



---

Junos<sup>®</sup> OS

## Chassis Cluster Feature Guide for SRX Series Devices



---

Modified: 2018-04-17

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

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. and/or its affiliates in the United States and other countries. All other trademarks may be property of their respective owners.

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

*Junos® OS Chassis Cluster Feature Guide for SRX Series Devices*  
Copyright © 2018 Juniper Networks, Inc. All rights reserved.

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

#### YEAR 2000 NOTICE

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

#### END USER LICENSE AGREEMENT

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

# Table of Contents

	About the Documentation . . . . .	xix
	Documentation and Release Notes . . . . .	xix
	Supported Platforms . . . . .	xix
	Using the Examples in This Manual . . . . .	xix
	Merging a Full Example . . . . .	xx
	Merging a Snippet . . . . .	xx
	Documentation Conventions . . . . .	xxi
	Documentation Feedback . . . . .	xxiii
	Requesting Technical Support . . . . .	xxiii
	Self-Help Online Tools and Resources . . . . .	xxiii
	Opening a Case with JTAC . . . . .	xxiv
<b>Chapter 1</b>	<b>Overview . . . . .</b>	<b>25</b>
	Chassis Cluster Overview . . . . .	25
	Chassis Cluster Overview . . . . .	25
	Benefits of Chassis Cluster . . . . .	26
	Chassis Cluster Functionality . . . . .	26
	Chassis Cluster Modes . . . . .	26
	How Chassis Clustering Works? . . . . .	26
	IPv6 Clustering Support . . . . .	27
	Chassis Cluster Limitations . . . . .	27
	Chassis Cluster Features Supported on SRX Series Devices . . . . .	30
	Chassis Cluster Supported Features (SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500) . . . . .	30
	Chassis Cluster-Supported Features (SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500) . . . . .	48
	Chassis Cluster Supported Features (SRX5800, SRX5600, and SRX5400) . . . . .	51
	Chassis Cluster-Supported Features (SRX5800, SRX5600, SRX5400, and SRX4600) . . . . .	72
	Chassis Cluster License Requirements . . . . .	75
	Understanding Chassis Cluster Licensing Requirements . . . . .	75
	Installing Licenses on the SRX Series Devices in a Chassis Cluster . . . . .	75
	Verifying Licenses on an SRX Series Device in a Chassis Cluster . . . . .	77

<b>Chapter 2</b>	<b>Setting Up a Chassis Cluster</b>	<b>81</b>
	SRX Series Chassis Cluster Configuration Overview	81
	Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming	86
	Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Devices	87
	Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX4600 Devices	90
	Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX4100 and SRX4200 Devices	91
	Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX5800, SRX5600, and SRX5400 Devices	93
	Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX3600, SRX3400, and SRX1400	95
	Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX650, SRX550, SRX240, SRX210, SRX110, and SRX100	97
	FPC Slot Numbering in SRX Series Device Cards	98
	Preparing Your Equipment for Chassis Cluster Formation	99
	Connecting SRX Series Devices to Create a Chassis Cluster	102
	Example: Setting the Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster	110
	Chassis Cluster Management Interfaces	113
	Understanding Management Interface on an Active Chassis Cluster	113
	Example: Configuring the Chassis Cluster Management Interface	114
	Chassis Cluster Fabric Interfaces	119
	Understanding Chassis Cluster Fabric Interfaces	119
	Supported Fabric Interface Types for SRX Series Devices (SRX300 Series, SRX550M, SRX1500, SRX4100/SRX4200, and SRX5000 Series)	120
	Supported Fabric Interface Types for SRX Series Devices (SRX650, SRX550, SRX240, SRX210, and SRX100 Devices)	121
	Jumbo Frame Support	121
	Understanding Fabric Interfaces on SRX5000 Line Devices for IOC2 and IOC3	121
	Understanding Session RTOs	122
	Understanding Data Forwarding	123
	Understanding Fabric Data Link Failure and Recovery	124
	Example: Configuring the Chassis Cluster Fabric Interfaces	125
	Verifying Chassis Cluster Data Plane Interfaces	127
	Viewing Chassis Cluster Data Plane Statistics	127
	Clearing Chassis Cluster Data Plane Statistics	128
	Chassis Cluster Control Plane Interfaces	129
	Understanding Chassis Cluster Control Plane and Control Links	129
	Understanding Chassis Cluster Control Links	130
	Example: Configuring Chassis Cluster Control Ports	131
	Verifying Chassis Cluster Control Plane Statistics	133
	Clearing Chassis Cluster Control Plane Statistics	134

