices in a chassis cluster. This feature allows users to use Ethernet switching features on both nodes of a chassis cluster. The Ethernet ports on either of the nodes can be configured for family Ethernet switching. Users can configure a Layer 2 VLAN domain with member ports from both the nodes and the Layer 2 switching protocols on both the devices.



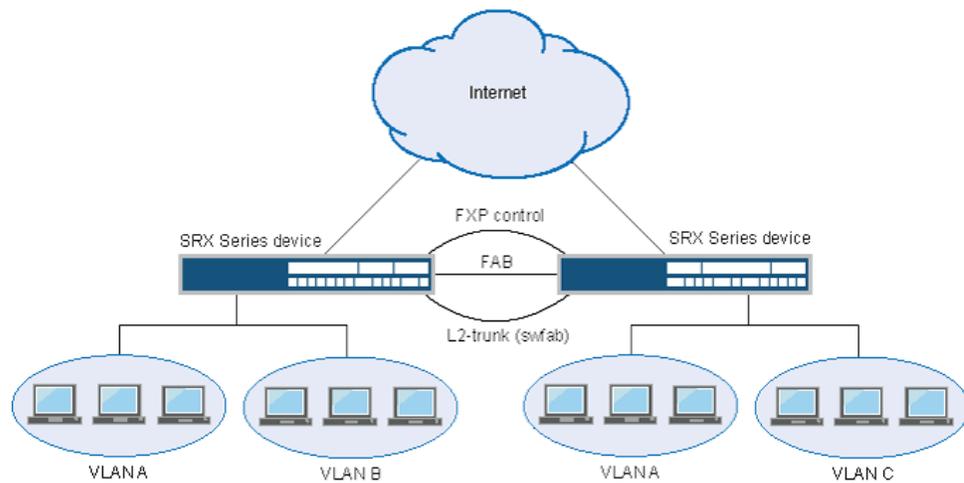
**NOTE:** On SRX550M devices, Layer 2 Ethernet switching is supported in chassis cluster mode.



**NOTE:** On SRX300, SRX320, SRX340, SRX345, and SRX1500 devices, Layer 2 Ethernet switching is not supported in chassis cluster mode.

Figure 25 on page 242 shows the Layer 2 switching across chassis cluster nodes:

Figure 25: Layer 2 Ethernet Switching Across Chassis Cluster Nodes



To ensure that Layer 2 switching works seamlessly across chassis cluster nodes, a dedicated physical link connecting the nodes is required. This type of link is called a *switching fabric interface (swfab)*. Its purpose is to carry Layer 2 traffic between the nodes.



**NOTE:** Configuring a LAG with members across nodes is not supported.



**WARNING:** If a swfab interface is not configured on both the nodes and if you try to configure Ethernet switching-related features on the nodes, behavior of the nodes might be unpredictable.

## Understanding Chassis Cluster Failover and New Primary Election

When chassis cluster failover occurs, a new primary node is elected and the Ethernet Switching Daemon (ESWD) runs in a different node. During failover, the chassis control subsystem is restarted. Also during failover, the traffic outage occurs until the PICs are up and the VLAN entries are reprogrammed. After failover, all Layer 2 protocols reconverge because Layer 2 protocols states are not maintained in the secondary node.



**NOTE:** The Q-in-Q feature in chassis cluster mode is not supported because of chip limitation for swfab interface configuration in Broadcom chipsets.

### Related Documentation

- [Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode \(CLI\) on page 243](#)
- [Example: Configuring IRB and VLAN with Members Across Two Nodes \(CLI\) on page 244](#)
- [Example: Configuring Aggregated Ethernet Device with LAG and LACP \(CLI\) on page 246](#)

## Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode (CLI)

---

**Supported Platforms** SRX1500, SRX550M, vSRX

This example shows how to configure swfab to enable switching in chassis cluster mode.

- [Requirements on page 243](#)
- [Overview on page 243](#)
- [Configuration on page 243](#)

### Requirements

The physical link used as the switch fabric members must be directly connected. Switching supported ports must be used for swfab interfaces.

Before you begin, read through the following example to understand the configuration of chassis cluster fabric:

- [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)

### Overview

New pseudointerfaces swfab0 and swfab1 will be created for Layer 2 fabric functionality. Users need to configure dedicated Ethernet ports on each side of the node to be associated with the swfab interface.

### Configuration

#### Step-by-Step Procedure

To configure swfab interfaces:

1. Configure swfab0 and swfab1 to associate switch fabric interfaces to enable switching across the nodes. Note that swfab0 corresponds to node 0 and swfab1 corresponds to node 1.

```
{primary:node0} [edit]
user@host# set interfaces swfab0 fabric-options member-interfaces ge-0/0/6
user@host# set interfaces swfab0 fabric-options member-interfaces ge-0/0/7
user@host# set interfaces swfab1 fabric-options member-interfaces ge-5/0/6
user@host# set interfaces swfab1 fabric-options member-interfaces ge-5/0/7
```

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

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

#### Verification

---

**Purpose** Verify that the user will be allowed to configure multiple ports as members of swfab ports.

**Action** From configuration mode, enter the **show interfaces swfab0** command to view the configured interfaces for each port.

```
user@host# show interfaces swfab0
fabric-options{
  member-interfaces {
    ge-0/0/6;
    ge-0/0/7;
  }
}
```

From the configuration mode, enter the **show chassis cluster ethernet-switching interfaces** command to view the appropriate member interfaces.

```
user@host# show chassis cluster ethernet-switching interfaces
swfab0:
  Name           Status
  ge-0/0/6       up
  ge-0/0/7       up
swfab1:
  Name           Status
  ge-5/0/6       up
  ge-5/0/7       up
```

**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 34](#)

## Example: Configuring IRB and VLAN with Members Across Two Nodes (CLI)

**Supported Platforms** [SRX1500, SRX550M, vSRX](#)

- [Requirements on page 244](#)
- [Overview on page 244](#)
- [Configuration on page 244](#)
- [Verification on page 246](#)

### Requirements

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

### Overview

This example shows configuration of IRB and configuration of VLAN with members across node 0 and node 1.

### Configuration

**Step-by-Step Procedure** To configure VLAN, follow the steps from 1 to 4 and then commit the configuration. To configure IRB, follow the steps from 1 to 8.

1. Configure Ethernet switching on the node0 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-2/0/0 unit 0 family ethernet-switching
```

- Configure Ethernet switching on the node1 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-11/0/0 unit 0 family ethernet-switching
```

- Create VLAN vlan10 with vlan-id 10.

```
{primary:node0} [edit]
user@host# set vlans vlan10 vlan-id 10
```

- Add ports from both nodes to the VLAN.

```
{primary:node0} [edit]
user@host# set vlans vlan10 interface ge-2/0/0
user@host# set vlans vlan10 interface ge-11/0/0
```

- Assign an IP address to the VLAN.

```
{primary:node0} [edit]
user@host# set interfaces vlan unit 10 family inet address 45.45.45.1/24
```

- Associate Layer 3 VLAN interface to vlan10.

```
{primary:node0} [edit]
user@host# set vlans vlan10 l3-interface vlan.10
```

- Check the configuration by entering the **show vlans** and **show interfaces** commands.

```
user@host# show vlans
vlan10 {
  vlan-id 10;
  interface {
    ge-2/0/0.0;
    ge-11/0/0.0;
  }
  l3-interface vlan.10;
}

user@host# show interfaces
ge-2/0/0 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-11/0/0 {
  unit 0 {
    family ethernet-switching;
  }
}
vlan {
  unit 10 {
    family inet {
      address 45.45.45.1/24;
    }
  }
}
```

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

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

## Verification

### Verifying VLAN and IRB

**Purpose** To verify that the configurations of VLAN and IRB are working properly.

**Action** From configuration mode, enter the **show interfaces terse ge-2/0/0** command to view the node 0 interface.

```
user@host# run show interfaces terse ge-2/0/0
Interface      Admin Link Proto  Local          Remote
ge-2/0/0       up    up
ge-2/0/0.0     up    up    eth-switch
```

From configuration mode, enter the **show interfaces terse ge-11/0/0** command to view the node 1 interface.

```
user@host# run show interfaces terse ge-11/0/0
Interface      Admin Link Proto  Local          Remote
ge-11/0/0      up    up
ge-11/0/0.0    up    up    eth-switch
```

From configuration mode, enter the **show vlans** command to view the VLAN interface.

```
user@host# run show vlans
Name      Tag      Interfaces
default   1        None
vlan10    10       ge-2/0/0.0*, ge-11/0/0.0*
```

**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 34](#)

### Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI)

**Supported Platforms** SRX1500, SRX550M, vSRX

- [Requirements on page 246](#)
- [Overview on page 246](#)
- [Configuration on page 247](#)
- [Verification on page 248](#)

## Requirements

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

## Overview

This example shows the configuration of aggregated Ethernet (ae) devices with LAG and LACP.

## Configuration

### Step-by-Step Procedure

To configure LAG:

1. Configure the number of ae devices with LAG interface that you need to create.
 

```
[edit]
user@host# set chassis aggregated-devices ethernet device-count 5
```
2. Add a port to the ae device with LAG.
 

```
[edit]
user@host# set interfaces ge-2/0/1 gigether-options 802.3ad ae0
user@host# set interfaces ge-2/0/2 gigether-options 802.3ad ae0
```
3. Configure LACP for the ae device with LAG.
 

```
[edit]
user@host# set interfaces ae0 aggregated-ether-options lacp active
```
4. Configure family Ethernet switching for the ae device with LAG.
 

```
[edit]
user@host# set interfaces ae0 unit 0 family ethernet-switching
```
5. Configure VLAN.
 

```
[edit]
user@host# set vlans vlan20 vlan-id 20
```
6. Add the ae interface to the VLAN.
 

```
[edit]
user@host# set vlans vlan20 interface ae0
```
7. Check the configuration by entering the **show vlans** and **show interfaces** commands

```
user@host# show vlans
vlan20 {
  vlan-id 20;
  interface {
    ae0.0;
  }
}

user@host# show interfaces
ge-2/0/1 {
  gigether-options {
    802.3ad ae0;
  }
}
ge-2/0/2 {
  gigether-options {
    802.3ad ae0;
  }
}
ae0 {
  aggregated-ether-options {
    lacp {
      active;
    }
  }
  unit 0 {
```

```

        family ethernet-switching;
    }
}

```

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

```

[edit]
user@host# commit

```



**NOTE:** Likewise, you can configure other devices with LAG and LACP.

## Verification

### Verifying Aggregated Ethernet Device with LAG and LACP

**Purpose** Verify that you can configure ae devices with LAG and LACP.

**Action** From configuration mode, enter the **show lacp interfaces** to view the LACP interfaces.

```

user@host# run show lacp interfaces
Aggregated interface: ae0
LACP state:      Role  Exp  Def  Dist  Col  Syn  Aggr  Timeout  Activity
ge-2/0/1        Actor No   No   Yes  Yes  Yes  Yes  Fast     Active
ge-2/0/1        Partner No   No   Yes  Yes  Yes  Yes  Fast     Active
ge-2/0/2        Actor No   No   Yes  Yes  Yes  Yes  Fast     Active
ge-2/0/2        Partner No   No   Yes  Yes  Yes  Yes  Fast     Active
LACP protocol:  Receive State  Transmit State  Mux State
ge-2/0/1        Current  Fast periodic  Collecting distributing
ge-2/0/2        Current  Fast periodic  Collecting distributing

```

From configuration mode, enter the **show vlans** command to view the VLAN interfaces.

```

user@host# run show vlans
Name      Tag  Interfaces
default   1    None
vlan20    20   ae0.0

```

From configuration mode, enter the **show interfaces (interface name)** command to view the status of the ge-2/0/1 and ge-2/0/2 interfaces.

```

user@host# run show interfaces ge-2/0/1 terse
Interface      Admin Link Proto  Local          Remote
ge-2/0/1       up    up
ge-2/0/1.0     up    up   aenet  --> ae0.0

user@host# run show interfaces ge-2/0/2 terse
Interface      Admin Link Proto  Local          Remote
ge-2/0/2       up    up
ge-2/0/2.0     up    up   aenet  --> ae0.0

```

**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 34](#)

# Configuring Media Access Control Security (MACsec)

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)

## Understanding Media Access Control Security (MACsec) for SRX Series

---

**Supported Platforms** [SRX340, SRX345](#)

Media Access Control Security (MACsec) is an industry-standard security technology that provides secure communication for all traffic on Ethernet links. MACsec provides point-to-point security on Ethernet links between directly connected nodes and is capable of identifying and preventing most security threats, including denial of service, intrusion, man-in-the-middle, masquerading, passive wiretapping, and playback attacks.

MACsec allows you to secure an Ethernet link for almost all traffic, including frames from the Link Layer Discovery Protocol (LLDP), Link Aggregation Control Protocol (LACP), Dynamic Host Configuration Protocol (DHCP), Address Resolution Protocol (ARP), and other protocols that are not typically secured on an Ethernet link because of limitations with other security solutions. MACsec can be used in combination with other security protocols such as IP Security (IPsec) and Secure Sockets Layer (SSL) to provide end-to-end network security.

This topic contains the following sections:

- [How MACsec Works on page 249](#)
- [Understanding Connectivity Associations and Secure Channels on page 250](#)
- [Understanding Static Connectivity Association Key Security Mode on page 250](#)
- [MACsec Considerations on page 251](#)

### How MACsec Works

MACsec provides industry-standard security through the use of secured point-to-point Ethernet links. The point-to-point links are secured after matching security keys—a user-configured pre-shared key when you enable MACsec using static connectivity association key (CAK) security mode, a user-configured static secure association key when you enable MACsec using static secure association key (SAK) security mode—are

exchanged and verified between the interfaces at each end of the point-to-point Ethernet link. Other user-configurable parameters, such as MAC address or port, must also match on the interfaces on each side of the link to enable MACsec.

Once MACsec is enabled on a point-to-point Ethernet link, all traffic traversing the link is MACsec-secured through the use of data integrity checks and, if configured, encryption.

The data integrity checks verify the integrity of the data. MACsec appends an 8-byte header and a 16-byte tail to all Ethernet frames traversing the MACsec-secured point-to-point Ethernet link, and the header and tail are checked by the receiving interface to ensure that the data was not compromised while traversing the link. If the data integrity check detects anything irregular about the traffic, the traffic is dropped.

MACsec can also be used to encrypt all traffic on the Ethernet link. The encryption used by MACsec ensures that the data in the Ethernet frame cannot be viewed by anybody monitoring traffic on the link. MACsec encryption is optional and user-configurable; you can enable MACsec to ensure the data integrity checks are performed while still sending unencrypted data “in the clear” over the MACsec-secured link, if desired.

MACsec is configured on point-to-point Ethernet links between MACsec-capable interfaces. If you want to enable MACsec on multiple Ethernet links, you must configure MACsec individually on each point-to-point Ethernet link.

## Understanding Connectivity Associations and Secure Channels

MACsec is configured in connectivity associations. MACsec is enabled when a connectivity association is assigned to an interface.

When you are configuring MACsec using static secure association key (SAK) security mode, you must configure secure channels within a connectivity association. The secure channels are responsible for transmitting and receiving data on the MACsec-enabled link, and also responsible for transmitting SAKs across the link to enable and maintain MACsec. A single secure channel is uni-directional—it can only be used to apply MACsec to inbound or outbound traffic. A typical connectivity association when MACsec is enabled using SAK security mode contains two secure channels—one secure channel for inbound traffic and another secure channel for outbound traffic.

When you enable MACsec using static CAK or dynamic security mode, you have to create and configure a connectivity association. Two secure channels—one secure channel for inbound traffic and another secure channel for outbound traffic—are automatically created. The automatically-created secure channels do not have any user-configurable parameters; all configuration is done in the connectivity association outside of the secure channels.

## Understanding Static Connectivity Association Key Security Mode

When you enable MACsec using static connectivity association key (CAK) security mode, two security keys—a connectivity association key (CAK) that secures control plane traffic and a randomly-generated secure association key (SAK) that secures data plane traffic—are used to secure the point-to-point Ethernet link. Both keys are regularly exchanged between both devices on each end of the point-to-point Ethernet link to ensure link security.

You initially establish a MACsec-secured link using a pre-shared key when you are using static CAK security mode to enable MACsec. A pre-shared key includes a connectivity association name (CKN) and its own connectivity association key (CAK). The CKN and CAK are configured by the user in the connectivity association and must match on both ends of the link to initially enable MACsec.

Once matching pre-shared keys are successfully exchanged, the MACsec Key Agreement (MKA) protocol is enabled. The MKA protocol is responsible for maintaining MACsec on the link, and decides which switch on the point-to-point link becomes the key server. The key server then creates an SAK that is shared with the switch at the other end of the point-to-point link only, and that SAK is used to secure all data traffic traversing the link. The key server will continue to periodically create and share a randomly-created SAK over the point-to-point link for as long as MACsec is enabled.

You enable MACsec using static CAK security mode by configuring a connectivity association on both ends of the link. All configuration is done within the connectivity association but outside of the secure channel. Two secure channels—one for inbound traffic and one for outbound traffic—are automatically created when using static CAK security mode. The automatically-created secure channels do not have any user-configurable parameters that cannot already be configured in the connectivity association.

We recommend enabling MACsec using static CAK security mode. Static CAK security mode ensures security by frequently refreshing to a new random security key and by only sharing the security key between the two devices on the MACsec-secured point-to-point link. Additionally, some optional MACsec features—replay protection, SCI tagging, and the ability to exclude traffic from MACsec—are only available when you enable MACsec using static CAK security mode.

## MACsec Considerations

All types of Spanning Tree Protocol frames cannot currently be encrypted using MACsec.

The connectivity association can be defined anywhere, either global or node specific or any other configuration group as long as it is visible to the MACsec interface configuration.



**NOTE:** 802.1x protocol process (daemon) does not support restart on the SRX345 device.

### Related Documentation

- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)

## Configuring Media Access Control Security (MACsec)

**Supported Platforms** [SRX340, SRX345](#)

Media Access Control Security (MACsec) is an industry-standard security technology that provides secure communication for almost all types of traffic on Ethernet links.

MACsec provides point-to-point security on Ethernet links between directly-connected nodes and is capable of identifying and preventing most security threats, including denial of service, intrusion, man-in-the-middle, masquerading, passive wiretapping, and playback attacks.

You can configure MACsec to secure point-to-point Ethernet links connecting SRX Series devices, or on Ethernet links connecting a device to a host device such as a PC, phone, or server. Each point-to-point Ethernet link that you want to secure using MACsec must be configured independently. You can enable MACsec on device-to-device links using static connectivity association key (CAK) security mode.



**BEST PRACTICE:** We recommend enabling MACsec using static CAK security mode on device-to-device links. Static CAK security mode ensures security by frequently refreshing to a new random secure association key (SAK) and by only sharing the SAK between the two devices on the MACsec-secured point-to-point link. Additionally, some optional MACsec features—replay protection, SCI tagging, and the ability to exclude traffic from MACsec—are only available for MACsec-secured device-to-device connections that are enabled using static CAK security mode.

The configuration steps for both processes are provided in this document.

- [Configuring MACsec Using Static Connectivity Association Key Security Mode on page 253](#)
- [Configuring Static CAK on the Chassis Cluster Control Port on page 257](#)
- [Configuring Static CAK on the Chassis Cluster Fabric Port on page 258](#)
- [Considerations for Configuring MACsec on Control Ports on page 258](#)
- [Configuring MACsec on Control Ports on page 258](#)
- [Configuring MACsec on Fabric Ports on page 259](#)

## Configuring MACsec Using Static Connectivity Association Key Security Mode

You can enable MACsec using static connectivity association key (CAK) security mode on a point-to-point Ethernet link connecting devices. This procedure shows you how to configure MACsec using static CAK security mode.



**BEST PRACTICE:** We recommend enabling MACsec using static CAK security mode on device-to-device links. Static CAK security mode ensures security by frequently refreshing to a new random secure association key (SAK) and by only sharing the SAK between the two devices on the MACsec-secured point-to-point link. Additionally, some optional MACsec features—replay protection, SCI tagging, and the ability to exclude traffic from MACsec—are only available for MACsec-secured device-to-device connections that are enabled using static CAK security mode.

When you enable MACsec using static CAK security mode, a pre-shared key is exchanged between the devices on each end of the point-to-point Ethernet link. The pre-shared key includes a connectivity association name (CKN) and a connectivity association key (CAK). The CKN and CAK are configured by the user in the connectivity association and must match on both ends of the link to initially enable MACsec.