	Chassis Cluster Redundancy Groups . . . . .	134
	Understanding Chassis Cluster Redundancy Groups . . . . .	134
	Understanding Chassis Cluster Redundancy Group 0: Routing Engines . . . . .	135
	Understanding Chassis Cluster Redundancy Groups 1 Through 128 . . . . .	136
	Example: Configuring Chassis Cluster Redundancy Groups . . . . .	138
	Chassis Cluster Redundant Ethernet Interfaces . . . . .	140
	Understanding Chassis Cluster Redundant Ethernet Interfaces . . . . .	140
	Example: Configuring Chassis Cluster Redundant Ethernet Interfaces . . . . .	145
	Configuring Chassis Clustering on an SRX Series Devices . . . . .	150
	Example: Configuring Chassis Clustering on an SRX Series Devices . . . . .	151
	Viewing a Chassis Cluster Configuration . . . . .	164
	Viewing Chassis Cluster Statistics . . . . .	164
	Clearing Chassis Cluster Statistics . . . . .	166
	Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes . . . . .	166
	Verifying Chassis Cluster Configuration Synchronization Status . . . . .	167
	Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Devices in a Chassis Cluster . . . . .	168
	Conditional Route Advertisement over Redundant Ethernet Interfaces on SRX Series Devices in a Chassis Cluster . . . . .	176
	Understanding Conditional Route Advertising in a Chassis Cluster . . . . .	176
	Example: Configuring Conditional Route Advertising in a Chassis Cluster . . . . .	178
<b>Chapter 3</b>	<b>Configuring Redundancy and Failover in a Chassis Cluster . . . . .</b>	<b>183</b>
	Chassis Cluster Dual Control Links . . . . .	183
	Understanding Chassis Cluster Dual Control Links . . . . .	183
	Benefits of Dual Control Links . . . . .	184
	Dual Control Links Functionality Requirements . . . . .	184
	Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster . . . . .	185
	Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices . . . . .	186
	Example: Configuring Chassis Cluster Control Ports for Dual Control Links . . . . .	188
	Chassis Cluster Dual Fabric Links . . . . .	190
	Understanding Chassis Cluster Dual Fabric Links . . . . .	190
	Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports . . . . .	191
	Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports . . . . .	193
	Monitoring of Global-Level Objects in a Chassis Cluster . . . . .	196
	Understanding SPU Monitoring . . . . .	196
	Understanding flowd Monitoring . . . . .	197
	Understanding Cold-Sync Monitoring . . . . .	198
	Understanding Cold-Sync Monitoring with SPU Replacement or Expansion . . . . .	198

Monitoring Chassis Cluster Interfaces . . . . .	200
Understanding Chassis Cluster Redundancy Group Interface Monitoring . . . . .	200
Benefits of Monitoring Chassis Cluster Redundancy Group	
Interfaces . . . . .	201
Example: Configuring Chassis Cluster Redundancy Group Interface	
Monitoring . . . . .	201
Monitoring IP Addresses on a Chassis Cluster . . . . .	229
IP Monitoring Overview . . . . .	230
Benefits of Monitoring IP Addresses in a Chassis Cluster . . . . .	232
Understanding Chassis Cluster Redundancy Group IP Address	
Monitoring . . . . .	232
Example: Configuring Chassis Cluster Redundancy Group IP Address	
Monitoring . . . . .	235
Example: Configuring IP Monitoring on SRX5000 Line Devices for IOC2 and	
IOC3 . . . . .	238
Configuring Cluster Failover Parameters . . . . .	245
Understanding Chassis Cluster Control Link Heartbeats, Failure, and	
Recovery . . . . .	245
Understanding Chassis Cluster Control Link Heartbeats . . . . .	245
Understanding Chassis Cluster Control Link Failure and Recovery . . . . .	246
Example: Configuring Chassis Cluster Control Link Recovery . . . . .	248
Chassis Cluster Redundancy Group Failover . . . . .	249
Understanding Chassis Cluster Redundancy Group Failover . . . . .	249
Preemptive Failover Delay Timer . . . . .	250
Understanding Chassis Cluster Redundancy Group Manual Failover . . . . .	253
Initiating a Chassis Cluster Manual Redundancy Group Failover . . . . .	255
Example: Configuring a Chassis Cluster with a Dampening Time Between	
Back-to-Back Redundancy Group Failovers . . . . .	257
Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group	
Failover . . . . .	258
Verifying Chassis Cluster Failover Status . . . . .	259
Clearing Chassis Cluster Failover Status . . . . .	260
<b>Chapter 4 Chassis Cluster Operations . . . . .</b>	<b>261</b>
Aggregated Ethernet Interfaces in a Chassis Cluster . . . . .	261
Understanding Link Aggregation Groups in a Chassis Cluster . . . . .	261
Example: Configuring Link Aggregation Groups in a Chassis Cluster . . . . .	263
Understanding Link Aggregation Group Failover in a Chassis Cluster . . . . .	267
Scenario 1: Monitored Interface Weight Is 255 . . . . .	267
Scenario 2: Monitored Interface Weight Is 75 . . . . .	268
Scenario 3: Monitored Interface Weight Is 100 . . . . .	268
Understanding LACP on Chassis Clusters . . . . .	269
Chassis Cluster Redundant Ethernet Interface Link Aggregation	
Groups . . . . .	269
Sub-LAGs . . . . .	270
Supporting Hitless Failover . . . . .	270
Managing Link Aggregation Control PDUs . . . . .	271
Example: Configuring LACP on Chassis Clusters . . . . .	271
Example: Configuring Chassis Cluster Minimum Links . . . . .	274

Example: Configuring Chassis Cluster Redundant Ethernet Interface Link	
Aggregation Groups on an SRX5000 Line Device with IOC2 or IOC3 . . .	275
Understanding VRRP on SRX Series Devices . . . . .	279
Overview of VRRP on SRX Series Devices . . . . .	279
Benefits of VRRP . . . . .	280
Sample VRRP Topology . . . . .	280
SRX Series Devices Support for VRRPv3 . . . . .	281
Limitations of VRRPv3 Features . . . . .	281
Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet	
Interfaces . . . . .	282
NTP Time Synchronization on Chassis Cluster . . . . .	291
NTP Time Synchronization on SRX Series Devices . . . . .	291
Example: Simplifying Network Management by Synchronizing the Primary	
and Backup Nodes with NTP . . . . .	292
Active/Passive Chassis Cluster Deployments . . . . .	298
Understanding Active/Passive Chassis Cluster Deployment . . . . .	298
Example: Configuring an Active/Passive Chassis Cluster on SRX5800	
Devices . . . . .	299
Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) . . . . .	316
Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) . . . . .	328
Understanding Active/Passive Chassis Cluster Deployment with an IPsec	
Tunnel . . . . .	329
Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec	
Tunnel . . . . .	330
Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec	
Tunnel (J-Web) . . . . .	346
Multicast Routing and Asymmetric Routing on Chassis Cluster . . . . .	348
Understanding Multicast Routing on a Chassis Cluster . . . . .	348
Understanding PIM Data Forwarding . . . . .	349
Understanding Multicast and PIM Session Synchronization . . . . .	349
Understanding Asymmetric Routing on a Chassis Cluster . . . . .	349
Understanding Failures in the Trust Zone Redundant Ethernet	
Interface . . . . .	350
Understanding Failures in the Untrust Zone Interfaces . . . . .	350
Example: Configuring an Asymmetric Chassis Cluster Pair . . . . .	351
Ethernet Switching on Chassis Cluster . . . . .	362
Layer 2 Ethernet Switching Capability in a Chassis Cluster Mode . . . . .	362
Understanding Layer 2 Ethernet Switching Capability in a Chassis Cluster	
on SRX Series Devices . . . . .	362
Understanding Chassis Cluster Failover and New Primary Election . . . . .	363
Benefits of Ethernet Switching on Chassis Cluster . . . . .	363
Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis	
Cluster Mode on a Security Device . . . . .	364
Example: Configuring IRB and VLAN with Members Across Two Nodes on	
a Security Device . . . . .	366
Media Access Control Security (MACsec) on Chassis Cluster . . . . .	370
Understanding Media Access Control Security (MACsec) . . . . .	370
How MACsec Works . . . . .	370
Understanding Connectivity Associations and Secure Channels . . . . .	371

	Understanding Static Connectivity Association Key Security Mode . . . .	371
	MACsec Considerations . . . . .	372
	Configuring Media Access Control Security (MACsec) . . . . .	373
	Configuration Considerations When Configuring MACsec on Chassis Cluster Setup . . . . .	373
	Configuring MACsec Using Static Connectivity Association Key Security Mode . . . . .	375
	Configuring Static CAK on the Chassis Cluster Control Port . . . . .	379
	Configuring Static CAK on the Chassis Cluster Fabric Port . . . . .	380
	Configuring Static CAK on the Control Port of SRX4600 Device in Chassis Cluster . . . . .	380
	Verifying MACSEC Configuration . . . . .	381
	Understanding SCTP Behavior in Chassis Cluster . . . . .	384
<b>Chapter 5</b>	<b>Upgrading or Disabling a Chassis Cluster . . . . .</b>	<b>387</b>
	Upgrading Individual Devices in a Chassis Cluster Separately . . . . .	387
	Upgrading Devices in a Chassis Cluster Using ICU . . . . .	388
	Upgrading Both Devices in a Chassis Cluster Using ICU . . . . .	388
	Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster . . . . .	389
	Upgrading ICU Using a Build Available on an FTP Server . . . . .	390
	Aborting an Upgrade in a Chassis Cluster During an ICU . . . . .	390
	Upgrading a Chassis Cluster Using In-Service Software Upgrade . . . . .	392
	Understanding ISSU for a Chassis Cluster . . . . .	392
	ISSU System Requirements . . . . .	396
	Upgrading Both Devices in a Chassis Cluster Using ISSU . . . . .	397
	Rolling Back Devices in a Chassis Cluster After an ISSU . . . . .	399
	Enabling an Automatic Chassis Cluster Node Failback After an ISSU . . . .	400
	Understanding Log Error Messages for Troubleshooting ISSU-Related Problems . . . . .	400
	Chassisd Process Errors . . . . .	400
	Understanding Common Error Handling for ISSU . . . . .	400
	ISSU Support-Related Errors . . . . .	403
	Initial Validation Checks Failure . . . . .	403
	Installation-Related Errors . . . . .	404
	Redundancy Group Failover Errors . . . . .	405
	Kernel State Synchronization Errors . . . . .	405
	Managing Chassis Cluster ISSU-Related Problems . . . . .	405
	Viewing ISSU Progress . . . . .	405
	Stopping ISSU Process if it Halts During an Upgrade . . . . .	406
	Recovering the Node in Case of a Failed ISSU . . . . .	406
	Disabling a Chassis Cluster . . . . .	408
<b>Part 1</b>	<b>Configuration Statements and Operational Commands</b>	
<b>Chapter 6</b>	<b>Configuration Statements . . . . .</b>	<b>413</b>
	aggregated-devices . . . . .	415
	apply-groups (Chassis Cluster) . . . . .	416
	arp-detect . . . . .	417
	arp-throttle . . . . .	418