After the pre-shared keys are exchanged and verified, the MACsec Key Agreement (MKA) protocol, which enables and maintains MACsec on the link, is enabled. The MKA is responsible for selecting one of the two devices on the point-to-point link as the key server. The key server then creates a randomized security key that is shared only with the other device over the MACsec-secured link. The randomized security key enables and maintains MACsec on the point-to-point link. The key server will continue to periodically create and share a randomly-created security key over the point-to-point link for as long as MACsec is enabled.

You enable MACsec using static CAK security mode by configuring a connectivity association on both ends of the link. All configuration is done within the connectivity association but outside of the secure channel. Two secure channels—one for inbound traffic and one for outbound traffic—are automatically created when using static CAK security mode. The automatically-created secure channels do not have any user-configurable parameters that cannot already be configured in the connectivity association.

To configure MACsec using static CAK security mode to secure a device-to-device Ethernet link:

1. Create a connectivity association. You can skip this step if you are configuring an existing connectivity association.

```
[edit security macsec]
user@host# set connectivity-association connectivity-association-name
```

For instance, to create a connectivity association named **ca1**, enter:

```
[edit security macsec]
user@host# set connectivity-association ca1
```

2. Configure the MACsec security mode as **static-cak** for the connectivity association.

```
[edit security macsec]
user@host# set connectivity-association connectivity-association-name security-mode
static-cak
```

For instance, to configure the MACsec security mode to **static-cak** on connectivity association ca1:

```
[edit security macsec]
user@host# set connectivity-association ca1 security-mode static-cak
```

3. Create the pre-shared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

```
[edit security macsec]
user@host# set connectivity-association connectivity-association-name pre-shared-key ckn
hexadecimal-number
user@host# set connectivity-association connectivity-association-name pre-shared-key cak
hexadecimal-number
```

A pre-shared key is exchanged between directly-connected links to establish a MACsec-secure link. The pre-shared-key includes the CKN and the CAK. The CKN is a 64-digit hexadecimal number and the CAK is a 32-digit hexadecimal number. The CKN and the CAK must match on both ends of a link to create a MACsec-secured link.



**NOTE:** To maximize security, we recommend configuring all 64 digits of a CKN and all 32 digits of a CAK.

If you do not configure all 64 digits of a CKN or all 32 digits of a CAK, however, all remaining digits will be auto-configured to 0.

After the pre-shared keys are successfully exchanged and verified by both ends of the link, the MACsec Key Agreement (MKA) protocol is enabled and manages the secure link. The MKA protocol then elects one of the two directly-connected devices as the key server. The key server then shares a random security with the other device over the MACsec-secure point-to-point link. The key server will continue to periodically create and share a random security key with the other device over the MACsec-secured point-to-point link as long as MACsec is enabled.

To configure a CKN of **37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311** and CAK of **228ef255aa23ff6729ee664acb66e91f** on connectivity association ca1:

```
[edit security macsec]
user@host# set connectivity-association ca1 pre-shared-key ckn
37c9c2c45ddd012aa5bc8ef284aa23ff6729ee2e4acb66e91fe34ba2cd9fe311
user@host# set connectivity-association ca1 pre-shared-key cak
228ef255aa23ff6729ee664acb66e91f
```



**NOTE:** MACsec is not enabled until a connectivity association is attached to an interface. See the final step of this procedure to attach a connectivity association to an interface.

4. (Optional) Set the MKA key server priority.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set mka key-server-priority priority-number
```

Specifies the key server priority used by the MKA protocol to select the key server. The device with the lower *priority-number* is selected as the key server.

The default *priority-number* is 16.

If the **key-server-priority** is identical on both sides of the point-to-point link, the MKA protocol selects the interface with the lower MAC address as the key server. Therefore, if this statement is not configured in the connectivity associations at each end of a MACsec-secured point-to-point link, the interface with the lower MAC address becomes the key server.

To change the key server priority to 0 to increase the likelihood that the current device is selected as the key server when MACsec is enabled on the interface using connectivity association **ca1**:

```
[edit security macsec connectivity-association ca1]
user@host# set mka key-server-priority 0
```

To change the key server priority to 255 to decrease the likelihood that the current device is selected as the key server in connectivity association **ca1**:

```
[edit security macsec connectivity-association ca1]
user@host# set mka key-server-priority 255
```

- (Optional) Set the MKA transmit interval.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set mka transmit-interval interval
```

The MKA transmit interval setting sets the frequency for how often the MKA protocol data unit (PDU) is sent to the directly connected device to maintain MACsec connectivity on the link. A lower *interval* increases bandwidth overhead on the link; a higher *interval* optimizes MKA protocol communication.

The default *interval* is 2000ms. We recommend increasing the interval to 6000 ms in high-traffic load environments. The transmit interval settings must be identical on both ends of the link when MACsec using static CAK security mode is enabled.

For instance, if you wanted to increase the MKA transmit interval to 6000 milliseconds when connectivity association **ca1** is attached to an interface:

```
[edit security macsec connectivity-association ca1]
user@host# set mka transmit-interval 6000
```

- (Optional) Disable MACsec encryption.

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set no-encryption
```

Encryption is enabled for all traffic entering or leaving the interface when MACsec is enabled using static CAK security mode, by default.

When encryption is disabled, traffic is forwarded across the Ethernet link in clear text. You are able to view unencrypted data in the Ethernet frame traversing the link when you are monitoring it. The MACsec header is still applied to the frame, however, and all MACsec data integrity checks are run on both ends of the link to ensure the traffic sent or received on the link has not been tampered with and does not represent a security threat.

- (Optional) Set an offset for all packets traversing the link.

```
[edit security macsec connectivity-association connectivity-association-name]  
user@host# set offset (0 | 30 | 50)
```

For instance, if you wanted to set the offset to 30 in the connectivity association named **ca1**:

```
[edit security macsec connectivity-association ca1]  
user@host# set offset 30
```

The default offset is 0. All traffic in the connectivity association is encrypted when encryption is enabled and an **offset** is not set.

When the offset is set to 30, the IPv4 header and the TCP/UDP header are unencrypted while encrypting the rest of the traffic. When the offset is set to 50, the IPv6 header and the TCP/UDP header are unencrypted while encrypting the rest of the traffic.

You would typically forward traffic with the first 30 or 50 octets unencrypted if a feature needed to see the data in the octets to perform a function, but you otherwise prefer to encrypt the remaining data in the frames traversing the link. Load balancing features, in particular, typically need to see the IP and TCP/UDP headers in the first 30 or 50 octets to properly load balance traffic.

8. (Optional) Enable replay protection.

```
[edit security macsec connectivity-association connectivity-association-name]  
user@host# set replay-protect replay-window-size number-of-packets
```

When MACsec is enabled on a link, an ID number is assigned to each packet on the MACsec-secured link.

When replay protection is enabled, the receiving interface checks the ID number of all packets that have traversed the MACsec-secured link. If a packet arrives out of sequence and the difference between the packet numbers exceeds the replay protection window size, the packet is dropped by the receiving interface. For instance, if the replay protection window size is set to five and a packet assigned the ID of 1006 arrives on the receiving link immediately after the packet assigned the ID of 1000, the packet that is assigned the ID of 1006 is dropped because it falls outside the parameters of the replay protection window.

Replay protection is especially useful for fighting man-in-the-middle attacks. A packet that is replayed by a man-in-the-middle attacker on the Ethernet link will arrive on the receiving link out of sequence, so replay protection helps ensure the replayed packet is dropped instead of forwarded through the network.

Replay protection should not be enabled in cases where packets are expected to arrive out of order.

You can require that all packets arrive in order by setting the replay window size to 0.

To enable replay protection with a window size of five on connectivity association **ca1**:

```
[edit security macsec connectivity-association ca1]  
user@host# set replay-protect replay-window-size 5
```

9. (Optional) Exclude a protocol from MACsec.

```
[edit security macsec connectivity-association connectivity-association-name]  
user@host# set exclude-protocol protocol-name
```

For instance, if you did not want Link Level Discovery Protocol (LLDP) to be secured using MACsec:

```
[edit security macsec connectivity-association connectivity-association-name]
user@host# set exclude-protocol lldp
```

When this option is enabled, MACsec is disabled for all packets of the specified protocol—in this case, LLDP—that are sent or received on the link.

10. Create a connectivity association for enabling MACsec on a chassis cluster control interface.

```
[edit security macsec]
user@host# set security macsec cluster-control-port <port no> connectivity-association CA
```



**NOTE:** For SRX Series devices, ge-0/0/1 is a control port for the chassis cluster and assigned as cluster-control-port 0.

Assigning the connectivity association to an interface is the final configuration step for enabling MACsec on an interface.

For instance, to assign connectivity association ca1 to interface ge-0/0/1:

```
[edit security macsec]
user@host# set interfaces ge-0/0/1 connectivity-association ca1
```

11. Create a connectivity association for enabling MACsec on a chassis cluster fabric interface.

```
[edit security macsec]
user@host# set security macsec cluster-control-port <port no> connectivity-association
CA_FAB
```



**NOTE:** For SRX Series devices, ge-0/0/0 is a fabric port for the chassis cluster.

MACsec using static CAK security mode is not enabled until a connectivity association on the opposite end of the link is also configured, and contains pre-shared keys that match on both ends of the link.

## Configuring Static CAK on the Chassis Cluster Control Port

To establish a CA over a chassis cluster control link on two SRX Series devices.

1. Configure the MACsec security mode as **static-cak** for the connectivity association:

```
[edit security macsec]
user@host# set security macsec connectivity-association ca1 security-mode static-cak
```

2. Create the pre-shared key by configuring the connectivity association key name (CKN).

```
[edit security macsec]
user@host# set security macsec connectivity-association ca1 pre-shared-key ckn
"MACSEC_KEY_NAME"
```

3. Create the pre-shared key by configuring the connectivity association key (CAK).

```
[edit security macsec]
user@host# set security macsec connectivity-association ca1 pre-shared-key cak
"MACSEC_KEY"
```

4. Specify a chassis cluster control port for the connectivity association.

```
[edit security macsec]
user@host# set security macsec cluster-control-port 0 connectivity-association ca1
```

## Configuring Static CAK on the Chassis Cluster Fabric Port

To establish a CA over a chassis cluster fabric link on two SRX Series devices:

1. Configure the MACsec security mode as **static-cak** for the connectivity association.

```
[edit security macsec]
user@host# set security macsec connectivity-association ca2 security-mode static-cak
```

2. Create the pre-shared key by configuring the connectivity association key name (CKN).

```
[edit security macsec]
user@host# set security macsec connectivity-association ca2 pre-shared-key ckn
"MACSEC_KEY_NAME"
```

3. Create the pre-shared key by configuring the connectivity association key (CAK).

```
[edit security macsec]
user@host# set security macsec connectivity-association ca2 pre-shared-key cak
"MACSEC_KEY"
```

4. Specify a chassis cluster control ports to a connectivity association.

```
[edit security macsec]
user@host# set security macsec cluster-data-port ge-0/0/2 connectivity-association ca2
user@host# set security macsec cluster-data-port ge-5/0/2 connectivity-association ca2
```

## Considerations for Configuring MACsec on Control Ports

Control port states affect the integrity of a chassis cluster. Note the following important points:

- Any new MACsec chassis cluster port configurations or modifications to existing MACsec chassis cluster port configurations will require the chassis cluster to be disabled. Once disabled, you can apply the preceding configurations and reenables the chassis cluster.
- By default, chassis clusters synchronize all configurations. Correspondingly, you must monitor that synchronization does not lead to loss of any MACsec configurations. Otherwise, the chassis cluster will break. For example, for nonsymmetric, node-specific MACsec configurations, identical configurations should exist on both ends. That is, each node should contain the same configuration as the other node.



**NOTE:** The ineligible timer is 5 minutes when MACsec on the chassis cluster control port is enabled on SRX340 and SRX345 devices.

## Configuring MACsec on Control Ports

Follow these steps to configure MACsec on control ports:

1. If the chassis cluster is already up, disable it by using the **set chassis cluster disable** command and reboot both nodes.
2. Configure MACsec on the control port with its attributes as described in the preceding sections “[Configuring Static CAK on the Chassis Cluster Control Port](#)” on page 257. Both nodes must be configured independently with identical configurations.

3. Enable the chassis cluster by using `set chassis cluster cluster-id<id>` on both of the nodes. Reboot both nodes.



**NOTE:** For any change in the MACsec configurations of control ports, the above steps should be repeated.

## Configuring MACsec on Fabric Ports

Configuring MACsec leads to link state changes that can affect traffic capability of the link. When you configure fabric ports, keep the effective link state in mind. Incorrect MACsec configuration on both ends of the fabric links can move the link to an ineligible state. Note the following key points about configuring fabric links:

- Both ends of the links must be configured simultaneously when the chassis cluster is formed.
- Incorrect configuration can lead to fabric failures and errors in fabric recovery logic.



**NOTE:** Because of potential link failure scenarios, we recommend that fabric links be configured during formation of the chassis cluster.

### Related Documentation

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [macsec on page 304](#)



## PART 5

# Upgrading or Disabling Chassis Cluster

- [Upgrading Both Devices Separately on page 263](#)
- [Upgrading Both Devices Using ICU on page 265](#)
- [Disabling Chassis Cluster on page 271](#)



# Upgrading Both Devices Separately

- [Upgrading Individual Devices in a Chassis Cluster Separately on page 263](#)

## 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](#)
- [Upgrading Devices in a Chassis Cluster Using ICU for Branch SRX Series Devices on page 265](#)



# Upgrading Both Devices Using ICU

- [Upgrading Devices in a Chassis Cluster Using ICU on page 265](#)

## Upgrading Devices in a Chassis Cluster Using ICU

---

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

- [Upgrading Both Devices in a Chassis Cluster Using ICU on page 265](#)
- [Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster on page 266](#)
- [Upgrading ICU Using a Build Available on an FTP Server on page 267](#)
- [Aborting an Upgrade in a Chassis Cluster During an ICU on page 268](#)

## Upgrading Both Devices in a Chassis Cluster Using ICU

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

Starting with Junos OS Release 15.1X49-D50, in-band cluster upgrade (ICU) is supported on SRX1500 devices. The SRX1500 devices in a chassis cluster can be upgraded with a minimal service disruption using ICU. The chassis cluster ICU feature allows both devices in a cluster to be upgraded from supported Junos OS versions using a single command. You can enable this feature by executing the **request system software in-service-upgrade *image\_name*** command on the primary node. This command upgrades the Junos OS and reboots both the secondary node and the primary node in turn. During the ICU process, traffic outage is minimal; however, cold synchronization is provided between the two nodes.



**NOTE:** For SRX1500 devices, this feature is available only on Junos OS Release 15.1X49-D50 and later.

---

For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the devices in a chassis cluster can be upgraded with a minimal service disruption of approximately 30 seconds using ICU with the no-sync option. The chassis cluster ICU feature allows both devices in a cluster to be upgraded from supported Junos OS versions.

For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the impact on traffic is as follows:

- Drop in traffic (30 seconds approximately)
- Loss of security flow sessions

Before you begin, note the following:

- ICU is available with the no-sync option only for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.
- This feature is available only on Junos OS Release 11.2 R2 and later for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.
- Before starting ICU, you should ensure that sufficient disk space is available. See [“Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster” on page 266](#) and [“Upgrading ICU Using a Build Available on an FTP Server” on page 267](#).
- For SRX300, SRX320, SRX340, SRX345, and SRX550M devices, this feature cannot be used to downgrade to a build earlier than Junos OS 11.2 R2.

For SRX1500 devices, this feature cannot be used to downgrade to a build earlier than Junos OS 15.1X49-D50.

The upgrade is initiated with the Junos OS build locally available on the primary node of the device or on an FTP server.



NOTE:

- The primary node, RGO, changes to the secondary node after an ICU upgrade.
- During ICU, the chassis cluster redundancy groups are failed over to the primary node to change the cluster to active/passive mode.
- ICU states can be checked from the syslog or with the console/terminal logs.
- ICU requires that both nodes be running a dual-root partitioning scheme. ICU will not continue if it fails to detect dual-root partitioning on either of the nodes.



NOTE: Dual-root partitioning is not supported on SRX1500 devices.

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

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX



**NOTE:** Ensure that sufficient disk space is available for the Junos OS package in the `/var/tmp` location in the secondary node of the cluster.

To upgrade ICU using a build locally available on the primary node of a cluster:

1. Copy the Junos OS package build to the primary node at any location, or mount a network file server folder containing the Junos OS build.
2. Start ICU by entering the following command:

```
user@host> request system software in-service-upgrade image_name no-sync (for
SRX300, SRX320, SRX340, SRX345, and SRX550M devices)
```

```
user@host> request system software in-service-upgrade image_name (for SRX1500
devices)
```

## Upgrading ICU Using a Build Available on an FTP Server

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX



**NOTE:** Ensure that sufficient disk space is available for the Junos OS package in the `/var/tmp` location in both the primary and the secondary nodes of the cluster.

To upgrade ICU using a build available on an FTP server:

1. Place the Junos OS build on an FTP server.
2. (SRX300, SRX320, SRX340, SRX345, and SRX550M only) Start ICU by entering the following command:

```
user@root> request system software in-service-upgrade <ftp url for junos image>
no-sync
```

```
user@root> request system software in-service-upgrade
ftp://<user>:<password>@<server>:/<path> no-sync
```

This command upgrades the Junos OS and reboots both nodes in turn.

3. (SRX1500 only) Start ICU by entering the following command:

```
user@root> request system software in-service-upgrade <ftp url for junos image>
```

```
user@root> request system software in-service-upgrade
ftp://<user>:<password>@<server>:/<path>
```

This command upgrades the Junos OS and reboots both nodes in turn.



**NOTE:** The upgrade process displays the following warning message to reboot the system:

**WARNING:** A reboot is required to load this software correctly. Use the `request system reboot` command when software installation is complete.

This warning message can be ignored because the ICU process automatically reboots both the nodes.

## Aborting an Upgrade in a Chassis Cluster During an ICU

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX550M, vSRX

You can abort an ICU at any time by issuing the following command on the primary node:

```
request system software abort in-service-upgrade
```



**NOTE:** Issuing an `abort` command during or after the secondary node reboots puts the cluster in an inconsistent state. The secondary node boots up running the new Junos OS build, while the primary continues to run the older Junos OS build.

To recover from the chassis cluster inconsistent state, perform the following actions sequentially on the secondary node:

1. Issue an `abort` command:  

```
request system software abort in-service-upgrade
```
2. Roll back the Junos OS build by entering the following command:  

```
request system software rollback node < node-id >
```
3. Reboot the secondary node immediately by using the following command:  