authentication-key	419
authentication-type	420
cak	421
ckn	422
cluster (Chassis)	423
configuration-synchronize (Chassis Cluster)	425
connectivity-association	426
connectivity-association (MACsec Interfaces)	427
control-link-recovery	428
control-ports	429
exclude-protocol	430
fabric-options	431
gether-options (Chassis Cluster)	433
global-threshold	435
global-weight	436
gratuitous-arp-count	437
heartbeat-interval	438
heartbeat-threshold	439
hold-down-interval	440
include-sci	441
interface (Chassis Cluster)	442
interfaces (MACsec)	443
interface-monitor	444
ip-monitoring	445
key-server-priority (MACsec)	447
lacp (Interfaces)	448
link-protection (Chassis Cluster)	449
macsec	450
mka	451
must-secure	452
network-management	453
no-encryption (MACsec)	454
node (Chassis Cluster Redundancy Group)	455
ntp	456
ntp threshold	457
offset	459
preempt (Chassis Cluster)	461
pre-shared-key	462
priority (Protocols VRRP)	463
redundancy-group (Chassis Cluster)	464
redundant-ether-options	466
redundant-parent (Interfaces)	467
redundant-pseudo-interface-options	468
replay-protect	469
replay-window-size	470
reth-count (Chassis Cluster)	471
retry-count (Chassis Cluster)	472
retry-interval (Chassis Cluster)	473
route-active-on	474

	security-mode .....	475
	traceoptions (Chassis Cluster) .....	476
	transmit-interval (MACsec) .....	478
	virtual-address .....	479
	vrrp-group .....	480
	weight .....	482
<b>Chapter 7</b>	<b>Operational Commands .....</b>	<b>483</b>
	clear chassis cluster control-plane statistics .....	485
	clear chassis cluster data-plane statistics .....	486
	clear chassis cluster failover-count .....	487
	clear chassis cluster ip-monitoring failure-count .....	489
	clear chassis cluster ip-monitoring failure-count ip-address .....	490
	clear chassis cluster statistics .....	491
	request chassis cb .....	492
	request chassis cluster configuration-synchronize .....	493
	request chassis cluster failover node .....	494
	request chassis cluster failover redundancy-group .....	495
	request chassis cluster failover reset .....	497
	request chassis fpc .....	498
	request chassis cluster in-service-upgrade abort (ISSU) .....	499
	request security internal-security-association refresh .....	500
	request system scripts add .....	501
	request system reboot .....	504
	request system software in-service-upgrade (Maintenance) .....	505
	request system software rollback (SRX Series) .....	510
	set chassis cluster cluster-id node node-number reboot .....	511
	show chassis cluster control-plane statistics .....	512
	show chassis cluster data-plane interfaces .....	514
	show chassis cluster data-plane statistics .....	515
	show chassis cluster ethernet-switching interfaces .....	518
	show chassis cluster ethernet-switching status .....	519
	show chassis cluster information .....	521
	show chassis cluster information configuration-synchronization .....	526
	show chassis cluster information issu .....	528
	show chassis cluster interfaces .....	530
	show chassis cluster ip-monitoring status redundancy-group .....	535
	show chassis cluster statistics .....	538
	show chassis cluster status .....	542
	show chassis environment (Security) .....	545
	show chassis environment cb .....	549
	show chassis ethernet-switch .....	552
	show chassis fabric plane .....	556
	show chassis fabric plane-location .....	562
	show chassis fabric summary .....	564
	show chassis hardware (View) .....	567
	show chassis routing-engine (View) .....	578
	show configuration chassis cluster traceoptions .....	581
	set date ntp .....	582

show interfaces (Gigabit Ethernet) . . . . .	584
show system ntp threshold . . . . .	598
show security macsec connections . . . . .	599
show security macsec statistics (SRX Series Devices) . . . . .	601
show security mka statistics . . . . .	605
show security mka sessions (SRX Series Device . . . . .	607
show security internal-security-association . . . . .	609
show system license (View) . . . . .	610
show vrrp . . . . .	613