```
request system reboot
```



**NOTE:** You must execute the above steps sequentially to complete the recovery process and avoid cluster instability.

Table 26 on page 269 lists the options and their descriptions for the `request system software in-service-upgrade` command.

Table 26: request system software in-service-upgrade Output Fields

Options	Description
no-sync	Disables the flow state from syncing up when the old secondary node has booted with a new Junos OS image.  <b>NOTE:</b> This option is not available on SRX1500 devices.
no-tcp-syn-check	Creates a window wherein the TCP SYN check for the incoming packets will be disabled. The default value for the window is 7200 seconds (2 hours).  <b>NOTE:</b> This option is not available on SRX1500 devices.
no-validate	Disables the validation of the configuration at the time of the installation. The system behavior is similar to <b>software add</b> .
unlink	Removes the package from the local media after installation.

**NOTE:**

- During ICU, if an abort command is executed, ICU will abort only after the current operation finishes. This is required to avoid any inconsistency with the devices.

For example, if formatting and upgrade of a node is in progress, ICU aborts after this operation finishes.

- After an abort, ICU will try to roll back the build on the nodes if the upgrading nodes step was completed.

**Related Documentation**

- [Verifying a Chassis Cluster Configuration on page 97](#)



# Disabling Chassis Cluster

- [Disabling Chassis Cluster on page 271](#)

## 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 263](#)
- [Upgrading Both Devices in a Chassis Cluster Using an ISSU for High-End SRX Series Devices](#)
- [Upgrading Devices in a Chassis Cluster Using ICU for Branch SRX Series Devices on page 265](#)



## PART 6

# Configuration Statements and Operational Commands

- Configuration Statements on page 275
- Operational Commands on page 333



## CHAPTER 25

# Configuration Statements

- `apply-groups` (Chassis Cluster) on page 277
- `cak` on page 278
- `ckn` on page 279
- `cluster` (Chassis) on page 280
- `configuration-synchronize` (Chassis Cluster) on page 281
- `connectivity-association` on page 282
- `connectivity-association` (MACsec Interfaces) on page 283
- `control-link-recovery` on page 283
- `device-count` (Chassis Cluster) on page 284
- `no-encryption` (MACsec) on page 285
- `exclude-protocol` on page 286
- `ethernet` (Chassis Cluster) on page 287
- `fabric-options` on page 288
- `gether-options` (Chassis Cluster) on page 289
- `global-threshold` on page 290
- `global-weight` on page 291
- `gratuitous-arp-count` on page 292
- `heartbeat-interval` on page 293
- `heartbeat-threshold` on page 294
- `hold-down-interval` on page 295
- `include-sci` on page 296
- `interface` (Chassis Cluster) on page 297
- `interfaces` (MACsec) on page 298
- `interface-monitor` on page 299
- `ip-monitoring` on page 300
- `key-server-priority` (MACsec) on page 301
- `lACP` (Interfaces) on page 302
- `link-protection` (Chassis Cluster) on page 303

- [macsec](#) on page 304
- [member-interfaces](#) on page 305
- [mka](#) on page 305
- [must-secure](#) on page 306
- [network-management](#) on page 307
- [node \(Chassis Cluster\)](#) on page 308
- [node \(Chassis Cluster Redundancy Group\)](#) on page 308
- [offset](#) on page 309
- [preempt \(Chassis Cluster\)](#) on page 310
- [pre-shared-key](#) on page 311
- [priority \(Chassis Cluster\)](#) on page 312
- [redundancy-group \(Chassis Cluster\)](#) on page 313
- [redundancy-interface-process](#) on page 314
- [redundant-ether-options](#) on page 315
- [redundant-parent \(Interfaces\)](#) on page 316
- [redundant-pseudo-interface-options](#) on page 316
- [replay-protect](#) on page 317
- [replay-window-size](#) on page 318
- [reth-count \(Chassis Cluster\)](#) on page 319
- [reth \(Interfaces\)](#) on page 320
- [retry-count \(Chassis Cluster\)](#) on page 325
- [retry-interval \(Chassis Cluster\)](#) on page 326
- [route-active-on](#) on page 326
- [security-mode](#) on page 327
- [traceoptions \(Chassis Cluster\)](#) on page 328
- [transmit-interval \(MACsec\)](#) on page 330
- [weight](#) on page 331

---

## apply-groups (Chassis Cluster)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	apply-groups [ <i>node</i> ]
<b>Hierarchy Level</b>	[edit chassis cluster]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	Apply node-specific parameters to each node in a chassis cluster.
<b>Options</b>	<i>node</i> —Each node (node0 or node1) in a chassis cluster.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">cluster (Chassis) on page 280</a></li></ul>

## cak

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	<i>ckn hexadecimal-number;</i>
<b>Hierarchy Level</b>	[edit security macsec connectivity-association pre-shared-key]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies the connectivity association key (CAK) for a pre-shared key.</p> <p>A pre-shared key includes a connectivity association key name (CKN) and a CAK. A pre-shared key is exchanged between two devices at each end of a point-to-point link to enable MACsec using dynamic security keys. The MACsec Key Agreement (MKA) protocol is enabled once the pre-shared keys are successfully exchanged. The pre-shared key—the CKN and CAK—must match on both ends of a link</p>
<b>Default</b>	No CAK exists, by default.
<b>Options</b>	<p><i>hexadecimal-number</i> —The key name, in hexadecimal format.</p> <p>The key name is 32 hexadecimal characters in length. If you enter a key name that is less than 32 characters long, the remaining characters are set to 0.</p>
<b>Required Privilege Level</b>	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li><li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li><li>• <a href="#">macsec on page 304</a></li></ul>

---

**ckn**

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	<code>ckn hexadecimal-number;</code>
<b>Hierarchy Level</b>	[edit security macsec connectivity-association pre-shared-key]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies the connectivity association key name (CKN) for a pre-shared key.</p> <p>A pre-shared key includes a CKN and a connectivity association key (CAK). A pre-shared key is exchanged between two devices at each end of a point-to-point link to enable MACsec using dynamic security keys. The MACsec Key Agreement (MKA) protocol is enabled once the pre-shared keys are successfully exchanged. The pre-shared key—the CKN and CAK—must match on both ends of a link</p>
<b>Default</b>	No CKN exists, by default.
<b>Options</b>	<p><i>hexadecimal-number</i> —The key name, in hexadecimal format.</p> <p>The key name is 32 hexadecimal characters in length. If you enter a key name that is less than 32 characters long, the remaining characters are set to 0.</p>
<b>Required Privilege Level</b>	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li><li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li><li>• <a href="#">macsec on page 304</a></li></ul>

## 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;
    }
}

```

<b>Hierarchy Level</b>	[edit chassis]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	Configure a chassis cluster.
<b>Options</b>	The remaining statements are explained separately. See <a href="#">CLI Explorer</a> .
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">ip-monitoring on page 300</a></li> </ul>

## configuration-synchronize (Chassis Cluster)

---

<b>Supported Platforms</b>	<a href="#">SRX Series, vSRX</a>
<b>Syntax</b>	<pre> configuration-synchronize {     no-secondary-bootup-auto; } </pre>
<b>Hierarchy Level</b>	[edit chassis cluster]
<b>Release Information</b>	Statement introduced in Junos OS Release 12.1X47-D10.
<b>Description</b>	Disables the automatic chassis cluster synchronization between the primary and secondary nodes. To reenble automatic chassis cluster synchronization, use the <b>delete chassis cluster configuration-synchronize no-secondary-bootup-auto</b> command in configuration mode.
<b>Options</b>	<b>no-secondary-bootup-auto</b> —Disable the automatic chassis cluster synchronization between the primary and secondary nodes.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 179</a></li> <li>• <a href="#">request chassis cluster configuration-synchronize on page 342</a></li> <li>• <a href="#">show chassis cluster information configuration-synchronization on page 367</a></li> </ul>

## connectivity-association

---

**Supported Platforms** SRX340, SRX345

**Syntax** connectivity-association *connectivity-association-name*;  
 exclude-protocol *protocol-name*;  
 include-sci;  
 mka {  
   must-secure;  
   key-server-priority *priority-number*;  
   transmit-interval *interval*;  
 }  
 no-encryption;  
 offset (0|30|50);  
 pre-shared-key {  
   cak *hexadecimal-number*;  
   ckn *hexadecimal-number*;  
 }  
 replay-protect {  
   replay-window-size *number-of-packets*;  
 }  
 security-mode *security-mode*;  
 }

**Hierarchy Level** [edit security macsec]

**Release Information** Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Create or configure a MACsec connectivity association.

A connectivity association is not applying MACsec to traffic until it is associated with an interface. MACsec connectivity associations are associated with interfaces using the **interfaces** statement in the [edit security macsec] hierarchy.

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

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)

## connectivity-association (MACsec Interfaces)

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	connectivity-association <i>connectivity-association-name</i> ;
<b>Hierarchy Level</b>	[edit security macsec cluster-data-port/cluster-control-port]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	Applies a connectivity association to an interface, which enables Media Access Control Security (MACsec) on that interface.
<b>Default</b>	No connectivity associations are associated with any interfaces.
<b>Required Privilege Level</b>	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li> <li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li> <li>• <a href="#">macsec on page 304</a></li> </ul>

## control-link-recovery

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	control-link-recovery;
<b>Hierarchy Level</b>	[edit chassis cluster]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.5.
<b>Description</b>	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.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">interface (Chassis Cluster) on page 297</a></li> </ul>

## device-count (Chassis Cluster)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	device-count <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis aggregated-devices ethernet] [edit chassis aggregated-devices sonnet]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.2.
<b>Description</b>	Configure the number of aggregated logical devices.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">cluster (Chassis) on page 280</a></li><li>• <a href="#">Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI) on page 246</a></li></ul>

---

## no-encryption (MACsec)

---

**Supported Platforms** [SRX340, SRX345](#)

**Syntax** no-encryption;

**Hierarchy Level** [edit security macsec connectivity-association security-mode static-cak]

**Release Information** Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Enable MACsec encryption within a secure channel.

You can enable MACsec without enabling encryption. If a connectivity association with a secure channel that has not enabled MACsec encryption is associated with an interface, traffic is forwarded across the Ethernet link in clear text. You are, therefore, able to view this unencrypted traffic when you are monitoring the link. The MACsec header is still applied to the frame, however, and all MACsec data integrity checks are run on both ends of the link to ensure the traffic has not been tampered with and does not represent a security threat.

Traffic traversing a MAC-enabled point-to-point Ethernet link traverses the link at the same speed regardless of whether encryption is enabled or disabled. You cannot increase the speed of traffic traversing a MACsec-enabled Ethernet link by disabling encryption.

When MACsec is configuring using static connectivity association key (CAK) security mode, the encryption setting is configured outside of the secure channel using the **no-encryption** configuration statement.

**Default** MACsec encryption is disabled when MACsec is configured, by default.

**Required Privilege Level** admin—To view this statement in the configuration.

admin-control—To add this statement to the configuration.

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)

## exclude-protocol

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	exclude-protocol <i>protocol-name</i> ;
<b>Hierarchy Level</b>	[edit security macsec connectivity-association]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies protocols whose packets are not secured using Media Access Control Security (MACsec) when MACsec is enabled on a link using static connectivity association key (CAK) security mode.</p> <p>When this option is enabled in a connectivity association that is attached to an interface, MACsec is not enabled for all packets of the specified protocols that are sent and received on the link.</p>
<b>Default</b>	Disabled.
	All packets are secured on a link when MACsec is enabled, with the exception of all types of Spanning Tree Protocol (STP) packets.
<b>Options</b>	<p><i>protocol-name</i> —Specifies the name of the protocol that should not be MACsec-secured.</p> <p>Options include:</p> <ul style="list-style-type: none"><li>• <b>cdp</b> —Cisco Discovery Protocol.</li><li>• <b>lACP</b> —Link Aggregation Control Protocol.</li><li>• <b>lldp</b> —Link Level Discovery Protocol.</li></ul>
<b>Required Privilege Level</b>	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li><li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li><li>• <a href="#">macsec on page 304</a></li></ul>

## 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 280](#)
- [Example: Configuring Aggregated Ethernet Device with LAG and LACP \(CLI\) on page 246](#)

## 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** `interface`—To view this statement in the configuration.

**Level** `interface-control`—To add this statement to the configuration.

- Related Documentation**
- [Example: Configuring the Chassis Cluster Fabric Interfaces on page 59](#)
  - [member-interfaces on page 305](#)

## 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](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 77](#)

## global-threshold

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	global-threshold <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group <i>group-number</i> ip-monitoring ]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.1.
<b>Description</b>	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.
<b>Options</b>	<i>number</i> —Value at which the IP monitoring weight is applied against the redundancy group failover threshold. <b>Range:</b> 0 through 255 <b>Default:</b> 0
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">ip-monitoring on page 300</a></li></ul>

## global-weight

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	global-weight <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group <i>group-number</i> ip-monitoring ]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.1.
<b>Description</b>	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.
<b>Options</b>	<i>number</i> —Combined weight assigned to all monitored IP addresses. A higher weight value indicates a greater importance. <b>Range:</b> 0 through 255 <b>Default:</b> 255
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">ip-monitoring on page 300</a></li></ul>

## gratuitous-arp-count

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	gratuitous-arp-count <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group group-number]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	Specify the number of gratuitous Address Resolution Protocol (ARP) requests to send on an active interface after failover.
<b>Options</b>	<i>number</i> —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. <b>Range:</b> 1 through 16 <b>Default:</b> 4
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">redundancy-group (Chassis Cluster) on page 313</a></li></ul>

## 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 280](#)

## heartbeat-threshold

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	heartbeat-threshold <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0. Statement updated in Junos OS Release 10.4.
<b>Description</b>	<p>Set the number of consecutive missed heartbeat signals that a device in a chassis cluster must exceed to trigger failover of the active node.</p> <p>The <b>heartbeat-threshold</b> option works in combination with the <b>heartbeat-interval</b> 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.</p> <p>For example, a <b>heartbeat-threshold</b> of 3 and a <b>heartbeat-interval</b> 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 <b>heartbeat-threshold</b>, the <b>heartbeat-interval</b>, or both. A <b>heartbeat-threshold</b> of 5 and a <b>heartbeat-interval</b> of 1000 milliseconds would yield a wait time of 5 seconds. Setting the <b>heartbeat-threshold</b> to 4 and the <b>heartbeat-interval</b> to 1250 milliseconds would also yield a wait time of 5 seconds.</p>
<b>Options</b>	<p><i>number</i> —Number of consecutive missed heartbeats.</p> <p><b>Range:</b> 3 through 8</p> <p><b>Default:</b> 3</p>
<b>Required Privilege Level</b>	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">cluster (Chassis) on page 280</a></li></ul>

## hold-down-interval

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	hold-down-interval <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group <i>group-number</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.0.
<b>Description</b>	<p>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).</p> <p>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.</p> <p>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.</p>
<b>Options</b>	<p><i>number</i>—Number of seconds specified for the interval.</p> <p><b>Range:</b> For redundancy group 0, 300 through 1800 seconds; for redundancy group 1 through 128, 0 through 1800 seconds.</p> <p><b>Default:</b> For redundancy group 0, 300 seconds; for redundancy group 1 through 128, 1 second.</p>
<b>Required Privilege Level</b>	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">cluster (Chassis) on page 280</a></li> </ul>

## include-sci

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	include-sci;
<b>Hierarchy Level</b>	[edit security macsec connectivity-association]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies that the SCI tag should be appended to each packet on a link that has enabled MACsec.</p> <p>You must enable SCI tagging on a switch that is enabling MACsec on an Ethernet link connecting to an SRX device.</p> <p>SCI tags are automatically appended to packets leaving a MACsec-enabled interface on an SRX device. This option is, therefore, not available on an SRX device.</p> <p>You should only use this option when connecting a switch to an SRX device, or to a host device that requires SCI tagging. SCI tags are eight octets long, so appending an SCI tag to all traffic on the link adds a significant amount of unneeded overhead.</p>
<b>Default</b>	<p>SCI tagging is enabled on an SRX device that have enabled MACsec using static connectivity association key (CAK) security mode, by default.</p> <p>SCI tagging is disabled on all other interfaces, by default.</p>
<b>Required Privilege Level</b>	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li><li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li><li>• <a href="#">macsec on page 304</a></li></ul>

## 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**

- **rethX.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 280](#)

## interfaces (MACsec)

---

**Supported Platforms** [SRX340, SRX345](#)

**Syntax**

```
interfaces interface-name {  
    connectivity-association connectivity-association-name;  
}
```

**Hierarchy Level** [edit security macsec]

**Release Information** Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Applies the specified connectivity association to the specified interface to enable MACsec.

One connectivity association can be applied to multiple interfaces.

You must always use this statement to apply a connectivity association to an interface to enable MACsec. You must complete this configuration step regardless of whether MACsec is enabled using static connectivity association key (CAK) security mode.

**Default** Interfaces are not associated with any connectivity associations, by default.

**Required Privilege** admin—To view this statement in the configuration.

**Level** admin-control—To add this statement to the configuration.

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)

## 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 280](#)

## 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 290](#)
- [global-weight on page 291](#)

- [weight](#)
- [Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 134](#)

## key-server-priority (MACsec)

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	<code>key-server-priority <i>priority-number</i>;</code>
<b>Hierarchy Level</b>	[edit security macsec connectivity-association mka]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies the key server priority used by the MACsec Key Agreement (MKA) protocol to select the key server when MACsec is enabled using static connectivity association key (CAK) security mode.</p> <p>The switch with the lower <i>priority-number</i> is selected as the key server.</p> <p>If the <i>priority-number</i> is identical on both sides of a point-to-point link, the MKA protocol selects the device with the lower MAC address as the key server.</p>
<b>Default</b>	The default key server priority number is 16.
<b>Options</b>	<p><b><i>priority-number</i></b> —Specifies the MKA server election priority number.</p> <p>The <i>priority-number</i> can be any number between 0 and 255. The lower the number, the higher the priority.</p>
<b>Required Privilege Level</b>	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li> <li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li> <li>• <a href="#">macsec on page 304</a></li> </ul>

## 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)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	link-protection { non-revertive; }
<b>Hierarchy Level</b>	[edit chassis aggregated-devices ethernet lacp]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.2.
<b>Description</b>	Enable Link Aggregation Control Protocol (LACP) link protection at the global (chassis) level.
<b>Options</b>	<b>non-revertive</b> —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.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">cluster (Chassis) on page 280</a></li><li>• <a href="#">Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI) on page 246</a></li></ul>

## macsec

**Supported Platforms** SRX340, SRX345

**Syntax**

```
macsec {
  connectivity-association connectivity-association-name {
    exclude-protocol protocol-name;
    include-sci;
    mka {
      must-secure;
      key-server-priority priority-number;
      transmit-interval interval;
    }
    no-encryption;
    offset (0|30|50);
    pre-shared-key {
      cak hexadecimal-number;
      ckn hexadecimal-number;
    }
    replay-protect {
      replay-window-size number-of-packets;
    }
    security-mode security-mode;
  }
  interfaces interface-name {
    connectivity-association connectivity-association-name;
  }
}
```

**Hierarchy Level** [edit security]

**Release Information** Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Configure Media Access Control Security (MACsec).

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

**Related Documentation**

- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)

## member-interfaces

---

<b>Supported Platforms</b>	<a href="#">SRX Series</a> , <a href="#">vSRX</a>
<b>Syntax</b>	<code>member-interfaces <i>member-interface-name</i>;</code>
<b>Hierarchy Level</b>	[edit interfaces <i>interface-name</i> fabric-options]
<b>Release Information</b>	Statement introduced in Junos OS Release 8.5.
<b>Description</b>	Specify the member interface name. Member interfaces that connect to each other must be of the same type.
<b>Options</b>	<i>member-interface-name</i> —Member interface name.
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Interfaces</a></li> </ul>

## mka

---

<b>Supported Platforms</b>	<a href="#">SRX340</a> , <a href="#">SRX345</a>
<b>Syntax</b>	<pre>mka {   must-secure;   key-server-priority <i>priority-number</i>;   transmit-interval <i>interval</i>; }</pre>
<b>Hierarchy Level</b>	[edit security macsec connectivity-association]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	Specify parameters for the MACsec Key Agreement (MKA) protocol.
<b>Options</b>	The remaining statements are explained separately.
<b>Required Privilege Level</b>	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li> <li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li> <li>• <a href="#">macsec on page 304</a></li> </ul>

## must-secure

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	must-secure;
<b>Hierarchy Level</b>	[edit security macsec connectivity-association mka]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies that all traffic travelling on the MACsec-secured link must be MACsec-secured to be forwarded onward.</p> <p>When the <b>must-secure</b> option is enabled, all traffic that is not MACsec-secured that is received on the interface is dropped.</p> <p>When the <b>must-secure</b> option is disabled, all traffic from devices that support MACsec is MACsec-secured while traffic received from devices that do not support MACsec is forwarded through the network.</p> <p>The <b>must-secure</b> option is particularly useful in scenarios where multiple devices, such as a phone and a PC, are accessing the network through the same Ethernet interface. If one of the devices supports MACsec while the other device does not support MACsec, the device that doesn't support MACsec can continue to send and receive traffic over the network—provided the <b>must-secure</b> option is disabled—while traffic to and from the device that supports MACsec is MACsec-secured. In this scenario, traffic to the device that is not MACsec-secured must be VLAN-tagged.</p>
<b>Default</b>	The <b>must-secure</b> option is disabled.
<b>Required Privilege Level</b>	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li><li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li><li>• <a href="#">macsec on page 304</a></li></ul>

## network-management

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	<pre>network-management {   cluster-master; }</pre>
<b>Hierarchy Level</b>	[edit chassis cluster]
<b>Release Information</b>	Statement introduced in Junos OS Release 11.1.
<b>Description</b>	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 <code>&lt;get-system-information&gt;</code> RPC returns a <code>&lt;chassis-cluster&gt;</code> tag in all SRX Series devices. When NSM receives this tag, it models SRX Series clusters as devices with autonomous control planes.
<b>Options</b>	<code>cluster-master</code> —Enable in-band management on the primary cluster node through NSM.
<b>Required Privilege Level</b>	<code>interface</code> —To view this statement in the configuration. <code>interface-control</code> —To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">cluster (Chassis) on page 280</a></li></ul>

## node (Chassis Cluster)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	node (0   1) { priority <i>number</i> ; }
<b>Hierarchy Level</b>	[edit chassis cluster]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	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.
<b>Options</b>	<i>node-number</i> —Cluster node number. <b>Range:</b> 0 through 1
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">cluster (Chassis) on page 280</a></li> </ul>

## node (Chassis Cluster Redundancy Group)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	node (0   1) { priority <i>number</i> ; }
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group <i>group-number</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	Identify each cluster node in a redundancy group and set its relative priority for mastership.
<b>Options</b>	<i>node-number</i> —Cluster node number, set with the chassis cluster node <i>node-number</i> statement.  The remaining statements are explained separately. See <a href="#">CLI Explorer</a> .
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">redundancy-group (Chassis Cluster) on page 313</a></li> </ul>

## offset

**Supported Platforms** SRX340, SRX345

**Syntax** offset (0 |30 | 50);

**Hierarchy Level** [edit security macsec connectivity-association]  
[edit security macsec connectivity-association security-mode static-cakl]

**Release Information** Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Specifies the number of octets in an Ethernet frame that are sent in unencrypted plain-text when encryption is enabled for MACsec.

Setting the offset to 30 allows a feature to see the IPv4 header and the TCP/UDP header while encrypting the remaining traffic. Setting the offset to 50 allows a feature to see the IPv6 header and the TCP/UDP header while encrypting the remaining traffic.

You would typically forward traffic with the first 30 or 50 octets unencrypted if a feature needed to see the data in the octets to perform a function, but you otherwise prefer to encrypt the remaining data in the frames traversing the link. Load balancing features, in particular, typically need to see the IP and TCP/UDP headers in the first 30 or 50 octets to properly load balance traffic.

You configure the **offset** in the [edit security macsec connectivity-association] hierarchy when you are enabling MACsec using static connectivity association key (CAK) or dynamic security mode.

**Default** 0

**Options** **0**—Specifies that no octets are unencrypted. When you set the offset to 0, all traffic on the interface where the connectivity association or secure channel is applied is encrypted.

**30**—Specifies that the first 30 octets of each Ethernet frame are unencrypted.



**NOTE:** In IPv4 traffic, setting the offset to 30 allows a feature to see the IPv4 header and the TCP/UDP header while encrypting the rest of the traffic. An offset of 30, therefore, is typically used when a feature needs this information to perform a task on IPv4 traffic.

**50**—Specified that the first 50 octets of each Ethernet frame are unencrypted.



**NOTE:** In IPv6 traffic, setting the offset to 50 allows a feature to see the IPv6 header and the TCP/UDP header while encrypting the rest of the traffic. An offset of 50, therefore, is typically used when a feature needs this information to perform a task on IPv6 traffic.

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

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)

## 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 313](#)

## pre-shared-key

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	<pre>pre-shared-key {   cak hexadecimal-number;   ckn hexadecimal-number; }</pre>
<b>Hierarchy Level</b>	[edit security macsec connectivity-association]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies the pre-shared key used to enable MACsec using static connectivity association key (CAK) security mode.</p> <p>A pre-shared key includes a connectivity association key name (CKN) and a connectivity association key (CAK). A pre-shared key is exchanged between two devices at each end of a point-to-point link to enable MACsec using static CAK security mode. The MACsec Key Agreement (MKA) protocol is enabled after the pre-shared keys are successfully verified and exchanged. The pre-shared key—the CKN and CAK—must match on both ends of a link.</p>
<b>Default</b>	No pre-shared keys exist, by default.
<b>Options</b>	The remaining statements are explained separately.
<b>Required Privilege Level</b>	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li> <li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li> <li>• <a href="#">macsec on page 304</a></li> </ul>

## priority (Chassis Cluster)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	priority <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group <i>group-number</i> node <i>node-number</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	Define the priority of a node (device) in a redundancy group. Initiating a failover with the <b>request chassis cluster failover node</b> or <b>request chassis cluster failover redundancy-group</b> command overrides the priority settings.
<b>Options</b>	<i>number</i> —Priority value of the node. The eligible node with the highest priority is elected master. <b>Range:</b> 1 through 254
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">redundancy-group (Chassis Cluster) on page 313</a></li></ul>

## 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 300](#)

## 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 280](#)

## 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](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 77](#)

## redundant-parent (Interfaces)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	redundant-parent <i>redundant-ethernet-interface-name</i> ;
<b>Hierarchy Level</b>	[edit interfaces <i>interface-name</i> gigheter-options] [edit interfaces <i>interface-name</i> fastether-options]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.2.
<b>Description</b>	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).
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Example: Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv4 and IPv6 Addresses on page 77</a></li> </ul>

## redundant-pseudo-interface-options

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	redundant-pseudo-interface-options { redundancy-group <i>redundancy-group-number</i> ; }
<b>Hierarchy Level</b>	[edit interfaces lo0]
<b>Release Information</b>	Statement introduced in Junos OS Release 12.1X44-D10.
<b>Description</b>	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.
<b>Options</b>	<i>redundancy-group-number</i> — Configure the redundancy group number. <b>Range:</b> 0 through 255
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Loopback Interface for a High Availability VPN</a></li> </ul>

## replay-protect

---

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	<pre>replay-protect {   replay-window-size <i>number-of-packets</i>; }</pre>
<b>Hierarchy Level</b>	[edit security macsec connectivity-association]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Enable replay protection for MACsec.</p> <p>A replay window size specified using the <b>replay-window-size</b><i>number-of-packets</i> statement must be specified to enable replay protection.</p>
<b>Options</b>	The remaining statements are explained separately.
<b>Required Privilege Level</b>	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li><li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li><li>• <a href="#">macsec on page 304</a></li></ul>

## replay-window-size

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	replay-window-size <i>number-of-packets</i> ;
<b>Hierarchy Level</b>	[edit security macsec connectivity-association replay-protect]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Specifies the size of the replay protection window.</p> <p>This statement has to be configured to enable replay protection.</p> <p>When MACsec is enabled on an Ethernet link, an ID number is assigned to each packet entering the link. The ID number of the packet is checked by the receiving interface after the packet has traversed the MACsec-enabled link.</p> <p>When replay protection is enabled, the sequence of the ID number of received packets are checked. If the packet arrives out of sequence and the difference between the packet numbers exceeds the replay protection window size, the packet is dropped by the receiving interface. For instance, if the replay protection window size is set to five and a packet assigned the ID of 1006 arrives on the receiving link immediately after the packet assigned the ID of 1000, the packet that is assigned the ID of 1006 is dropped because it falls outside the parameters of the replay protection window.</p> <p>Replay protection is especially useful for fighting man-in-the-middle attacks. A packet that is replayed by a man-in-the-middle attacker on the Ethernet link will arrive on the receiving link out of sequence, so replay protection helps ensure the replayed packet is dropped instead of forwarded through the network.</p> <p>Replay protection should not be enabled in cases where packets are expected to arrive out of order.</p>
<b>Default</b>	Replay protection is disabled.
<b>Options</b>	<p><i>number-of-packets</i> —Specifies the size of the replay protection window, in packets.</p> <p>When this variable is set to 0, all packets that arrive out-of-order are dropped.</p>
<b>Required Privilege Level</b>	admin—To view this statement in the configuration. admin-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li><li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li><li>• <a href="#">macsec on page 304</a></li></ul>

## reth-count (Chassis Cluster)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	reth-count <i>number</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	Specify the number of redundant Ethernet ( <b>reth</b> ) interfaces allowed in the chassis cluster. Note that the number of <b>reth</b> interfaces configured determines the number of redundancy groups that can be configured.
<b>Options</b>	<i>number</i> —Number of redundant Ethernet interfaces allowed. <b>Range:</b> 1 through 128 <b>Default:</b> 0
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">cluster (Chassis) on page 280</a></li></ul>

## 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-timesseconds> | 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;
      output;
    }
  }
}
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 77](#)
- [cluster \(Chassis\) on page 280](#)
- [redundant-ether-options on page 315](#)
- [lACP \(Interfaces\) on page 302](#)

## 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 280](#)

## retry-interval (Chassis Cluster)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	retry-interval <i>interval</i> ;
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group <i>group-number</i> ip-monitoring ]
<b>Release Information</b>	Statement introduced in Junos OS Release 10.1.
<b>Description</b>	Specify the ping packet send frequency (in seconds) for each IP address monitored by the redundancy group. (See <b>retry-count</b> for a related IP address monitoring configuration variable.)
<b>Options</b>	<i>interval</i> —Pause time between each ping sent to each IP address monitored by the redundancy group. <b>Range:</b> 1 to 30 seconds <b>Default:</b> 1 second
<b>Required Privilege Level</b>	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">ip-monitoring on page 300</a></li></ul>

## route-active-on

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	route-active-on (node0   node1);
<b>Hierarchy Level</b>	[edit policy-options condition <i>condition-name</i> ]
<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	For chassis cluster configurations, identify the device (node) on which a route is active.
<b>Options</b>	node0   node1—Node in a chassis cluster.
<b>Required Privilege Level</b>	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">cluster (Chassis) on page 280</a></li></ul>

## security-mode

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	security-mode <i>security-mode</i> ;
<b>Hierarchy Level</b>	[edit security macsec connectivity-association]
<b>Release Information</b>	Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	<p>Configure the MACsec security mode for the connectivity association.</p> <p>We recommend enabling MACsec on switch-to-switch Ethernet links using static connectivity association key (CAK) security mode. Static CAK security mode ensures security by frequently refreshing to a new random secure association key (SAK) and by only sharing the SAK between the two devices on the MACsec-secured point-to-point link. Additionally, some optional MACsec features—replay protection, SCI tagging, and the ability to exclude traffic from MACsec—are only available when you enable MACsec using static CAK security mode.</p>
<b>Options</b>	<p><i>security-mode</i> —Specifies the MACsec security mode. Options include:</p> <ul style="list-style-type: none"> <li>• <b>dynamic</b>—Dynamic mode.</li> </ul> <p>Dynamic security mode is used to enable MACsec on switch-to-host Ethernet links. In dynamic mode, a master key is retrieved from a RADIUS server by a switch and a host as part of the AAA handshake in separate transactions. The MKA protocol is enabled when the master key is exchanged between the switch and the host.</p> <ul style="list-style-type: none"> <li>• <b>static-cak</b> —Static connectivity association key (CAK) mode.</li> </ul> <p>Static CAK security mode is used to enable MACsec on switch-to-switch Ethernet links. In <b>static-cak</b> mode, the switch at one end of the point-to-point link acts as the key server and regularly transmits a randomized key using a process that does not transmit any traffic outside of the MACsec-secured point-to-point link.</p>
<b>Required Privilege Level</b>	<p>admin—To view this statement in the configuration.</p> <p>admin-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li> <li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li> <li>• <a href="#">macsec on page 304</a></li> </ul>

## 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 280](#)

## transmit-interval (MACsec)

---

**Supported Platforms** [SRX340, SRX345](#)

**Syntax** `transmit-interval interval;`

**Hierarchy Level** [edit security macsec connectivity-association mka]

**Release Information** Statement introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Specifies the transmit interval for MACsec Key Agreement (MKA) protocol data units (PDUs).

The MKA transmit interval setting sets the frequency for how often the MKA PDU is sent to the directly connected device to maintain MACsec on a point-to-point Ethernet link. A lower *interval* increases bandwidth overhead on the link; a higher *interval* optimizes the MKA protocol data unit exchange process.

The transmit interval settings must be identical on both ends of the link when MACsec using static connectivity association key (CAK) security mode is enabled.

We recommend increasing the interval to 6000 ms in high-traffic load environments.

**Default** The default transmit interval is 10000 milliseconds (10 seconds).



**NOTE:** Configuring aggressive transmit interval will lead to broken chassis cluster.

---

**Options** *interval* —Specifies the transmit interval, in milliseconds.

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

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)

## weight

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	<code>weight number;</code>
<b>Hierarchy Level</b>	[edit chassis cluster redundancy-group <i>group-number</i> interface-monitor <i>interface</i> ] [edit chassis cluster redundancy-group <i>group-number</i> ip-monitoring <i>IP-address</i> ]
<b>Release Information</b>	Statement modified in Junos OS Release 10.1.
<b>Description</b>	<p>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.</p> <ul style="list-style-type: none"> <li>interface-monitor objects—If the object fails, its weight is deducted from the threshold of its redundancy group;</li> <li>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.</li> </ul> <p>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.</p>
<b>Options</b>	<p><i>number</i> —Weight assigned to the interface or monitored IP address. A higher weight value indicates a greater importance.</p> <p><b>Range:</b> 0 through 255</p>
<b>Required Privilege Level</b>	<p>interface—To view this statement in the configuration.</p> <p>interface-control—To add this statement to the configuration.</p>
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><a href="#">cluster (Chassis) on page 280</a></li> </ul>



## CHAPTER 26

# Operational Commands

- clear chassis cluster control-plane statistics
- clear chassis cluster data-plane statistics
- clear chassis cluster failover-count
- clear chassis cluster ip-monitoring failure-count
- clear chassis cluster ip-monitoring failure-count ip-address
- clear chassis cluster statistics
- request chassis cluster configuration-synchronize
- request chassis cluster failover node
- request chassis cluster failover redundancy-group
- request chassis cluster failover reset
- request chassis fpc
- request system reboot
- request system software in-service-upgrade (Maintenance)
- request system software rollback (SRX Series)
- set chassis cluster cluster-id node reboot
- show chassis cluster control-plane statistics
- show chassis cluster data-plane interfaces
- show chassis cluster data-plane statistics
- show chassis cluster ethernet-switching interfaces
- show chassis cluster ethernet-switching status
- show chassis cluster information
- show chassis cluster information configuration-synchronization
- show chassis cluster interfaces
- show chassis cluster ip-monitoring status redundancy-group
- show chassis cluster statistics
- show chassis cluster status
- show chassis environment (Security)
- 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 interfaces (Gigabit Ethernet) SRX device`
- `show security macsec connections`
- `show security macsec statistics for SRX device`
- `show security mka statistics`
- `show security mka sessions for SRX device`

---

## clear chassis cluster control-plane statistics

---

<b>Supported Platforms</b>	<a href="#">SRX Series, vSRX</a>
<b>Syntax</b>	<code>clear chassis cluster control-plane statistics</code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.3.
<b>Description</b>	Clear the control plane statistics of a chassis cluster.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">show chassis cluster control-plane statistics on page 355</a></li></ul>
<b>List of Sample Output</b>	<a href="#">clear chassis cluster control-plane statistics on page 335</a>
<b>Output Fields</b>	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

---

<b>Supported Platforms</b>	<a href="#">SRX Series, vSRX</a>
<b>Syntax</b>	<code>clear chassis cluster data-plane statistics</code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.3.
<b>Description</b>	Clear the data plane statistics of a chassis cluster.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">show chassis cluster data-plane statistics on page 358</a></li></ul>
<b>List of Sample Output</b>	<a href="#">clear chassis cluster data-plane statistics on page 336</a>
<b>Output Fields</b>	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 343](#)
- [request chassis cluster failover reset on page 345](#)
- [show chassis cluster status on page 381](#)

**List of Sample Output** [show chassis cluster status on page 337](#)  
[clear chassis cluster failover-count on page 337](#)  
[show chassis cluster status on page 337](#)

**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 340](#)

**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 337](#)
- [clear chassis cluster ip-monitoring failure-count on page 339](#)

**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

---

<b>Supported Platforms</b>	<a href="#">SRX Series, vSRX</a>
<b>Syntax</b>	<code>clear chassis cluster statistics</code>
<b>Release Information</b>	Command introduced in Junos OS Release 9.3.
<b>Description</b>	Clear the control plane and data plane statistics of a chassis cluster.
<b>Required Privilege Level</b>	clear
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">show chassis cluster statistics on page 377</a></li></ul>
<b>List of Sample Output</b>	<a href="#">clear chassis cluster statistics on page 341</a>
<b>Output Fields</b>	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 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 179](#)
- [Verifying Chassis Cluster Configuration Synchronization Status on page 180](#)
- [NTP Time Synchronization on SRX Series Devices on page 181](#)

**List of Sample Output** [request chassis cluster configuration-synchronize on page 342](#)

**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 node

**Supported Platforms** [SRX Series, vSRX](#)

**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 337](#)
  - [request chassis cluster failover reset on page 345](#)
  - [show chassis cluster status on page 381](#)

**List of Sample Output** [request chassis cluster failover node 0 redundancy-group 1 on page 343](#)

**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 redundancy-group

### Supported Platforms

**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 144](#)
  - [Verifying Chassis Cluster Failover Status on page 146](#)

**List of Sample Output** [request chassis cluster failover redundancy-group 0 node 1 on page 344](#)

**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 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 337](#)
- [request chassis cluster failover node on page 343](#)
- [show chassis cluster status on page 381](#)

**List of Sample Output** [request chassis cluster failover reset redundancy-group 0 on page 345](#)

**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 fpc

---

**Supported Platforms** [SRX Series](#)

**Syntax** `request chassis fpc (offline | online | restart) slot slot-number`

**Release Information** Command modified in Junos OS Release 9.2.

**Description** Control the operation of the Flexible PIC Concentrator (FPC).



**NOTE:** The SRX5K-SPC-2-10-40 (SPC1) and SRX5K-SPC-4-15-320 (SPC2) does not support the `request chassis fpc` command.

**Options** **offline**—Take the FPC offline.

**online**—Bring the FPC online.

**restart**—Restart the FPC.

**slot *slot-number***—Specify the FPC slot number.

**Required Privilege Level** maintenance

**Related Documentation**

- [show chassis fpc \(View\)](#)

**Output Fields** When you enter this command, you are provided feedback on the status of your request.

### Sample Output

#### request chassis fpc

```
user@host> request chassis fpc online slot 0
FPC 0 already online
```

## request system reboot

**Supported Platforms** [SRX Series, vSRX](#)

**Syntax** `request system reboot <at time> <in minutes><media><message 'text'>`

**Release Information** Command introduced in Junos OS Release 10.1.  
Command **hypervisor** option introduced in Junos OS Release 15.1X49-D10 for vSRX.  
Command introduced in Junos OS Release 15.1X49-D50 for SRX1500 devices.

**Description** Reboot the software.

- Options**
- *at time*— Specifies the time at which to reboot the device . You can specify time in one of the following ways:
    - *now*— Reboots the device immediately. This is the default.
    - *+minutes*— Reboots the device in the number of minutes from now that you specify.
    - *yymmddhhmm*— Reboots the device at the absolute time on the date you specify. Enter the year, month, day, hour (in 24-hour format), and minute.
    - *hh:mm*— Reboots the device at the absolute time you specify, on the current day. Enter the time in 24-hour format, using a colon (:) to separate hours from minutes.
  - *in minutes*— Specifies the number of minutes from now to reboot the device. This option is a synonym for the *at +minutes* option
  - *media type*— Specifies the boot device to boot the device from:
    - *disk/internal*— Reboots from the internal media. This is the default.
    - *usb*— Reboots from the USB storage device.
    - *compact flash*— Reboots from the external CompactFlash card.



**NOTE:** The **media** command option is not available on vSRX.

- *message text*— Provides a message to display to all system users before the device reboots.

Example: `request system reboot at 5 in 50 media internal message stop`

**Required Privilege Level** maintenance

**Related Documentation**

- [request system software rollback \(SRX Series\) on page 353](#)

## 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 353](#)

List of Sample Output

[request system software in-service-upgrade \(High-End SRX Series Devices\) on page 349](#)  
[request system software in-service-upgrade \(SRX300, SRX320, SRX340, SRX345, and SRX550M devices\) on page 350](#)

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
Initiating in-service-upgrade

node0:
-----
Initiating 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)

---

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	request system software rollback <node-id>
<b>Release Information</b>	Command introduced in Junos OS Release 10.1. Command introduced in Junos OS Release 15.1X49-D50 for SRX1500 devices.
<b>Description</b>	Revert to the software that was loaded at the last successful <b>request system software add</b> command. Example: <b>request system software rollback</b> .
<b>Options</b>	<i>node-id</i> —Identification number of the chassis cluster node. It can be 0 or 1.
<b>Required Privilege Level</b>	maintenance
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">request system reboot on page 347</a></li></ul>

## 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 on page 49](#)
  - [Example: Setting the Chassis Cluster Node ID and Cluster ID for High-End SRX Series Devices](#)
  - [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

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis cluster control-plane statistics
<b>Release Information</b>	Command introduced in Junos OS Release 9.3. Output changed to support dual control ports in Junos OS Release 10.0.
<b>Description</b>	Display information about chassis cluster control plane statistics.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">clear chassis cluster control-plane statistics on page 335</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis cluster control-plane statistics on page 356</a> <a href="#">show chassis cluster control-plane statistics (SRX5000 line devices) on page 356</a>
<b>Output Fields</b>	Table 27 on page 355 lists the output fields for the <b>show chassis cluster control-plane statistics</b> command. Output fields are listed in the approximate order in which they appear.