# List of Figures

<b>Chapter 2</b>	<b>Setting Up a Chassis Cluster</b> . . . . .	<b>81</b>
	Figure 1: Chassis Cluster Flow Diagram (SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, and SRX4600 Devices) . . . . .	83
	Figure 2: Chassis Cluster Flow Diagram (SRX5800, SRX5600, SRX5400 Devices) . . . . .	83
	Figure 3: Slot Numbering in SRX300 Chassis Cluster . . . . .	89
	Figure 4: Slot Numbering in SRX320 Chassis Cluster . . . . .	89
	Figure 5: Slot Numbering in SRX340 Chassis Cluster . . . . .	89
	Figure 6: Slot Numbering in SRX345 Chassis Cluster . . . . .	90
	Figure 7: Slot Numbering in SRX550M Chassis Cluster . . . . .	90
	Figure 8: Slot Numbering in SRX1500 Chassis Cluster . . . . .	90
	Figure 9: Slot Numbering in SRX4600 Chassis Cluster . . . . .	91
	Figure 10: Slot Numbering in SRX4100 Chassis Cluster . . . . .	92
	Figure 11: Slot Numbering in SRX4200 Chassis Cluster . . . . .	92
	Figure 12: Slot Numbering in SRX5800 Chassis Cluster . . . . .	94
	Figure 13: Chassis Cluster Slot Numbering for SRX3600 Devices . . . . .	96
	Figure 14: Chassis Cluster Slot Numbering for SRX3400 Devices . . . . .	96
	Figure 15: Chassis Cluster Slot Numbering for SRX1400 Devices . . . . .	97
	Figure 16: Chassis Cluster Slot Numbering for SRX650 Devices . . . . .	97
	Figure 17: Chassis Cluster Slot Numbering for SRX550 Devices . . . . .	97
	Figure 18: Chassis Cluster Slot Numbering for SRX240 Devices . . . . .	97
	Figure 19: Chassis Cluster Slot Numbering for SRX220 Devices . . . . .	97
	Figure 20: Chassis Cluster Slot Numbering for SRX210 Devices . . . . .	98
	Figure 21: Chassis Cluster Slot Numbering for SRX100 Devices . . . . .	98
	Figure 22: Connecting SRX300 Devices in a Chassis Cluster . . . . .	103
	Figure 23: Connecting SRX320 Devices in a Chassis Cluster . . . . .	103
	Figure 24: Connecting SRX340 Devices in a Chassis Cluster . . . . .	103
	Figure 25: Connecting SRX345 Devices in a Chassis Cluster . . . . .	104
	Figure 26: Connecting SRX550M Devices in a Chassis Cluster . . . . .	104
	Figure 27: Connecting SRX1500 Devices in a Chassis Cluster . . . . .	104
	Figure 28: Connecting SRX4600 Devices in a Chassis Cluster . . . . .	105
	Figure 29: Connecting SRX4100 Devices in a Chassis Cluster . . . . .	105
	Figure 30: Connecting SRX4200 Devices in a Chassis Cluster . . . . .	105
	Figure 31: Connecting SRX5800 Devices in a Chassis Cluster . . . . .	106
	Figure 32: Connecting SRX5600 Devices in a Chassis Cluster . . . . .	106
	Figure 33: Connecting SRX5400 Devices in a Chassis Cluster . . . . .	107
	Figure 34: Connecting SRX3600 Devices in a Chassis Cluster . . . . .	107
	Figure 35: Connecting SRX3400 Devices in a Chassis Cluster . . . . .	107
	Figure 36: Connecting SRX1400 Devices in a Chassis Cluster . . . . .	108
	Figure 37: Connecting SRX650 Devices in a Chassis Cluster . . . . .	108

	Figure 38: Connecting SRX550 Devices in a Chassis Cluster . . . . .	108
	Figure 39: Connecting SRX240 Devices in a Chassis Cluster . . . . .	108
	Figure 40: Connecting SRX220 Devices in a Chassis Cluster . . . . .	109
	Figure 41: Connecting SRX210 Devices in a Chassis Cluster . . . . .	109
	Figure 42: Connecting SRX110 Devices in a Chassis Cluster . . . . .	109
	Figure 43: Connecting SRX100 Devices in a Chassis Cluster . . . . .	109
	Figure 44: SRX Series Devices (SRX1500) In Chassis Cluster . . . . .	153
	Figure 45: Eight-Queue CoS on Redundant Ethernet Interfaces . . . . .	169
	Figure 46: Conditional Route Advertising . . . . .	177
	Figure 47: Conditional Route Advertising on SRX Series Devices in a Chassis Cluster . . . . .	179
<b>Chapter 3</b>	<b>Configuring Redundancy and Failover in a Chassis Cluster . . . . .</b>	<b>183</b>
	Figure 48: Connecting Dual Control Links (SRX5800 Devices) . . . . .	185
	Figure 49: SRX Series Chassis Cluster Interface Monitoring Topology Example . . . . .	203
	Figure 50: Transition from Primary State to Secondary State with Preemptive Delay . . . . .	252
<b>Chapter 4</b>	<b>Chassis Cluster Operations . . . . .</b>	<b>261</b>
	Figure 51: Topology for LAGs Connecting SRX Series Devices in Chassis Cluster to an EX Series Switch . . . . .	272
	Figure 52: Basic VRRP on SRX Series Switches . . . . .	280
	Figure 53: VRRP on Redundant interface . . . . .	283
	Figure 54: Synchronizing Time From Peer Node Through Control Link . . . . .	293
	Figure 55: Active/Passive Chassis Cluster Scenario . . . . .	298
	Figure 56: Basic Active/Passive Chassis Clustering on an SRX Series Device Topology Example . . . . .	301
	Figure 57: Active/Passive Chassis Cluster Topology . . . . .	317
	Figure 58: Active/Passive Chassis Cluster with IPsec Tunnel Scenario (SRX Series Devices) . . . . .	330
	Figure 59: Active/Passive Chassis Cluster with IPsec Tunnel Topology (SRX Series Devices) . . . . .	332
	Figure 60: Asymmetric Routing Chassis Cluster Scenario . . . . .	350
	Figure 61: Asymmetric Routing Chassis Cluster Topology . . . . .	352
	Figure 62: Layer 2 Ethernet Switching Across Chassis Cluster Nodes . . . . .	363