Table 27: show chassis cluster control-plane statistics Output Fields

Field Name	Field Description
Control link statistics	<p>Statistics of the control link used by chassis cluster traffic. Statistics for <b>Control link 1</b> are displayed when you use dual control links (SRX5600 and SRX5800 devices only).</p> <ul style="list-style-type: none"> <li>• <b>Heartbeat packets sent</b>—Number of heartbeat messages sent on the control link.</li> <li>• <b>Heartbeat packets received</b>—Number of heartbeat messages received on the control link.</li> <li>• <b>Heartbeat packet errors</b>—Number of heartbeat packets received with errors on the control link.</li> </ul>
Fabric link statistics	<p>Statistics of the fabric link used by chassis cluster traffic. Statistics for <b>Child Link 1</b> are displayed when you use dual fabric links.</p> <ul style="list-style-type: none"> <li>• <b>Probes sent</b>—Number of probes sent on the fabric link.</li> <li>• <b>Probes received</b>—Number of probes received on the fabric link.</li> </ul>
Switch fabric link statistics	<p>Statistics of the switch fabric link used by chassis cluster traffic.</p> <ul style="list-style-type: none"> <li>• <b>Probe state</b>—State of the probe, <b>UP</b> or <b>DOWN</b>.</li> <li>• <b>Probes sent</b>—Number of probes sent.</li> <li>• <b>Probes received</b>—Number of probes received.</li> <li>• <b>Probe rcv error</b>—Error in receiving probe.</li> <li>• <b>Probe send error</b>—Error in sending probe.</li> </ul>

## 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

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis cluster data-plane interfaces
<b>Release Information</b>	Command introduced in Junos OS Release 10.2.
<b>Description</b>	Display the status of the data plane interface (also known as a fabric interface) in a chassis cluster configuration.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">cluster (Chassis) on page 280</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis cluster data-plane interfaces on page 357</a>
<b>Output Fields</b>	Table 28 on page 357 lists the output fields for the <b>show chassis cluster data-plane interfaces</b> command. Output fields are listed in the approximate order in which they appear.

Table 28: show chassis cluster data-plane interfaces Output Fields

Field Name	Field Description
fab0/fab1	Name of the logical fabric interface. <ul style="list-style-type: none"> <li>• <b>Name</b>—Name of the physical Ethernet interface.</li> <li>• <b>Status</b>—State of the fabric interface: <b>up</b> or <b>down</b>.</li> </ul>

## 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

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis cluster data-plane statistics
<b>Release Information</b>	Command introduced in Junos OS Release 9.3.
<b>Description</b>	Display information about chassis cluster data plane statistics.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>clear chassis cluster data-plane statistics on page 336</li> </ul>
<b>List of Sample Output</b>	show chassis cluster data-plane statistics on page 359
<b>Output Fields</b>	Table 29 on page 358 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 29: show chassis cluster data-plane statistics Output Fields**

Field Name	Field Description
Services Synchronized	<ul style="list-style-type: none"> <li><b>Service name</b>—Name of the service.</li> <li><b>Rtos sent</b>—Number of runtime objects (RTOs) sent.</li> <li><b>Rtos received</b>—Number of RTOs received.</li> <li><b>Translation context</b>—Messages synchronizing Network Address Translation (NAT) translation context.</li> <li><b>Incoming NAT</b>—Messages synchronizing incoming Network Address Translation (NAT) service.</li> <li><b>Resource manager</b>—Messages synchronizing resource manager groups and resources.</li> <li><b>Session create</b>—Messages synchronizing session creation.</li> <li><b>Session close</b>—Messages synchronizing session close.</li> <li><b>Session change</b>—Messages synchronizing session change.</li> <li><b>Gate create</b>—Messages synchronizing creation of pinholes (temporary openings in the firewall).</li> <li><b>Session ageout refresh request</b>—Messages synchronizing request session after age-out.</li> <li><b>Session ageout refresh reply</b>—Messages synchronizing reply session after age-out.</li> <li><b>IPsec VPN</b>—Messages synchronizing VPN session.</li> <li><b>Firewall user authentication</b>—Messages synchronizing firewall user authentication session.</li> <li><b>MGCP ALG</b>—Messages synchronizing MGCP ALG sessions.</li> <li><b>H323 ALG</b>—Messages synchronizing H.323 ALG sessions.</li> <li><b>SIP ALG</b>—Messages synchronizing SIP ALG sessions.</li> <li><b>SCCP ALG</b>—Messages synchronizing SCCP ALG sessions.</li> <li><b>PPTP ALG</b>—Messages synchronizing PPTP ALG sessions.</li> <li><b>RTSP ALG</b>—Messages synchronizing RTSP ALG sessions.</li> </ul>

## Sample Output

### show chassis cluster data-plane statistics

```
user@host> show chassis cluster data-plane statistics
Services Synchronized:
  Service name                RT0s sent  RT0s received
  Translation context         0          0
  Incoming NAT                0          0
  Resource manager            0          0
  Session create              0          0
  Session close               0          0
  Session change              0          0
  Gate create                  0          0
  Session ageout refresh requests 0          0
  Session ageout refresh replies 0          0
  IPsec VPN                   0          0
  Firewall user authentication 0          0
  MGCP ALG                    0          0
  H323 ALG                    0          0
  SIP ALG                     0          0
  SCCP ALG                    0          0
  PPTP ALG                    0          0
  RTSP ALG                    0          0
```

## 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 280](#)
- *Ethernet Switching and Layer 2 Transparent Mode Feature Guide for Security Devices*

**List of Sample Output** [show chassis cluster ethernet-switching interfaces on page 360](#)

**Output Fields** [Table 30 on page 360](#) 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 30: show chassis cluster ethernet-switching interfaces Output Fields**

Field Name	Field Description
swfab0/swfab1	Name of the switch fabric interface. <ul style="list-style-type: none"> <li>• Name—Name of the physical interface.</li> <li>• Status—State of the swfab interface: up or down.</li> </ul>

## 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

<b>Supported Platforms</b>	SRX1500, SRX550M, vSRX
<b>Syntax</b>	show chassis cluster ethernet-switching status
<b>Release Information</b>	Command introduced in Junos OS Release 11.1.
<b>Description</b>	Display the Ethernet switching status of the chassis cluster.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">cluster (Chassis) on page 280</a></li> <li>• <i>Ethernet Switching and Layer 2 Transparent Mode Feature Guide for Security Devices</i></li> </ul>

**List of Sample Output** [show chassis cluster ethernet-switching status on page 362](#)

**Output Fields** [Table 31 on page 361](#) 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 31: show chassis cluster ethernet-switching status Output Fields**

Field Name	Field Description
<b>Cluster ID</b>	<p>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.</p> <p><b>NOTE:</b> 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.</p> <p><b>NOTE:</b> 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.</p>
<b>Redundancy-Group</b>	ID number (1-255) of a redundancy group in the chassis cluster.
<b>Node name</b>	Node (device) in the chassis cluster ( <b>node0</b> or <b>node1</b> ).
<b>Priority</b>	Assigned priority for the redundancy group on that node.

**Table 31: show chassis cluster ethernet-switching status Output Fields (continued)**

Field Name	Field Description
<b>Status</b>	<p>State of the redundancy group (<b>Primary</b>, <b>Secondary</b>, <b>Lost</b>, or <b>Unavailable</b>).</p> <ul style="list-style-type: none"> <li>• <b>Primary</b>—Redundancy group is active and passing traffic.</li> <li>• <b>Secondary</b>—Redundancy group is passive and not passing traffic.</li> <li>• <b>Lost</b>—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.</li> <li>• <b>Unavailable</b>—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.</li> </ul>
<b>Preempt</b>	<ul style="list-style-type: none"> <li>• <b>Yes</b>: Mastership can be preempted based on priority.</li> <li>• <b>No</b>: Change in priority will not preempt mastership.</li> </ul>
<b>Manual failover</b>	<ul style="list-style-type: none"> <li>• <b>Yes</b>: If mastership is set manually through the CLI.</li> <li>• <b>No</b>: Mastership is not set manually through the CLI.</li> </ul>

## 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 rcv errors: 0
Probe send errors: 0

```

## show chassis cluster information

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis cluster information
<b>Release Information</b>	Command introduced in Junos OS Release 12.1X47-D10.
<b>Description</b>	Display chassis cluster messages. The messages indicate each node's health condition and details of the monitored failure.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show chassis cluster status on page 381</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis cluster information on page 363</a> <a href="#">show chassis cluster information on page 364</a>
<b>Output Fields</b>	Table 32 on page 363 lists the output fields for the <b>show chassis cluster information</b> command. Output fields are listed in the approximate order in which they appear.

Table 32: show chassis cluster information Output Fields

Field Name	Field Description
Node	Node (device) in the chassis cluster ( <b>node0</b> or <b>node1</b> ).
Redundancy Group Information	<ul style="list-style-type: none"> <li>• Redundancy Group—ID number (0 - 255) of a redundancy group in the cluster.</li> <li>• Current State—State of the redundancy group: <b>primary</b>, <b>secondary</b>, <b>hold</b>, or <b>secondary-hold</b>.</li> <li>• Weight—Relative importance of the redundancy group.</li> <li>• Time—Time when the redundancy group changed the state.</li> <li>• From—State of the redundancy group before the change.</li> <li>• To—State of the redundancy group after the change.</li> <li>• Reason—Reason for the change of state of the redundancy group.</li> </ul>
Chassis cluster LED information	<ul style="list-style-type: none"> <li>• Current LED color—Current color state of the LED.</li> <li>• Last LED change reason—Reason for change of state of the LED.</li> </ul>

## 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 179](#)
- [NTP Time Synchronization on SRX Series Devices on page 181](#)
- [Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 181](#)
- [request chassis cluster configuration-synchronize on page 342](#)

**List of Sample Output** [show chassis cluster information configuration-synchronization on page 367](#)

**Output Fields** [Table 33 on page 367](#) 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 33: show chassis cluster information configuration-synchronization Output Fields**

Field Name	Field Description
Node name	Node (device) in the chassis cluster ( <b>node0</b> or <b>node1</b> ).
Status	<ul style="list-style-type: none"> <li>• Activation status—State of automatic configuration synchronization: <b>Enabled</b> or <b>Disabled</b>.</li> <li>• Last sync operation—Status of the last synchronization.</li> <li>• Last sync result—Result of the last synchronization.</li> <li>• Last sync mgd messages—Management daemon messages of the last synchronization.</li> </ul>
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 interfaces

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis cluster interfaces
<b>Release Information</b>	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.
<b>Description</b>	Display the status of the control interface in a chassis cluster configuration.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">cluster (Chassis) on page 280</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis cluster interfaces on page 370</a> <a href="#">show chassis cluster interfaces (SRX5000 line devices) on page 371</a> <a href="#">show chassis cluster interfaces on page 372</a> <a href="#">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 372</a>
<b>Output Fields</b>	Table 34 on page 369 lists the output fields for the <b>show chassis cluster interfaces</b> command. Output fields are listed in the approximate order in which they appear.

Table 34: show chassis cluster interfaces Output Fields

Field Name	Field Description
Control link status	State of the chassis cluster control interface: <b>up</b> or <b>down</b> .
Control interfaces	<ul style="list-style-type: none"> <li>• <b>Index</b>—Index number of the chassis cluster control interface.</li> <li>• <b>Name</b>—Name of the chassis cluster control interface.</li> <li>• <b>Monitored-Status</b>—Monitored state of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Internal SA</b>—State of the internal SA option on the chassis cluster control link: <b>enabled</b> or <b>disabled</b>.</li> </ul> <p><i>NOTE:</i> This field is available only on high-end SRX Series devices.</p> <ul style="list-style-type: none"> <li>• <b>Security</b>—State of MACsec on chassis cluster control interfaces.</li> </ul>
Fabric link status	State of the fabric interface: <b>up</b> or <b>down</b> .

Table 34: show chassis cluster interfaces Output Fields (*continued*)

Field Name	Field Description
<b>Fabric interfaces</b>	<ul style="list-style-type: none"> <li>• <b>Name</b>—Name of the fabric interface.</li> <li>• <b>Child-interface</b>—Name of the child fabric interface.</li> <li>• <b>Status</b>—State of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Security</b>—State of MACsec on chassis cluster fabric interfaces.</li> </ul>
<b>Redundant-ethernet Information</b>	<ul style="list-style-type: none"> <li>• <b>Name</b>—Name of the redundant Ethernet interface.</li> <li>• <b>Status</b>—State of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Redundancy-group</b>—Identification number (1–255) of the redundancy group associated with the redundant Ethernet interface.</li> </ul>
<b>Redundant-pseudo-interface Information</b>	<ul style="list-style-type: none"> <li>• <b>Name</b>—Name of the redundant pseudointerface.</li> <li>• <b>Status</b>—State of the redundant pseudointerface: <b>up</b> or <b>down</b>.</li> <li>• <b>Redundancy-group</b>—Identification number (1–255) of the redundancy group associated with the redundant pseudointerface.</li> </ul>
<b>Interface Monitoring</b>	<ul style="list-style-type: none"> <li>• <b>Interface</b>—Name of the interface to be monitored.</li> <li>• <b>Weight</b>—Relative importance of the interface to redundancy group operation.</li> <li>• <b>Status</b>—State of the interface: <b>up</b> or <b>down</b>.</li> <li>• <b>Redundancy-group</b>—Identification number of the redundancy group associated with the interface.</li> </ul>

## 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          0

```

## 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 (Physical/Monitored)	Security
fab0	xe-1/0/3	Up / Down	Disabled
fab1	xe-7/0/3	Up / Down	Disabled

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          0

```

## 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 (Physical/Monitored)	Security
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 (Physical/Monitored)	Security
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

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	<code>show chassis cluster ip-monitoring status</code> <code>&lt;redundancy-group group-number&gt;</code>
<b>Release Information</b>	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.
<b>Description</b>	Display the status of all monitored IP addresses for a redundancy group.
<b>Options</b>	<ul style="list-style-type: none"> <li><code>none</code>— Display the status of monitored IP addresses for all redundancy groups on the node.</li> <li><code>redundancy-group group-number</code>— Display the status of monitored IP addresses under the specified redundancy group.</li> </ul>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li><code>clear chassis cluster failover-count</code></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis cluster ip-monitoring status on page 375</a> <a href="#">show chassis cluster ip-monitoring status redundancy-group on page 376</a>
<b>Output Fields</b>	Table 35 on page 374 lists the output fields for the <code>show chassis cluster ip-monitoring status</code> command.

Table 35: 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: <b>reachable</b> , <b>unreachable</b> , and <b>unknown</b> . 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 <a href="#">Table 36 on page 375</a> .

Table 35: show chassis cluster ip-monitoring status Output Fields (*continued*)

Field Name	Field Description
<b>Weight</b>	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 36: show chassis cluster ip-monitoring status redundancy group Reason Fields

Reason	Reason Description
<b>No route to host</b>	The router could not resolve the ARP, which is needed to send the ICMP packet to the host with the monitored IP address.
<b>No auxiliary IP found</b>	The redundant Ethernet interface does not have an auxiliary IP address configured.
<b>Reth child not up</b>	A child interface of a redundant Ethernet interface is down.
<b>redundancy-group state unknown</b>	Unable to obtain the state (primary, secondary, secondary-hold, disable) of a redundancy-group.
<b>No reth child MAC address</b>	Could not extract the MAC address of the redundant Ethernet child interface.
<b>Secondary link not monitored</b>	The secondary link might be down (the secondary child interface of a redundant Ethernet interface is either down or non-functional).
<b>Unknown</b>	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

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis cluster statistics
<b>Release Information</b>	Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0.
<b>Description</b>	Display information about chassis cluster services and interfaces.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">clear chassis cluster statistics on page 341</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis cluster statistics on page 378</a> <a href="#">show chassis cluster statistics (SRX5000 line devices) on page 379</a> <a href="#">show chassis cluster statistics (SRX5000 line devices) on page 380</a>
<b>Output Fields</b>	Table 37 on page 377 lists the output fields for the <b>show chassis cluster statistics</b> command. Output fields are listed in the approximate order in which they appear.

**Table 37: show chassis cluster statistics Output Fields**

Field Name	Field Description
<b>Control link statistics</b>	<p>Statistics of the control link used by chassis cluster traffic. Statistics for <b>Control link 1</b> are displayed when you use dual control links (SRX5000 lines only). Note that the output for the SRX5000 lines will always show <b>Control link 0</b> and <b>Control link 1</b> statistics, even though only one control link is active or working.</p> <ul style="list-style-type: none"> <li>• <b>Heartbeat packets sent</b>—Number of heartbeat messages sent on the control link.</li> <li>• <b>Heartbeat packets received</b>—Number of heartbeat messages received on the control link.</li> <li>• <b>Heartbeat packet errors</b>—Number of heartbeat packets received with errors on the control link.</li> </ul>
<b>Fabric link statistics</b>	<p>Statistics of the fabric link used by chassis cluster traffic. Statistics for <b>Child Link 1</b> are displayed when you use dual fabric links.</p> <ul style="list-style-type: none"> <li>• <b>Probes sent</b>—Number of probes sent on the fabric link.</li> <li>• <b>Probes received</b>—Number of probes received on the fabric link.</li> </ul>

Table 37: show chassis cluster statistics Output Fields (*continued*)

Field Name	Field Description
Services Synchronized	<ul style="list-style-type: none"> <li>• <b>Service name</b>—Name of the service.</li> <li>• <b>Rtos sent</b>—Number of runtime objects (RTOs) sent.</li> <li>• <b>Rtos received</b>—Number of RTOs received.</li> <li>• <b>Translation context</b>—Messages synchronizing Network Address Translation (NAT) translation context.</li> <li>• <b>Incoming NAT</b>—Messages synchronizing incoming Network Address Translation (NAT) service.</li> <li>• <b>Resource manager</b>—Messages synchronizing resource manager groups and resources.</li> <li>• <b>Session create</b>—Messages synchronizing session creation.</li> <li>• <b>Session close</b>—Messages synchronizing session close.</li> <li>• <b>Session change</b>—Messages synchronizing session change.</li> <li>• <b>Gate create</b>—Messages synchronizing creation of pinholes (temporary openings in the firewall).</li> <li>• <b>Session ageout refresh request</b>—Messages synchronizing request session after age-out.</li> <li>• <b>Session ageout refresh reply</b>—Messages synchronizing reply session after age-out.</li> <li>• <b>IPsec VPN</b>—Messages synchronizing VPN session.</li> <li>• <b>Firewall user authentication</b>—Messages synchronizing firewall user authentication session.</li> <li>• <b>MGCP ALG</b>—Messages synchronizing MGCP ALG sessions.</li> <li>• <b>H323 ALG</b>—Messages synchronizing H.323 ALG sessions.</li> <li>• <b>SIP ALG</b>—Messages synchronizing SIP ALG sessions.</li> <li>• <b>SCCP ALG</b>—Messages synchronizing SCCP ALG sessions.</li> <li>• <b>PPTP ALG</b>—Messages synchronizing PPTP ALG sessions.</li> <li>• <b>RTSP ALG</b>—Messages synchronizing RTSP ALG sessions.</li> <li>• <b>MAC address learning</b>—Messages synchronizing MAC address learning.</li> </ul>

## 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           RT0s sent  RT0s 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 313](#)
  - [clear chassis cluster failover-count on page 337](#)
  - [request chassis cluster failover node on page 343](#)
  - [request chassis cluster failover reset on page 345](#)

**List of Sample Output** [show chassis cluster status on page 382](#)  
[show chassis cluster status redundancy-group 1 on page 383](#)

**Output Fields** [Table 38 on page 381](#) lists the output fields for the `show chassis cluster status` command. Output fields are listed in the approximate order in which they appear.

**Table 38: 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 ( <code>node0</code> or <code>node1</code> ).
Priority	Assigned priority for the redundancy group on that node.

Table 38: show chassis cluster status Output Fields (*continued*)

Field Name	Field Description
Status	<p>State of the redundancy group (<b>Primary</b>, <b>Secondary</b>, <b>Lost</b>, or <b>Unavailable</b>).</p> <ul style="list-style-type: none"> <li>• <b>Primary</b>—Redundancy group is active and passing traffic.</li> <li>• <b>Secondary</b>—Redundancy group is passive and not passing traffic.</li> <li>• <b>Lost</b>—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.</li> <li>• <b>Unavailable</b>—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.</li> </ul>
Preempt	<ul style="list-style-type: none"> <li>• <b>Yes</b>: Mastership can be preempted based on priority.</li> <li>• <b>No</b>: Change in priority will not preempt the mastership.</li> </ul>
Manual failover	<ul style="list-style-type: none"> <li>• <b>Yes</b>: If the Mastership is set manually through the CLI with the <b>request chassis cluster failover node</b> or <b>request chassis cluster failover redundancy-group</b> command. This overrides <b>Priority</b> and <b>Preempt</b>.</li> <li>• <b>No</b>: Mastership is not set manually through the CLI.</li> </ul>
Monitor-failures	<ul style="list-style-type: none"> <li>• <b>None</b>: Cluster working properly.</li> <li>• <b>Monitor Failure code</b>: Cluster is not working properly and the respective failure code is displayed.</li> </ul>

## 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 403](#)

**List of Sample Output** [show chassis environment on page 384](#)

**Output Fields** [Table 39 on page 384](#) lists the output fields for the **show chassis environment** command. Output fields are listed in the approximate order in which they appear.

**Table 39: 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: <b>OK</b> , <b>Testing</b> (during initial power-on), <b>Failed</b> , or <b>Absent</b> .

## 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 ethernet-switch

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis ethernet-switch
<b>Release Information</b>	Command introduced in Junos OS Release 9.2.
<b>Description</b>	SRX Series devices display information about the ports on the Control Board (CB) Ethernet switch.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">cluster (Chassis) on page 280</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis ethernet-switch on page 388</a>
<b>Output Fields</b>	<a href="#">Table 40 on page 388</a> lists the output fields for the <b>show chassis ethernet-switch</b> command. Output fields are listed in the approximate order in which they appear.

**Table 40: 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: <ul style="list-style-type: none"> <li>• FPC0 (Flexible PIC Concentrator 0) through FPC7</li> <li>• Local controller</li> <li>• Routing Engine</li> <li>• Other Routing Engine (on a system with two Routing Engines)</li> <li>• SPMB (Switch Processor Mezzanine Board)</li> </ul>
Speed is	Speed at which the Ethernet link is running.
Duplex is	Duplex type of the Ethernet link: <b>full</b> or <b>half</b> .
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

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis fabric plane
<b>Release Information</b>	Command introduced in Junos OS Release 9.2.
<b>Description</b>	Show state of fabric management plane.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show chassis fabric plane-location on page 398</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis fabric plane(SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4) on page 393</a>
<b>Output Fields</b>	Table 41 on page 392 lists the output fields for the <b>show chassis fabric plane</b> command. Output fields are listed in the approximate order in which they appear.

**Table 41: 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: <ul style="list-style-type: none"> <li>• <b>ACTIVE</b>—SIB is operational and running.</li> <li>• <b>FAULTY</b>— SIB is in alarmed state where the SIB's plane is not operational for the following reasons:               <ul style="list-style-type: none"> <li>• On-board fabric ASIC is not operational.</li> <li>• Fiber-optic connector faults.</li> <li>• FPC connector faults.</li> <li>• SIB midplane connector faults.</li> </ul> </li> </ul>	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: <ul style="list-style-type: none"> <li>• <b>Links ok</b>: Link between SIB and FPC is active.</li> <li>• <b>Link error</b>: Link between SIB and FPC is not operational.</li> <li>• <b>Unused</b>: No FPC is present.</li> </ul>	none

Table 41: show chassis fabric plane Output Fields (*continued*)

Field Name	Field Description	Level of output
State	<p>State of the fabric plane:</p> <ul style="list-style-type: none"> <li>• <b>Online:</b> Fabric plane is operational and running and links on the SIB are operational.</li> <li>• <b>Offline:</b> Fabric plane state is <b>Offline</b> because the plane does not have four or more F2S and one F13 online.</li> <li>• <b>Empty:</b> Fabric plane state is <b>Empty</b> if all SIBs in the plane are absent.</li> <li>• <b>Spare:</b> Fabric plane is redundant and can be operational if the operational fabric plane encounters an error.</li> <li>• <b>Check:</b> Fabric plane is in alarmed state due to the following reason and the cause of the error must be resolved: <ul style="list-style-type: none"> <li>• One or more SIBs (belonging to the fabric plane) in the <b>Online</b> or <b>Spare</b> states has transitioned to the <b>Check</b> state. <b>Check</b> state of the SIB can be caused by link errors or destination errors.</li> </ul> </li> <li>• <b>Fault:</b> Fabric plane is in alarmed state if one or more SIBs belonging to the plane are in the <b>Fault</b> state. A SIB can be in the <b>Fault</b> state because of the following reasons: <ul style="list-style-type: none"> <li>• On-board fabric ASIC is not operational.</li> <li>• Fiber-optic connector faults.</li> <li>• FPC connector faults.</li> <li>• SIB midplane connector faults.</li> <li>• Link errors have exceeded the threshold.</li> </ul> </li> </ul>	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

<b>Supported Platforms</b>	SRX Series, vSRX
<b>Syntax</b>	show chassis fabric plane-location
<b>Release Information</b>	Command introduced in Junos OS Release 9.2.
<b>Description</b>	Show fabric plane location.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">show chassis fabric plane on page 392</a></li> </ul>
<b>List of Sample Output</b>	<a href="#">show chassis fabric plane-location(SRX5600 and SRX5800 devices with SRX5000 line SCB II (SRX5K-SCBE) and SRX5K-RE-1800X4) on page 398</a>
<b>Output Fields</b>	<a href="#">Table 42 on page 398</a> lists the output fields for the <b>show chassis fabric plane-location</b> command. Output fields are listed in the approximate order in which they appear.

**Table 42: show chassis fabric plane-location Output Fields**

Field Name	Field Description
Plane <i>n</i>	Plane number.
Control Board <i>n</i>	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 392](#)
- [show chassis fabric plane-location on page 398](#)

**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 401](#)

**Output Fields** [Table 43 on page 400](#) lists the output fields for the **show chassis fabric summary** command. Output fields are listed in the approximate order in which they appear.

**Table 43: show chassis fabric summary Output Fields**

Field Name	Field Description
Plane	Plane number.

Table 43: show chassis fabric summary Output Fields (*continued*)

Field Name	Field Description
<b>State</b>	<p>State of the SIB or FPC:</p> <ul style="list-style-type: none"> <li>• <b>Online</b>—Switch Interface Board (SIB) is operational and running.</li> <li>• <b>Empty</b>—SIB is powered down.</li> <li>• <b>Check</b>—SIB is in the <b>Check</b> state because of the following reasons: <ul style="list-style-type: none"> <li>• SIB is not inserted properly.</li> <li>• 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.</li> <li>• 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: <ul style="list-style-type: none"> <li>• Link errors caused by a link training failure at initialization time—The Packet Forwarding Engine does not use the SIB to send traffic. The <b>show chassis fabric fpcs</b> command shows <b>Plane disabled</b> as status for this link.</li> <li>• Link errors caused by CRC errors detected at runtime—The Packet Forwarding Engine continues to use the SIB to send traffic. The <b>show chassis fabric fpcs</b> command shows <b>Link error</b> as the status for this link.</li> </ul> </li> </ul> </li> </ul> <p>For information about link and destination errors, issue the <b>show chassis fabric fpcs</b> commands.</p> <ul style="list-style-type: none"> <li>• <b>Spare</b>—SIB is redundant and will move to active state if one of the working SIBs fails.</li> </ul>
<b>Errors</b>	<p>Indicates whether there is any error on the SIB.</p> <ul style="list-style-type: none"> <li>• <b>None</b>—No errors</li> <li>• <b>Link Errors</b>—Fabric link errors were found on the SIB RX link.</li> <li>• <b>Cell drops</b>—Fabric cell drops were found on the SIB ASIC.</li> <li>• <b>Link, Cell drops</b>—Both link errors and cell drops were detected on at least one of the FPC's fabric links.</li> </ul> <p><b>NOTE:</b> The <b>Errors</b> column is empty only when the FPC or SIB is offline.</p>
<b>Uptime</b>	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)

<b>Supported Platforms</b>	SRX Series
<b>Syntax</b>	<code>show chassis hardware</code> <clei-models   detail   extensive   models   node ( <i>node-id</i>   all   local   primary )>
<b>Release Information</b>	Command introduced in Junos OS Release 9.2. Command modified in Junos OS Release 9.2 to include <b>node</b> option.
<b>Description</b>	Display chassis hardware information.
<b>Options</b>	<ul style="list-style-type: none"> <li>• <b>clei-models</b>—(Optional) Display Common Language Equipment Identifier Code (CLEI) barcode and model number for orderable field-replaceable units (FRUs).</li> <li>• <b>detail   extensive</b>—(Optional) Display the specified level of output.</li> <li>• <b>models</b>—(Optional) Display model numbers and part numbers for orderable FRUs.</li> <li>• <b>node</b>—(Optional) For chassis cluster configurations, display chassis hardware information on a specific node (device) in the cluster. <ul style="list-style-type: none"> <li>• <b>node-id</b>—Identification number of the node. It can be 0 or 1.</li> <li>• <b>local</b>—Display information about the local node.</li> <li>• <b>primary</b>—Display information about the primary node.</li> </ul> </li> </ul>
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <i>Juniper Networks Devices Processing Overview</i></li> <li>• <i>Interface Naming Conventions</i></li> </ul>
<b>Output Fields</b>	Table 44 on page 403 lists the output fields for the <b>show chassis hardware</b> command. Output fields are listed in the approximate order in which they appear.

Table 44: show chassis hardware Output Fields

Field Name	Field Description
<b>Item</b>	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.
<b>Version</b>	Revision level of the chassis component.
<b>Part Number</b>	Part number for the chassis component.
<b>Serial Number</b>	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 44: show chassis hardware Output Fields (continued)**

Field Name	Field Description
<b>Assb ID or Assembly ID</b>	Identification number that describes the FRU hardware.
<b>FRU model number</b>	Model number of FRU hardware component.
<b>CLEI code</b>	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.
<b>EEPROM Version</b>	ID EEPROM version used by hardware component: 0x01 (version 1) or 0x02 (version 2).

Table 44: show chassis hardware Output Fields (*continued*)

Field Name	Field Description
Description	<p data-bbox="578 390 951 415">Brief description of the hardware item:</p> <ul data-bbox="578 443 1424 1579" style="list-style-type: none"> <li data-bbox="578 443 808 468">• Type of power supply.</li> <li data-bbox="578 474 1424 636">• Switch Control Board (SCB) Starting with Junos OS Release 12.1X47-D15, the SRX5K-SCBE (SCB2) is introduced. <ul data-bbox="602 562 1424 1010" style="list-style-type: none"> <li data-bbox="602 562 1424 636">• 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.</li> <li data-bbox="602 642 1373 695">• With an SCB2, a third SCB is supported. If a third SCB is plugged in, it provides intra-chassis fabric redundancy.</li> <li data-bbox="602 701 1424 810">• 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.</li> <li data-bbox="602 816 1424 926">• 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.</li> <li data-bbox="602 932 1424 1010">• 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.</li> </ul> </li> <li data-bbox="578 1016 1424 1089">Starting with Junos OS Release 15.1X49-D10, the SRX5K-SCB3 (SCB3) with enhanced midplanes is introduced. <ul data-bbox="602 1100 1424 1579" style="list-style-type: none"> <li data-bbox="602 1100 1344 1125">• All existing SCB software that is supported by SCB2 is supported on SCB3.</li> <li data-bbox="602 1131 1305 1157">• SRX5K-RE-1800X4 (RE2). Mixed Routing Engine use is not supported.</li> <li data-bbox="602 1163 1424 1241">• 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.</li> <li data-bbox="602 1247 1424 1325">• 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.</li> <li data-bbox="602 1331 1424 1388">• SCB3 supports up to 400 Gbps per slot with old midplanes and up to 500 Gbps per slot with new midplanes.</li> <li data-bbox="602 1394 1084 1419">• SCB3 supports fabric intra-chassis redundancy.</li> <li data-bbox="602 1425 1424 1503">• 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).</li> <li data-bbox="602 1509 1036 1535">• SCB3 has a second external Ethernet port.</li> <li data-bbox="602 1541 1127 1566">• Fabric bandwidth increasing mode is not supported.</li> </ul> </li> </ul>

Table 44: show chassis hardware Output Fields (*continued*)

Field Name	Field Description
	<ul style="list-style-type: none"> <li>• Type of Flexible PIC Concentrator (FPC), Physical Interface Card (PIC), Modular Interface Cards (MICs), and PIMs.</li> <li>• IOCs           <ul style="list-style-type: none"> <li>Starting with Junos OS Release 15.1X49-D10, the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) are introduced.</li> <li>• IOC3 has two types of IOC3 MPCs, which have different built-in MICs: the 24x10GE + 6x40GE MPC and the 2x100GE + 4x10GE MPC.</li> <li>• IOC3 supports SCB3 and SRX5000 line backplane and enhanced backplane.</li> <li>• 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.</li> <li>• IOC3 interoperates with SCB2 and SCB3.</li> <li>• IOC3 interoperates with the SRX5K-SPC-4-15-320 (SPC2) and the SRX5K-MPC (IOC2).</li> <li>• The maximum power consumption for one IOC3 is 645W. An enhanced power module must be used.</li> <li>• The IOC3 does not support the following command to set a PIC to go offline or online:  <b>request chassis pic fpc-slot &lt;fpc-slot&gt; pic-slot &lt;pic-slot&gt; &lt;offline   online&gt; .</b></li> <li>• IOC3 supports 240 Gbps of throughput with the enhanced SRX5000 line backplane.</li> <li>• Chassis cluster functions the same as for the SRX5000 line IOC2.</li> <li>• IOC3 supports intra-chassis and inter-chassis fabric redundancy mode.</li> <li>• IOC3 supports ISSU and ISHU in chassis cluster mode.</li> <li>• IOC3 supports intra-FPC and Inter-FPC Express Path (previously known as <i>services offloading</i>) with IPv4.</li> <li>• NAT of IPv4 and IPv6 in normal mode and IPv4 for Express Path mode.</li> <li>• 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 <b>set chassis fpc &lt;slot&gt; pic &lt;pic&gt; power off</b> command to choose the PICs you want to power on.</li> </ul> </li> </ul> <p><b>NOTE:</b> Fabric bandwidth increasing mode is not supported on IOC3.</p> <ul style="list-style-type: none"> <li>• SRX Clustering Module (SCM)</li> <li>• Fan tray</li> <li>• For hosts, the Routing Engine type.           <ul style="list-style-type: none"> <li>• Starting with Junos OS Release 12.1X47-D15, the SRX5K-RE-1800X4 (RE2) Routing Engine is introduced.</li> <li>• 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.</li> </ul> </li> </ul> <p><b>NOTE:</b> The RE2 provides significantly better performance than the previously used Routing Engine, even with a single core.</p>

**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
FPC 0         REV 07   711-053832  ACMA3311      SRX Routing Engine
PIC 0         BUILTIN BUILTIN      FEB
Xcvr 12       REV 01   740-014132  61521013     12x1G-T-4x1G-SFP-4x10G
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
Airflow - AFO
Fan Tray 1
Airflow - AFO
Fan Tray 2
Airflow - AFO
Fan Tray 3
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 Module
PEM 0         Rev 03   740-034724  QCS17020203F  PS 4.1kW; 200-240V AC in
PEM 1         Rev 03   740-034724  QCS17020203C  PS 4.1kW; 200-240V AC in
PEM 2         Rev 04   740-034724  QCS17100200A  PS 4.1kW; 200-240V AC in
PEM 3         Rev 03   740-034724  QCS17080200M  PS 4.1kW; 200-240V AC in
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	AMBOHG3	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	C831XJ085	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	CF36KMOSY	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	MUCOMF2	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	AMGORBT	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	AN40MNV	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
PEM 0          Rev 03   740-034701   QCS13090900T  PS 1.4-2.6kW; 90-264V A

PEM 1          Rev 03   740-034701   QCS13090904T  PS 1.4-2.6kW; 90-264V A

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      BUILTIN      SRX5k DPC PPC
  PIC 0        BUILTIN  BUILTIN      BUILTIN      SPU Cp
  PIC 1        BUILTIN  BUILTIN      BUILTIN      SPU Flow
  PIC 2        BUILTIN  BUILTIN      BUILTIN      SPU Flow
  PIC 3        BUILTIN  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      BUILTIN      12x 10GE SFP+
  PIC 1        BUILTIN  BUILTIN      BUILTIN      12x 10GE SFP+
  PIC 2        BUILTIN  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      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      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      BUILTIN      SRX5k DPC PPC
  PIC 0        BUILTIN  BUILTIN      BUILTIN      SPU Flow
  PIC 1        BUILTIN  BUILTIN      BUILTIN      SPU Flow
  PIC 2        BUILTIN  BUILTIN      BUILTIN      SPU Flow
  PIC 3        BUILTIN  BUILTIN      BUILTIN      SPU Flow
Fan Tray      Enhanced Fan Tray

node1:
-----
Hardware inventory:
Item          Version  Part number  Serial number  Description
Chassis                               JN124FECOAGB  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	ALNON6C	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	COUIBCBAA	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 280](#)
- [request system snapshot \(SRX Series\)](#)

**List of Sample Output** [show chassis routing-engine \(Sample 1 - SRX550M\) on page 415](#)  
[show chassis routing-engine \(Sample 2 - vSRX\) on page 415](#)

**Output Fields** [Table 45 on page 414](#) lists the output fields for the **show chassis routing-engine** command. Output fields are listed in the approximate order in which they appear.

**Table 45: 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 45: 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

<b>Supported Platforms</b>	<a href="#">SRX Series, vSRX</a>
<b>Syntax</b>	show configuration chassis cluster traceoptions
<b>Release Information</b>	Command introduced in Junos OS Release 12.1.
<b>Description</b>	Display tracing options for the chassis cluster redundancy process.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">cluster (Chassis) on page 280</a></li> <li>• <a href="#">traceoptions (Chassis Cluster) on page 328</a></li> </ul>