# List of Tables

	<b>About the Documentation</b> . . . . .	<b>xix</b>
	Table 1: Notice Icons . . . . .	xxi
	Table 2: Text and Syntax Conventions . . . . .	xxii
<b>Chapter 1</b>	<b>Overview</b> . . . . .	<b>25</b>
	Table 3: Features Supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 in a Chassis Cluster . . . . .	31
	Table 4: Chassis Cluster Feature Support on SRX Series Devices . . . . .	48
	Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster . . . . .	51
	Table 6: Chassis Cluster Feature Support on SRX5800, SRX5600, SRX5400, and SRX4600 Devices . . . . .	72
<b>Chapter 2</b>	<b>Setting Up a Chassis Cluster</b> . . . . .	<b>81</b>
	Table 7: Slot Numbering Offsets . . . . .	86
	Table 8: Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming . . . . .	88
	Table 9: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX4600 Devices . . . . .	90
	Table 10: Chassis Cluster Interface Renumbering for SRX4600 . . . . .	91
	Table 11: Chassis Cluster Fabric Interface Details for SRX4600 . . . . .	91
	Table 12: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX4100 and SRX4200 Devices . . . . .	92
	Table 13: Chassis Cluster Interface Renumbering for SRX4100 and SRX4200 . . . . .	93
	Table 14: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX5000-Line Devices . . . . .	93
	Table 15: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX1400, SRX3400, and SRX3600 . . . . .	95
	Table 16: Supported Fabric Interface Types for SRX Series Devices . . . . .	121
	Table 17: Example of Redundancy Groups in a Chassis Cluster . . . . .	137
	Table 18: Maximum Number of Redundant Ethernet Interfaces Allowed (SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500) . . . . .	142
	Table 19: Maximum Number of Redundant Ethernet Interfaces Allowed (SRX100, SRX210, SRX220, SRX240, SRX550, and SRX650) . . . . .	142
	Table 20: SRX Series Services Gateways Interface Renumbering . . . . .	152
	Table 21: SRX Series Services Gateways Interface Settings . . . . .	154
	Table 22: SRX Series Services Gateways Interface Settings (SRX100, SRX210, SRX220, SRX240, SRX550) . . . . .	155
<b>Chapter 3</b>	<b>Configuring Redundancy and Failover in a Chassis Cluster</b> . . . . .	<b>183</b>