**List of Sample Output** [show configuration chassis cluster traceoptions on page 417](#)

**Output Fields** [Table 46 on page 417](#) 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 46: 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 interfaces (Gigabit Ethernet) SRX device

<b>Supported Platforms</b>	SRX340, SRX345
<b>Syntax</b>	<pre>show interfaces ge- fpc /pic/port &lt;brief   detail   extensive   terse&gt; &lt;descriptions&gt; &lt;media&gt; &lt;snmp-index snmp-index&gt; &lt;statistics&gt;</pre>
<b>Release Information</b>	Command introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.
<b>Description</b>	Display status information about the specified Gigabit Ethernet interface.
<b>Options</b>	<p><b>ge-fpc/pic/port</b>—Display standard information about the specified Gigabit Ethernet interface.</p> <p><b>brief   detail   extensive   terse</b>—(Optional) Display the specified level of output.</p> <p><b>descriptions</b>—(Optional) Display interface description strings.</p> <p><b>media</b>—(Optional) Display media-specific information about network interfaces.</p> <p><b>snmp-index snmp-index</b>—(Optional) Display information for the specified SNMP index of the interface.</p> <p><b>statistics</b>—(Optional) Display static interface statistics.</p>
<b>Additional Information</b>	In a logical system, this command displays information only about the logical interfaces and not about the physical interfaces.
<b>Required Privilege Level</b>	view
<b>Related Documentation</b>	<ul style="list-style-type: none"> <li>• <a href="#">Understanding Media Access Control Security (MACsec) for SRX Series on page 249</a></li> <li>• <a href="#">Configuring Media Access Control Security (MACsec) on page 251</a></li> <li>• <a href="#">macsec on page 304</a></li> <li>• <a href="#">show security mka sessions for SRX device on page 440</a></li> <li>• <a href="#">show security macsec statistics for SRX device on page 434</a></li> </ul>
<b>List of Sample Output</b>	<p><a href="#">show interfaces (Gigabit Ethernet) for Fabric on page 428</a></p> <p><a href="#">show interfaces detail for Fabric on page 429</a></p>
<b>Output Fields</b>	Table 47 on page 419 describes the output fields for the <b>show interfaces</b> (Gigabit Ethernet) command. Output fields are listed in the approximate order in which they appear. For Gigabit Ethernet IQ and IQE PICs, the traffic and MAC statistics vary by interface type. For more information, see <a href="#">Table 48 on page 428</a> .

Table 47: show interfaces (Gigabit Ethernet) Output Fields

Field Name	Field Description	Level of Output
<b>Physical Interface</b>		
<b>Physical interface</b>	Name of the physical interface.	All levels
<b>Enabled</b>	State of the interface. Possible values are described in the “Enabled Field” section under .	All levels
<b>Interface index</b>	Index number of the physical interface, which reflects its initialization sequence.	<b>detail</b> none
<b>SNMP ifIndex</b>	SNMP index number for the physical interface.	<b>detail</b> none
<b>Link-level type</b>	Encapsulation being used on the physical interface.	All levels
<b>MTU</b>	Maximum transmission unit size on the physical interface.	All levels
<b>Link-mode</b>	Type of the link used for transmission.	
<b>Speed</b>	Speed at which the interface is running.	All levels
<b>MAC-REWRITE Error</b>	Error of the MAC-REWRITE.	
<b>Loopback</b>	Loopback status: <b>Enabled</b> or <b>Disabled</b> . If loopback is enabled, type of loopback: <b>Local</b> or <b>Remote</b> .	All levels
<b>Source filtering</b>	Source filtering status: <b>Enabled</b> or <b>Disabled</b> .	All levels
<b>LAN-PHY mode</b>	10-Gigabit Ethernet interface operating in Local Area Network Physical Layer Device (LAN PHY) mode. LAN PHY allows 10-Gigabit Ethernet wide area links to use existing Ethernet applications.	All levels
<b>Flow control</b>	Flow control status: <b>Enabled</b> or <b>Disabled</b> .	All levels
<b>Auto-negotiation</b>	(Gigabit Ethernet interfaces) Autonegotiation status: <b>Enabled</b> or <b>Disabled</b> .	All levels
<b>Remote-fault</b>	(Gigabit Ethernet interfaces) Remote fault status: <ul style="list-style-type: none"> <li>• <b>Online</b>—Autonegotiation is manually configured as online.</li> <li>• <b>Offline</b>—Autonegotiation is manually configured as offline.</li> </ul>	All levels
<b>Device flags</b>	Information about the physical device.	All levels
<b>Interface flags</b>	Information about the interface.	All levels
<b>Link flags</b>	Information about the link.	All levels
<b>CoS queues</b>	Number of CoS queues configured.	<b>detail</b> none
<b>Hold-times</b>	Current interface hold-time up and hold-time down, in milliseconds (ms).	

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Current address</b>	Configured MAC address.	<b>detail none</b>
<b>Hardware address</b>	Hardware MAC address.	<b>detail none</b>
<b>Last flapped</b>	Date, time, and how long ago the interface went from down to up. The format is <b>Last flapped: year-month-day hour:minute:second:timezone (hour:minute:second ago)</b> . For example, <b>Last flapped: 2002-04-26 10:52:40 PDT (04:33:20 ago)</b> .	<b>detail none</b>
<b>Input Rate</b>	Input rate in bits per second (bps) and packets per second (pps). The value in this field also includes the Layer 2 overhead bytes for ingress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.	None
<b>Output Rate</b>	Output rate in bps and pps. The value in this field also includes the Layer 2 overhead bytes for egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.	None
<b>Statistics last cleared</b>	Time when the statistics for the interface were last set to zero.	<b>detail extensive</b>
<b>Egress account overhead</b>	Layer 2 overhead in bytes that is accounted in the interface statistics for egress traffic.	<b>detail extensive</b>
<b>Ingress account overhead</b>	Layer 2 overhead in bytes that is accounted in the interface statistics for ingress traffic.	<b>detail extensive</b>
<b>Traffic statistics</b>	<p>Number and rate of bytes and packets received and transmitted on the physical interface.</p> <ul style="list-style-type: none"> <li>• <b>Input bytes</b>—Number of bytes received on the interface. The value in this field also includes the Layer 2 overhead bytes for ingress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.</li> <li>• <b>Output bytes</b>—Number of bytes transmitted on the interface. The value in this field also includes the Layer 2 overhead bytes for egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.</li> <li>• <b>Input packets</b>—Number of packets received on the interface.</li> <li>• <b>Output packets</b>—Number of packets transmitted on the interface.</li> </ul> <p>Gigabit Ethernet and 10-Gigabit Ethernet IQ PICs count the overhead and CRC bytes.</p> <p>For Gigabit Ethernet IQ PICs, the input byte counts vary by interface type.</p>	<b>detail</b>

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
Egress queues	<p>Total number of egress queues supported on the specified interface.</p> <p><b>NOTE:</b> In DPCs that are not of the enhanced type, such as DPC 40x1GER, DPCE 20x1GE + 2x10GE R, or DPCE 40x1GE R, you might notice a discrepancy in the output of the <b>show interfaces</b> command because incoming packets might be counted in the Egress queues section of the output. This problem occurs on non-enhanced DPCs because the egress queue statistics are polled from IMQ (Inbound Message Queuing) block of the I-chip. The IMQ block does not differentiate between ingress and egress WAN traffic; as a result, the combined statistics are displayed in the egress queue counters on the Routing Engine. In a simple VPLS scenario, if there is no MAC entry in DMAC table (by sending unidirectional traffic), traffic is flooded and the input traffic is accounted in IMQ. For bidirectional traffic (MAC entry in DMAC table), if the outgoing interface is on the same I-chip then both ingress and egress statistics are counted in a combined way. If the outgoing interface is on a different I-chip or FPC, then only egress statistics are accounted in IMQ. This behavior is expected with non-enhanced DPCs</p>	detail
Queue counters (Egress)	<p>CoS queue number and its associated user-configured forwarding class name.</p> <ul style="list-style-type: none"> <li>• <b>Queued packets</b>—Number of queued packets.</li> <li>• <b>Transmitted packets</b>—Number of transmitted packets.</li> <li>• <b>Dropped packets</b>—Number of packets dropped by the ASIC's RED mechanism.</li> </ul> <p><b>NOTE:</b> Due to accounting space limitations on certain Type 3 FPCs (which are supported in M320 and T640 routers), the <b>Dropped packets</b> field does not always display the correct value for queue 6 or queue 7 for interfaces on 10-port 1-Gigabit Ethernet PICs.</p>	detail extensive
Active alarms and Active defects	<p>Ethernet-specific defects that can prevent the interface from passing packets. When a defect persists for a certain amount of time, it is promoted to an alarm. Based on the router configuration, an alarm can ring the red or yellow alarm bell on the router, or turn on the red or yellow alarm LED on the craft interface. These fields can contain the value <b>None</b> or <b>Link</b>.</p> <ul style="list-style-type: none"> <li>• <b>None</b>—There are no active defects or alarms.</li> <li>• <b>Link</b>—Interface has lost its link state, which usually means that the cable is unplugged, the far-end system has been turned off, or the PIC is malfunctioning.</li> </ul>	detail none
Interface transmit statistics	<p>Status of the <b>interface-transmit-statistics</b> configuration: Enabled or Disabled.</p> <ul style="list-style-type: none"> <li>• <b>Enabled</b>—When the <b>interface-transmit-statistics</b> statement is included in the configuration. If this is configured, the interface statistics show the actual transmitted load on the interface.</li> <li>• <b>Disabled</b>—When the <b>interface-transmit-statistics</b> statement is not included in the configuration. If this is not configured, the interface statistics show the offered load on the interface.</li> </ul>	detail

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
MACSec statistics	Output <ul style="list-style-type: none"> <li>• <b>Secure Channel Transmitted:</b></li> <li>• <b>Protected Packets, Encrypted Packets, Protected Bytes, Encrypted Bytes</b></li> </ul> Input <ul style="list-style-type: none"> <li>• <b>Secure Channel Received:</b></li> <li>• <b>Accepted Packets, Validated Bytes, Decrypted Bytes</b></li> </ul>	
OTN FEC statistics	The forward error correction (FEC) counters provide the following statistics: <ul style="list-style-type: none"> <li>• <b>Corrected Errors</b>—Count of corrected errors in the last second.</li> <li>• <b>Corrected Error Ratio</b>—Corrected error ratio in the last 25 seconds. For example, 1e-7 is 1 error per 10 million bits.</li> </ul>	<b>detail</b>
PCS statistics	(10-Gigabit Ethernet interfaces) Displays Physical Coding Sublayer (PCS) fault conditions from the WAN PHY or the LAN PHY device. <ul style="list-style-type: none"> <li>• <b>Bit errors</b>—Number of seconds during which at least one bit error rate (BER) occurred while the PCS receiver is operating in normal mode.</li> <li>• <b>Errored blocks</b>—Number of seconds when at least one errored block occurred while the PCS receiver is operating in normal mode.</li> </ul>	<b>detail extensive</b>

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
MAC statistics	<p>Receive and Transmit statistics reported by the PIC's MAC subsystem, including the following:</p> <ul style="list-style-type: none"> <li>• <b>Total octets</b> and <b>total packets</b>—Total number of octets and packets. For Gigabit Ethernet IQ PICs, the received octets count varies by interface type.</li> <li>• <b>Unicast packets</b>, <b>Broadcast packets</b>, and <b>Multicast packets</b>—Number of unicast, broadcast, and multicast packets.</li> <li>• <b>CRC/Align errors</b>—Total number of packets received that had a length (excluding framing bits, but including FCS octets) of between 64 and 1518 octets, inclusive, and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error).</li> <li>• <b>FIFO error</b>—Number of FIFO errors that are reported by the ASIC on the PIC. If this value is ever nonzero, the PIC or a cable is probably malfunctioning.</li> <li>• <b>MAC control frames</b>—Number of MAC control frames.</li> <li>• <b>MAC pause frames</b>—Number of MAC control frames with <b>pause</b> operational code.</li> <li>• <b>Oversized frames</b>—There are two possible conditions regarding the number of oversized frames: <ul style="list-style-type: none"> <li>• Packet length exceeds 1518 octets, or</li> <li>• Packet length exceeds MRU</li> </ul> </li> <li>• <b>Jabber frames</b>—Number of frames that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error. This definition of jabber is different from the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition in which any packet exceeds 20 ms. The allowed range to detect jabber is from 20 ms to 150 ms.</li> <li>• <b>Fragment frames</b>—Total number of packets that were less than 64 octets in length (excluding framing bits, but including FCS octets) and had either an FCS error or an alignment error. Fragment frames normally increment because both runts (which are normal occurrences caused by collisions) and noise hits are counted.</li> <li>• <b>VLAN tagged frames</b>—Number of frames that are VLAN tagged. The system uses the TPID of 0x8100 in the frame to determine whether a frame is tagged or not. <p><b>NOTE:</b> The 20-port Gigabit Ethernet MIC (MIC-3D-20GE-SFP) does not have hardware counters for VLAN frames. Therefore, the <b>VLAN tagged frames</b> field displays 0 when the <b>show interfaces</b> command is executed on a 20-port Gigabit Ethernet MIC. In other words, the number of VLAN tagged frames cannot be determined for the 20-port Gigabit Ethernet MIC.</p> </li> <li>• <b>Code violations</b>—Number of times an event caused the PHY to indicate "Data reception error" or "invalid data symbol error."</li> </ul>	extensive
OTN Received Overhead Bytes	APS/PCC0: 0x02, APS/PCC1: 0x11, APS/PCC2: 0x47, APS/PCC3: 0x58 Payload Type: 0x08	extensive
OTN Transmitted Overhead Bytes	APS/PCC0: 0x00, APS/PCC1: 0x00, APS/PCC2: 0x00, APS/PCC3: 0x00 Payload Type: 0x08	extensive

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
Filter statistics	<p>Receive and Transmit statistics reported by the PIC's MAC address filter subsystem. The filtering is done by the content-addressable memory (CAM) on the PIC. The filter examines a packet's source and destination MAC addresses to determine whether the packet should enter the system or be rejected.</p> <ul style="list-style-type: none"> <li>• <b>Input packet count</b>—Number of packets received from the MAC hardware that the filter processed.</li> <li>• <b>Input packet rejects</b>—Number of packets that the filter rejected because of either the source MAC address or the destination MAC address.</li> <li>• <b>Input DA rejects</b>—Number of packets that the filter rejected because the destination MAC address of the packet is not on the accept list. It is normal for this value to increment. When it increments very quickly and no traffic is entering the router from the far-end system, either there is a bad ARP entry on the far-end system, or multicast routing is not on and the far-end system is sending many multicast packets to the local router (which the router is rejecting).</li> <li>• <b>Input SA rejects</b>—Number of packets that the filter rejected because the source MAC address of the packet is not on the accept list. The value in this field should increment only if source MAC address filtering has been enabled. If filtering is enabled, if the value increments quickly, and if the system is not receiving traffic that it should from the far-end system, it means that the user-configured source MAC addresses for this interface are incorrect.</li> <li>• <b>Output packet count</b>—Number of packets that the filter has given to the MAC hardware.</li> <li>• <b>Output packet pad count</b>—Number of packets the filter padded to the minimum Ethernet size (60 bytes) before giving the packet to the MAC hardware. Usually, padding is done only on small ARP packets, but some very small IP packets can also require padding. If this value increments rapidly, either the system is trying to find an ARP entry for a far-end system that does not exist or it is misconfigured.</li> <li>• <b>Output packet error count</b>—Number of packets with an indicated error that the filter was given to transmit. These packets are usually aged packets or are the result of a bandwidth problem on the FPC hardware. On a normal system, the value of this field should not increment.</li> <li>• <b>CAM destination filters, CAM source filters</b>—Number of entries in the CAM dedicated to destination and source MAC address filters. There can only be up to 64 source entries. If source filtering is disabled, which is the default, the values for these fields should be 0.</li> </ul>	extensive
PMA PHY	<p>(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET error information:</p> <ul style="list-style-type: none"> <li>• <b>Seconds</b>—Number of seconds the defect has been active.</li> <li>• <b>Count</b>—Number of times that the defect has gone from inactive to active.</li> <li>• <b>State</b>—State of the error. Any state other than <b>OK</b> indicates a problem.</li> </ul> <p>Subfields are:</p> <ul style="list-style-type: none"> <li>• <b>PHY Lock</b>—Phase-locked loop</li> <li>• <b>PHY Light</b>—Loss of optical signal</li> </ul>	extensive

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>WIS section</b>	(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET error information: <ul style="list-style-type: none"> <li>• <b>Seconds</b>—Number of seconds the defect has been active.</li> <li>• <b>Count</b>—Number of times that the defect has gone from inactive to active.</li> <li>• <b>State</b>—State of the error. Any state other than <b>OK</b> indicates a problem.</li> </ul> Subfields are: <ul style="list-style-type: none"> <li>• <b>BIP-B1</b>—Bit interleaved parity for SONET section overhead</li> <li>• <b>SEF</b>—Severely errored framing</li> <li>• <b>LOL</b>—Loss of light</li> <li>• <b>LOF</b>—Loss of frame</li> <li>• <b>ES-S</b>—Errored seconds (section)</li> <li>• <b>SES-S</b>—Severely errored seconds (section)</li> <li>• <b>SEFS-S</b>—Severely errored framing seconds (section)</li> </ul>	<b>extensive</b>
<b>Logical Interface</b>		
<b>Logical interface</b>	Name of the logical interface.	All levels
<b>Index</b>	Index number of the logical interface, which reflects its initialization sequence.	<b>detail</b> none
<b>SNMP ifIndex</b>	SNMP interface index number for the logical interface.	<b>detail</b> none
<b>Generation</b>	Unique number for use by Juniper Networks technical support only.	<b>detail</b>
<b>Flags</b>	Information about the logical interface. .	All levels
<b>VLAN-Tag</b>	Rewrite profile applied to incoming or outgoing frames on the outer ( <b>Out</b> ) VLAN tag or for both the outer and inner ( <b>In</b> ) VLAN tags. <ul style="list-style-type: none"> <li>• <b>push</b>—An outer VLAN tag is pushed in front of the existing VLAN tag.</li> <li>• <b>pop</b>—The outer VLAN tag of the incoming frame is removed.</li> <li>• <b>swap</b>—The outer VLAN tag of the incoming frame is overwritten with the user-specified VLAN tag information.</li> <li>• <b>push</b>—An outer VLAN tag is pushed in front of the existing VLAN tag.</li> <li>• <b>push-push</b>—Two VLAN tags are pushed in from the incoming frame.</li> <li>• <b>swap-push</b>—The outer VLAN tag of the incoming frame is replaced by a user-specified VLAN tag value. A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame.</li> <li>• <b>swap-swap</b>—Both the inner and the outer VLAN tags of the incoming frame are replaced by the user-specified VLAN tag value.</li> <li>• <b>pop-swap</b>—The outer VLAN tag of the incoming frame is removed, and the inner VLAN tag of the incoming frame is replaced by the user-specified VLAN tag value. The inner tag becomes the outer tag in the final frame.</li> <li>• <b>pop-pop</b>—Both the outer and inner VLAN tags of the incoming frame are removed.</li> </ul>	<b>brief detail</b> none

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Demux</b>	IP demultiplexing (demux) value that appears if this interface is used as the demux underlying interface. The output is one of the following: <ul style="list-style-type: none"> <li>• Source Family Inet</li> <li>• Destination Family Inet</li> </ul>	<b>detail none</b>
<b>Encapsulation</b>	Encapsulation on the logical interface.	All levels
<b>ACI VLAN: Dynamic Profile</b>	Name of the dynamic profile that defines the agent circuit identifier (ACI) interface set. If configured, the ACI interface set enables the underlying Ethernet interface to create dynamic VLAN subscriber interfaces based on ACI information.	<b>brief detail none</b>
<b>Protocol</b>	Protocol family. .	<b>detail none</b>
<b>MTU</b>	Maximum transmission unit size on the logical interface.	<b>detail none</b>
<b>Neighbor Discovery Protocol (NDP) Queue Statistics</b>	NDP statistics for protocol <b>inet6</b> under logical interface statistics. <ul style="list-style-type: none"> <li>• <b>Max nh cache</b>—Maximum interface neighbor discovery nexthop cache size.</li> <li>• <b>New hold nh limit</b>—Maximum number of new unresolved nexthops.</li> <li>• <b>Curr nh cnt</b>—Current number of resolved nexthops in the NDP queue.</li> <li>• <b>Curr new hold cnt</b>—Current number of unresolved nexthops in the NDP queue.</li> <li>• <b>NH drop cnt</b>—Number of NDP requests not serviced.</li> </ul>	All levels
<b>Dynamic Profile</b>	Name of the dynamic profile that was used to create this interface configured with a Point-to-Point Protocol over Ethernet (PPPoE) family.	<b>detail none</b>
<b>Service Name Table</b>	Name of the service name table for the interface configured with a PPPoE family.	<b>detail none</b>
<b>Max Sessions</b>	Maximum number of PPPoE logical interfaces that can be activated on the underlying interface.	<b>detail none</b>
<b>Duplicate Protection</b>	State of PPPoE duplicate protection: <b>On</b> or <b>Off</b> . When duplicate protection is configured for the underlying interface, a dynamic PPPoE logical interface cannot be activated when an existing active logical interface is present for the same PPPoE client.	<b>detail none</b>
<b>Direct Connect</b>	State of the configuration to ignore DSL Forum VSAs: <b>On</b> or <b>Off</b> . When configured, the router ignores any of these VSAs received from a directly connected CPE device on the interface.	<b>detail none</b>
<b>AC Name</b>	Name of the access concentrator.	<b>detail none</b>
<b>Maximum labels</b>	Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.	<b>detail none</b>

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Traffic statistics</b>	<p>Number and rate of bytes and packets received and transmitted on the specified interface set.</p> <ul style="list-style-type: none"> <li>• <b>Input bytes, Output bytes</b>—Number of bytes received and transmitted on the interface set. The value in this field also includes the Layer 2 overhead bytes for ingress or egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.</li> <li>• <b>Input packets, Output packets</b>—Number of packets received and transmitted on the interface set.</li> </ul>	<b>detail</b>
<b>Local statistics</b>	Number and rate of bytes and packets destined to the router.	<b>detail</b>
<b>Generation</b>	Unique number for use by Juniper Networks technical support only.	<b>detail</b>
<b>Transit statistics</b>	<p>Number and rate of bytes and packets transiting the switch.</p> <p><b>NOTE:</b> For Gigabit Ethernet intelligent queuing 2 (IQ2) interfaces, the logical interface egress statistics might not accurately reflect the traffic on the wire when output shaping is applied. Traffic management output shaping might drop packets after they are tallied by the <b>Output bytes</b> and <b>Output packets</b> interface counters. However, correct values display for both of these egress statistics when per-unit scheduling is enabled for the Gigabit Ethernet IQ2 physical interface, or when a single logical interface is actively using a shared scheduler.</p>	
<b>Route Table</b>	Route table in which the logical interface address is located. For example, <b>0</b> refers to the routing table inet.0.	<b>detail none</b>
<b>Flags</b>	Information about protocol family flags. .	<b>detail</b>
<b>Donor interface</b>	(Unnumbered Ethernet) Interface from which an unnumbered Ethernet interface borrows an IPv4 address.	<b>detail none</b>
<b>Preferred source address</b>	(Unnumbered Ethernet) Secondary IPv4 address of the donor loopback interface that acts as the preferred source address for the unnumbered Ethernet interface.	<b>detail none</b>
<b>Input Filters</b>	Names of any input filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parentheses next to all interfaces.	<b>detail</b>
<b>Output Filters</b>	Names of any output filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parentheses next to all interfaces.	<b>detail</b>
<b>Mac-Validate Failures</b>	Number of MAC address validation failures for packets and bytes. This field is displayed when MAC address validation is enabled for the logical interface.	<b>detail none</b>
<b>Addresses, Flags</b>	Information about the address flags. .	<b>detail none</b>
<b><i>protocol-family</i></b>	Protocol family configured on the logical interface. If the protocol is <b>inet</b> , the IP address of the interface is also displayed.	<b>brief</b>

Table 47: show interfaces (Gigabit Ethernet) Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Flags</b>	Information about the address flag. .	<b>detail extensive none</b>
<b>Destination</b>	IP address of the remote side of the connection.	<b>detail extensive none</b>
<b>Local</b>	IP address of the logical interface.	<b>detail extensive none</b>
<b>Broadcast</b>	Broadcast address of the logical interface.	<b>detail extensive none</b>
<b>Generation</b>	Unique number for use by Juniper Networks technical support only.	<b>detail extensive</b>

Table 48: Gigabit Ethernet IQ PIC Traffic and MAC Statistics by Interface Type

Interface Type	Sample Command	Byte and Octet Counts Include	Comments
Inbound physical interface	<b>show interfaces ge-0/3/0 extensive</b>	Traffic statistics:  Input bytes: 496 bytes per packet, representing the Layer 2 packet  MAC statistics:  Received octets: 500 bytes per packet, representing the Layer 2 packet + 4 bytes	The additional 4 bytes are for the CRC.
Inbound logical interface	<b>show interfaces ge-0/3/0.50 extensive</b>	Traffic statistics:  Input bytes: 478 bytes per packet, representing the Layer 3 packet	
Outbound physical interface	<b>show interfaces ge-0/0/0 extensive</b>	Traffic statistics:  Input bytes: 490 bytes per packet, representing the Layer 3 packet + 12 bytes  MAC statistics:  Received octets: 478 bytes per packet, representing the Layer 3 packet	For input bytes, the additional 12 bytes include 6 bytes for the destination MAC address plus 4 bytes for VLAN plus 2 bytes for the Ethernet type.
Outbound logical interface	<b>show interfaces ge-0/0/0.50 extensive</b>	Traffic statistics:  Input bytes: 478 bytes per packet, representing the Layer 3 packet	

## Sample Output

### show interfaces (Gigabit Ethernet) for Fabric

```

user@host> show interfaces ge-0/0/2
Physical interface: ge-0/0/2, Enabled, Physical link is Up
  Interface index: 153, SNMP ifIndex: 513
  Link-level type: 64, MTU: 9014, LAN-PHY mode, Link-mode: Full-duplex,

```

```

Speed: 1000mbps, BPDU Error: None, MAC-REWRITE Error: None,
Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
Auto-negotiation: Enabled, Remote fault: Online
Device flags : Present Running
Interface flags: SNMP-Traps Internal: 0x0
Link flags : None
CoS queues : 8 supported, 8 maximum usable queues
Current address: 30:7c:5e:44:98:f0, Hardware address: 30:7c:5e:44:98:43
Last flapped : 2016-07-14 19:32:16 UTC (17:52:04 ago)
Input rate : 2328 bps (1 pps)
Output rate : 2264 bps (1 pps)
Active alarms : None
Active defects : None
Interface transmit statistics: Disabled

Logical interface ge-0/0/2.0 (Index 77) (SNMP ifIndex 537)
Flags: Up SNMP-Traps 0x0 Encapsulation: ENET2
Input packets : 139146
Output packets: 134074
Security: Zone: Null
Protocol aenet, AE bundle: fab0.0 Link Index: 0

```

### show interfaces detail for Fabric

```

user@host> show interfaces ge-0/0/2 detail
Physical interface: ge-0/0/2, Enabled, Physical link is Up
Interface index: 153, SNMP ifIndex: 513, Generation: 156
Link-level type: 64, MTU: 9014, LAN-PHY mode, Link-mode: Full-duplex,
Speed: 1000mbps, BPDU Error: None, MAC-REWRITE Error: None,
Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
Auto-negotiation: Enabled, Remote fault: Online
Device flags : Present Running
Interface flags: SNMP-Traps Internal: 0x0
Link flags : None
CoS queues : 8 supported, 8 maximum usable queues
Hold-times : Up 0 ms, Down 0 ms
Current address: 30:7c:5e:44:98:f0, Hardware address: 30:7c:5e:44:98:43
Last flapped : 2016-07-14 19:32:16 UTC (17:52:25 ago)
Statistics last cleared: Never
Traffic statistics:
Input bytes :          20300010          2328 bps
Output bytes :          19041600          2264 bps
Input packets:           139189           1 pps
Output packets:          134116           1 pps
Egress queues: 8 supported, 4 in use
Queue counters:      Queued packets  Transmitted packets      Dropped packets

    0                   134121                134121                0
    1                      0                      0                      0
    2                      0                      0                      0
    3                      0                      0                      0

Queue number:      Mapped forwarding classes
    0                best-effort
    1                expedited-forwarding
    2                assured-forwarding
    3                network-control
Active alarms : None

```

```

Active defects : None
Interface transmit statistics: Disabled
MACSec statistics:
  Output
    Secure Channel Transmitted
      Protected Packets      : 0
      Encrypted Packets     : 128645
      Protected Bytes       : 0
      Encrypted Bytes       : 16723638
  Input
    Secure Channel Received
      Accepted Packets      : 128647
      Validated Bytes      : 0
      Decrypted Bytes      : 16723790

Logical interface ge-0/0/2.0 (Index 77) (SNMP ifIndex 537) (Generation 144)
Flags: Up SNMP-Traps 0x0 Encapsulation: ENET2
Traffic statistics:
  Input bytes :          20300152
  Output bytes :         19149160
  Input packets:          139190
  Output packets:         134116
Local statistics:
  Input bytes :          748678
  Output bytes :          871206
  Input packets:           5273
  Output packets:          5379
Transit statistics:
  Input bytes :          19551474
  Output bytes :         18277954
  Input packets:          133917
  Output packets:         128737
                               2328 bps
                               2264 bps
                               1 pps
                               1 pps
Security: Zone: Null
Flow Statistics :
Flow Input statistics :
  Self packets :          0
  ICMP packets :          0
  VPN packets :          0
  Multicast packets :      0
  Bytes permitted by policy : 0
  Connections established : 0
Flow Output statistics:
  Multicast packets :      0
  Bytes permitted by policy : 0
Flow error statistics (Packets dropped due to):
  Address spoofing:        0
  Authentication failed:   0
  Incoming NAT errors:     0
  Invalid zone received packet: 0
  Multiple user authentications: 0
  Multiple incoming NAT:   0
  No parent for a gate:    0
  No one interested in self packets: 0
  No minor session:        0
  No more sessions:        0
  No NAT gate:             0
  No route present:        0
  No SA for incoming SPI:  0
  No tunnel found:         0
  No session for a gate:   0
  No zone or NULL zone binding 0

```

```
Policy denied: 0
Security association not active: 0
TCP sequence number out of window: 0
Syn-attack protection: 0
User authentication errors: 0
Protocol aenet, AE bundle: fab0.0 Link Index: 0, Generation: 159,
Route table: 0
```

## show security macsec connections

**Supported Platforms** [SRX340, SRX345](#)

**Syntax** show security macsec connections  
<interface *interface-name*>

**Release Information** Command introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Display the status of the active MACsec connections on the device.

**Options** **none**—Display MACsec connection information for all interfaces on the device.

**interface *interface-name***—(Optional) Display MACsec connection information for the specified interface only.

**Required Privilege Level** view

**Related Documentation**

- [show security mka statistics on page 438](#)

**List of Sample Output** [show security macsec connections on page 433](#)

**Output Fields** [Table 49 on page 432](#) lists the output fields for the **show security macsec connections** command. Output fields are listed in the approximate order in which they appear.

**Table 49: show security macsec connections Output Fields**

Field Name	Field Description
<b>Fields for Interface</b>	
<b>Interface name</b>	Name of the interface.
<b>CA name</b>	Name of the connectivity association.  A connectivity association is named using the <b>connectivity-association</b> statement when you are enabling MACsec.
<b>Cipher suite</b>	Name of the cipher suite used for encryption.
<b>Key server offset</b>	Offset setting.  The offset is set using the <b>offset</b> statement when configuring the connectivity association when using static connectivity association key (CAK) or dynamic security mode.
<b>Replay protect</b>	Replay protection setting. Replay protection is enabled when this output is <b>on</b> and disabled when this output is <b>off</b> .  You can enable replay protection using the <b>replay-protect</b> statement in the connectivity association.

## Sample Output

### show security macsec connections

```
user@host> show security macsec connections
Interface name: fxp1
  CA name: ca1
  Cipher suite: GCM-AES-128   Encryption: on
  Key server offset: 0       Include SCI: no
  Replay protect: off       Replay window: 0
```

## show security macsec statistics for SRX device

**Supported Platforms** [SRX340, SRX345](#)

**Syntax** show security macsec statistics  
<brief | detail>  
<interface *interface-name*>

**Release Information** Command introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Display Media Access Control Security (MACsec) statistics.

**Options** **none**—Display MACsec statistics in brief form for all interfaces on the switch.

**brief | detail**—(Optional) Display the specified level of output. Using the **brief** option is equivalent to entering the command with no options (the default). The **detail** option displays additional fields that are not visible in the **brief** output.



**NOTE:** The field names that only appear in this command output when you enter the **detail** option are mostly useful for debugging purposes by Juniper Networks support personnel.

**interface *interface-name***—(Optional) Display MACsec statistics for the specified interface only.

**Required Privilege Level** view

- Related Documentation**
- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
  - [Configuring Media Access Control Security \(MACsec\) on page 251](#)
  - [macsec on page 304](#)
  - [show interfaces \(Gigabit Ethernet\) SRX device on page 418](#)
  - [show security mka sessions for SRX device on page 440](#)

**List of Sample Output** [show security macsec statistics interface fxp1 detail on page 437](#)

**Output Fields** [Table 50 on page 435](#) lists the output fields for the **show security macsec statistics** command. Output fields are listed in the approximate order in which they appear.

The field names that appear in this command output only when you enter the **detail** option are mostly useful for debugging purposes by Juniper Networks support personnel. Those field names are, therefore, not included in this table.

Table 50: show security macsec statistics Output Fields

Field Name	Field Description	Level of Output
<b>Interface name</b>	Name of the interface.	All levels
<b>Fields for Secure Channel transmitted</b>		
<b>Encrypted packets</b>	Total number of packets transmitted out of the interface in the secure channel that were secured and encrypted using MACsec.  Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).	All levels
<b>Encrypted bytes</b>	Total number of bytes transmitted out of the interface in the secure channel that were secured and encrypted using MACsec.  Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).	All levels
<b>Protected packets</b>	Total number of packets transmitted out of the interface in the secure channel that were secured but not encrypted using MACsec.  Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).	All levels
<b>Protected bytes</b>	Total number of bytes transmitted out of the interface in the secure channel that were secured but not encrypted using MACsec.  Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).	All levels
<b>Fields for Secure Association transmitted</b>		
<b>Encrypted packets</b>	Total number of packets transmitted out of the interface in the connectivity association that were secured and encrypted using MACsec.  The total includes the data packets transmitted in the secure channel and the control packets secured using a connectivity association key (CAK).	All levels
<b>Protected packets</b>	Total number of packets transmitted out of the interface in the connectivity association that were secured but not encrypted using MACsec.  The total includes the data packets transmitted in the secure channel and the control packets secured using a connectivity association key (CAK).	All levels
<b>Fields for Secure Channel received</b>		

Table 50: show security macsec statistics Output Fields (*continued*)

Field Name	Field Description	Level of Output
<b>Accepted packets</b>	<p>The number of received packets that have been accepted by the secure channel on the interface. The secure channel is used to send all data plane traffic on a MACsec-enabled link.</p> <p>A packet is considered accepted for this counter when it has been received by this interface and it has passed the MACsec integrity check.</p> <p>This counter increments for traffic that is and is not encrypted using MACsec.</p>	All levels
<b>Validated bytes</b>	<p>The number of bytes that have been validated by the MACsec integrity check and received on the secure channel on the interface. The secure channel is used to send all data plane traffic on a MACsec-enabled link.</p> <p>This counter does not increment when MACsec encryption is disabled.</p>	All levels
<b>Decrypted bytes</b>	<p>The number of bytes received in the secure channel on the interface that have been decrypted. The secure channel is used to send all data plane traffic on a MACsec-enabled link.</p> <p>An encrypted byte has to be decrypted before it can be received on the receiving interface. The decrypted bytes counter is incremented for received traffic that was encrypted using MACsec.</p>	All levels
<b>Fields for Secure Association received</b>		
<b>Accepted packets</b>	<p>The number of received packets that have been accepted in the connectivity association on the interface. The counter includes all control and data plane traffic accepted on the interface.</p> <p>A packet is considered accepted for this counter when it has been received by this interface and it has passed the MACsec integrity check.</p>	All levels
<b>Validated bytes</b>	<p>The number of bytes that have been validated by the MACsec integrity check and received on the connectivity association on the interface. The counter includes all control and data plane traffic accepted on the interface.</p> <p>This counter does not increment when MACsec encryption is disabled.</p>	All levels
<b>Decrypted bytes</b>	<p>The number of bytes received in the connectivity association on the interface that have been decrypted. The counter includes all control and data plane traffic accepted on the interface.</p> <p>An encrypted byte has to be decrypted before it can be received on the receiving interface. The decrypted bytes counter is incremented for received traffic that was encrypted using MACsec.</p>	All levels

## Sample Output

show security macsec statistics interface fxp1 detail

```
user@host> show security macsec statistics interface fxp1 detail
```

```
Interface name: fxp1
Secure Channel transmitted
  Encrypted packets: 2397305
  Encrypted bytes: 129922480
  Protected packets: 0
  Protected bytes: 0
Secure Association transmitted
  Encrypted packets: 2397305
  Protected packets: 0
Secure Channel received
  Accepted packets: 2395850
  Validated bytes: 0
  Decrypted bytes: 131715088
Secure Association received
  Accepted packets: 2395850
  Validated bytes: 0
  Decrypted bytes: 0
```

## show security mka statistics

**Supported Platforms** [SRX340, SRX345](#)

**Syntax** show security mka statistics  
<interface *interface-name*>

**Release Information** Command introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Display MACsec Key Agreement (MKA) protocol statistics.

The output for this command does not include statistics for MACsec data traffic. For MACsec data traffic statistics, see [show security macsec statistics for SRX device](#).

**Options**

- **interface *interface-name***—(Optional) Display the MKA information for the specified interface only.

**Required Privilege Level** view

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)
- [show interfaces \(Gigabit Ethernet\) SRX device on page 418](#)
- [show security macsec statistics for SRX device on page 434](#)

**List of Sample Output** [show security mka statistics on page 439](#)

**Output Fields** [Table 51 on page 438](#) lists the output fields for the **show security mka statistics** command. Output fields are listed in the approximate order in which they appear.

**Table 51: show security mka statistics Output Fields**

Field Name	Field Description
Received packets	Number of received MKA control packets.  This counter increments for received MKA control packets only. This counter does not increment when data packets are received.
Transmitted packets	Number of transmitted MKA packets  This counter increments for transmitted MKA control packets only. This counter does not increment when data packets are transmitted.
Version mismatch packets	Number of version mismatch packets.

Table 51: show security mka statistics Output Fields (*continued*)

Field Name	Field Description
CAK mismatch packets	Number of Connectivity Association Key (CAK) mismatch packets.  This counter increments when the connectivity association key (CAK) and connectivity association key name (CKN), which are user-configured values that have to match to enable MACsec, do not match for an MKA control packet.
ICV mismatch packets	Number of ICV mismatched packets.  This counter increments when the connectivity association key (CAK) value does not match on both ends of a MACsec-secured Ethernet link.
Duplicate message identifier packets	Number of duplicate message identifier packets.
Duplicate message number packets	Number of duplicate message number packets.
Duplicate address packets	Number of duplicate source MAC address packets.
Invalid destination address packets	Number of invalid destination MAC address packets.
Formatting error packets	Number of formatting error packets.
Old Replayed message number packets	Number of old replayed message number packets.

## Sample Output

### show security mka statistics

```

user@host> show security mka statistics

Interface name: fxp1
Received packets:          3
Transmitted packets:      14
Version mismatch packets: 0
CAK mismatch packets:     0
ICV mismatch packets:     0
Duplicate message identifier packets: 0
Duplicate message number packets: 0
Duplicate address packets: 0
Invalid destination address packets: 0
Formatting error packets: 0
Old Replayed message number packets: 0

```

## show security mka sessions for SRX device

**Supported Platforms** [SRX340, SRX345](#)

**Syntax** `show security mka sessions  
<interface interface-name>`

**Release Information** Command introduced in Junos OS Release 15.1X49-D60 for SRX340 and SRX345 devices.

**Description** Display MACsec Key Agreement (MKA) session information.

**Options**

- `interface interface-name`—(Optional) Display the MKA information for the specified interface only.

**Required Privilege Level** view

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) for SRX Series on page 249](#)
- [Configuring Media Access Control Security \(MACsec\) on page 251](#)
- [macsec on page 304](#)
- [show interfaces \(Gigabit Ethernet\) SRX device on page 418](#)
- [show security macsec statistics for SRX device on page 434](#)

**List of Sample Output** [show security mka sessions for SRX device on page 441](#)

**Output Fields** [Table 51 on page 438](#) lists the output fields for the `show security mka statistics` command. Output fields are listed in the approximate order in which they appear.

**Table 52: show security mka sessions Output Fields**

Field Name	Field Description
Interface name	Name of the interface.
Member identifier	Name of the member identifier.
CAK name	Name of the Connectivity Association Key (CAK.. The CAK is configured using the <code>cak</code> keyword when configuring the pre-shared key.
Transmit interval	The transmit interval.
Outbound SCI	Name of the outbound secure channel identifier.
Message number	Number of the last data message.
Key server	Key server status. The switch is the key server when this output is <b>yes</b> . The switch is not the key server when this output is <b>no</b> .

Table 52: show security mka sessions Output Fields (*continued*)

Field Name	Field Description
Key number	Key number.
Key server priority	The key server priority. The key server priority can be set using the <b>key-server-priority</b> statement.
Latest SAK AN	Name of the latest secure association key (SAK) association number.
Latest SAK KI	Name of the latest secure association key (SAK) key identifier.
Previous SAK AN	Name of the previous secure association key (SAK) association number.
Previous SAK KI	Name of the previous secure association key (SAK) key identifier.

## Sample Output

### show security mka sessions for SRX device

```
user@host> show security mka sessions
```

```

Interface name: fxp1
Member identifier: 71235CA1B78D0AF7B3F29CFB
CAK name: AAAA
Transmit interval: 10000(ms)
Outbound SCI: 30:7C:5E:44:98:42/1
Message number: 2326      Key number: 2
Key server: yes           Key server priority: 16
Latest SAK AN: 1          Latest SAK KI: 71235CA1B78D0AF7B3F29CFB/2
Previous SAK AN: 0        Previous SAK KI: 71235CA1B78D0AF7B3F29CFB/1

```