	Table 23: IP Monitoring Results and Failover Action . . . . .	230
	Table 24: Maximum MACs Supported for IP Monitoring on IOC2 and IOC3 . . . . .	231
<b>Chapter 4</b>	<b>Chassis Cluster Operations . . . . .</b>	<b>261</b>
	Table 25: Group and Chassis Cluster Configuration Parameters . . . . .	317
	Table 26: Chassis Cluster Configuration Parameters . . . . .	317
	Table 27: Security Zone Configuration Parameters . . . . .	318
	Table 28: Security Policy Configuration Parameters . . . . .	319
	Table 29: Group and Chassis Cluster Configuration Parameters . . . . .	332
	Table 30: Chassis Cluster Configuration Parameters . . . . .	332
	Table 31: IKE Configuration Parameters . . . . .	334
	Table 32: IPsec Configuration Parameters . . . . .	334
	Table 33: Static Route Configuration Parameters . . . . .	334
	Table 34: Security Zone Configuration Parameters . . . . .	334
	Table 35: Security Policy Configuration Parameters . . . . .	335
	Table 36: Group and Chassis Cluster Configuration Parameters . . . . .	352
	Table 37: Chassis Cluster Configuration Parameters . . . . .	352
	Table 38: Security Zone Configuration Parameters . . . . .	353
	Table 39: Security Policy Configuration Parameters . . . . .	354
<b>Chapter 5</b>	<b>Upgrading or Disabling a Chassis Cluster . . . . .</b>	<b>387</b>
	Table 40: request system software in-service-upgrade Output Fields . . . . .	391
	Table 41: ISSU Platform Support . . . . .	396
	Table 42: ISSU-Related Errors and Solutions . . . . .	401
<b>Part 1</b>	<b>Configuration Statements and Operational Commands</b>	
<b>Chapter 7</b>	<b>Operational Commands . . . . .</b>	<b>483</b>
	Table 43: show chassis cluster control-plane statistics Output Fields . . . . .	512
	Table 44: show chassis cluster data-plane interfaces Output Fields . . . . .	514
	Table 45: show chassis cluster data-plane statistics Output Fields . . . . .	516
	Table 46: show chassis cluster ethernet-switching interfaces Output Fields . . . . .	518
	Table 47: show chassis cluster ethernet-switching status Output Fields . . . . .	519
	Table 48: show chassis cluster information Output Fields . . . . .	521
	Table 49: show chassis cluster information configuration-synchronization Output Fields . . . . .	526
	Table 50: show chassis cluster information issu Output Fields . . . . .	528
	Table 51: show chassis cluster interfaces Output Fields . . . . .	530
	Table 52: show chassis cluster ip-monitoring status Output Fields . . . . .	535
	Table 53: show chassis cluster ip-monitoring status redundancy group Reason Fields . . . . .	536
	Table 54: show chassis cluster statistics Output Fields . . . . .	538
	Table 55: show chassis cluster status Output Fields . . . . .	542
	Table 56: show chassis environment Output Fields . . . . .	545
	Table 57: show chassis environment cb Output Fields . . . . .	549
	Table 58: show chassis ethernet-switch Output Fields . . . . .	552
	Table 59: show chassis fabric plane Output Fields . . . . .	556
	Table 60: show chassis fabric plane-location Output Fields . . . . .	562
	Table 61: show chassis fabric summary Output Fields . . . . .	564
	Table 62: show chassis hardware Output Fields . . . . .	567



Table 63: show chassis routing-engine Output Fields . . . . .	578
Table 64: show configuration chassis cluster traceoptions Output Fields . . . . .	581
Table 65: show interfaces (Gigabit Ethernet) Output Fields . . . . .	585
Table 66: Gigabit Ethernet IQ PIC Traffic and MAC Statistics by Interface Type . . . . .	594
Table 67: show system ntp threshold Output Fields . . . . .	598
Table 68: show security macsec connections Output Fields . . . . .	599
Table 69: show security macsec statistics Output Fields . . . . .	602
Table 70: show security mka statistics Output Fields . . . . .	605
Table 71: show security mka sessions Output Fields . . . . .	607
Table 72: show security internal-security-association Output Fields . . . . .	609
Table 73: show system license Output Fields . . . . .	610
Table 74: show vrrp Output Fields . . . . .	614



# About the Documentation

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

## 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 <https://www.juniper.net/documentation/>.

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 <https://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.xml;
    }
  }
}
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.xml; }
```

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

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

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

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

For more information about the **load** command, see [CLI Explorer](#).

## Documentation Conventions

Table 1 on page xxi 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 xxii 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>
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> <b>show chassis alarms</b>  No alarms currently active
<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:  [edit] root@# <b>set system domain-name</b> <i>domain-name</i>
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 <b>stub</b> statement at the [edit protocols <b>ospf area area-id</b>] hierarchy level.</li> <li>The console port is labeled <b>CONSOLE</b>.</li> </ul>
< > (angle brackets)	Encloses optional keywords or variables.	<b>stub</b> <default-metric <i>metric</i> >;
(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.	<b>broadcast</b>   <b>multicast</b>  ( <i>string1</i>   <i>string2</i>   <i>string3</i> )
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	<b>community name members</b> [ <b>community-ids</b> ]
Indentation and braces ( { } )	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

## GUI Conventions

Table 2: Text and Syntax Conventions (continued)

Convention	Description	Examples
<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>
> (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 <https://www.juniper.net/documentation/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 <https://www.juniper.net/documentation/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 <https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <https://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: <https://www.juniper.net/customers/support/>
- Search for known bugs: <https://prsearch.juniper.net/>
- Find product documentation: <https://www.juniper.net/documentation/>
- Find solutions and answer questions using our Knowledge Base: <https://kb.juniper.net/>
- Download the latest versions of software and review release notes: <https://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum: <https://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <https://www.juniper.net/cm/>

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

## 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 <https://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 <https://www.juniper.net/support/requesting-support.html>.



## CHAPTER 1

# Overview

- [Chassis Cluster Overview on page 25](#)
- [Chassis Cluster Features Supported on SRX Series Devices on page 30](#)
- [Chassis Cluster License Requirements on page 75](#)

### Chassis Cluster Overview

---

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

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

- [Chassis Cluster Overview on page 25](#)
- [Chassis Cluster Limitations on page 27](#)

### Chassis Cluster Overview

The Junos OS provides high availability on SRX Series device by using chassis clustering. 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.

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

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

## Benefits of Chassis Cluster

---

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

## Chassis Cluster Functionality

---

Chassis cluster functionality includes:

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

## Chassis Cluster Modes

---

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

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

## How Chassis Clustering Works?

---

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

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

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

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

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

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

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

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

### IPv6 Clustering Support

SRX Series 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.

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

## Chassis Cluster Limitations

The SRX Series devices have the following chassis cluster limitations:

### Chassis Cluster

- Group VPN is not supported.
- VRRP is not supported.
- 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.
- When an SRX Series device is operating in chassis cluster mode and encounter any IA-chip access issue in an SPC or a I/O Card (IOC), a minor FPC alarm is activated to trigger redundancy group failover.
- On SRX5400, SRX5600, and SRX5800 devices, screen statistics data can be gathered on the primary device only.
- On SRX4600, SRX5400, SRX5600, and SRX5800 devices, in large chassis cluster configurations, if more than 1000 logical interfaces are used, the cluster heartbeat

timers are recommended to be increased from the default wait time before triggering failover. In a full-capacity implementation, we recommend increasing the wait to 8 seconds by modifying **heartbeat-threshold** and **heartbeat-interval** values in the **[edit chassis cluster]** hierarchy.

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

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

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

### Flow and Processing

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

**Port-mirroring configuration error.**

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

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

**Port-mirroring configuration error**

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

- When an SRX Series device is operating in chassis cluster mode and encounter any IA-chip (IA-chip is part of Juniper SPC1 and IOC1. It has direct impact on SPC1/IOC1 control plane) access issue in an SPC or a I/O Card (IOC), a minor FPC alarm is activated to trigger redundancy group failover.
- On SRX Series devices in a chassis cluster, when two logical systems are configured, the scaling limit crosses 13,000, which is very close to the standard scaling limit of

15,000, and a convergence time of 5 minutes results. This issue occurs because multicast route learning takes more time when the number of routes is increased.

- On SRX4600, SRX5400, SRX5600, and SRX5800 devices in a chassis cluster, if the primary node running the LACP process (lacpd) undergoes a graceful or ungraceful restart, the lacpd on the new primary node might take a few seconds to start or reset interfaces and state machines to recover unexpected synchronous results. Also, during failover, when the system is processing traffic packets or internal high-priority packets (deleting sessions or reestablishing tasks), medium-priority LACP packets from the peer (switch) are pushed off in the waiting queues, causing further delay.



**NOTE:** Flowd monitoring is supported on SRX100, SRX210, SRX240, SRX550M, SRX650, 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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.

**IPv6**

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

**GPRS**

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

**MIBs**

- The Chassis Cluster MIB is not supported.

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

**See Also** • [Preparing Your Equipment for Chassis Cluster Formation on page 99](#)

**Release History Table**

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

**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 81](#)
- [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)
- [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)

**Chassis Cluster Features Supported on SRX Series Devices**

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



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

**Chassis Cluster Supported Features (SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500)**

[Table 3 on page 31](#) lists the features that are supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices in a chassis cluster.

*Table 3: Features Supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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	Yes <small>NOTE: Supported on SRX1500, SRX4100, and SRX4200 devices only.</small>	Yes <small>NOTE: Supported on SRX1500, SRX4100, and SRX4200 devices only.</small>	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>	–	–	–	–

**Table 3: Features Supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 in a Chassis Cluster (continued)**

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<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
<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

**Table 3: Features Supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 in a Chassis Cluster (continued)**

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<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
<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	No	No	No	No
<b>GPRS</b>	GPRS (transparent mode and route mode)	No	No	No	No

*Table 3: Features Supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 in a Chassis Cluster (continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
GTPv2	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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 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 SRX Series devices until Junos OS Release 12.3X48.

## Chassis Cluster-Supported Features (SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500)



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

Table 4 on page 48 lists the chassis cluster features that are supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

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

Features	Supported on SRX Series
Active/backup Routing Engine group (RG0)	Yes
Active/active data redundancy groups (RGx)	Yes
Aggregate Interfaces (link aggregation)	Yes



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

Features	Supported on SRX Series
Autorecovery of fabric link	Yes
Chassis cluster extended cluster ID	Yes
Chassis cluster formation	Yes
Encrypted control link	No
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

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

Features	Supported on SRX Series
Layer 2 transparent mode LAG	Yes
Layer 3 LAG	Yes
Local interface support (non-reth)	Yes
In-service software upgrade (ISSU)	No
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

## Chassis Cluster Supported Features (SRX5800, SRX5600, and SRX5400)

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

[Table 5 on page 51](#) lists the features that are supported on SRX5800, SRX5600, and SRX5400 devices in a chassis cluster.

**Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices 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 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices 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 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)*

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

**Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices 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

Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
	SYN-ACK-ACK proxy protection	Yes	Yes	Yes	Yes
	SYN and FIN flags	Yes	Yes	Yes	Yes
	SYN flood protection	Yes	Yes	Yes	Yes
	SYN fragment protection	Yes	Yes	Yes	Yes
	TCP address sweep	Yes	Yes	Yes	Yes
	TCP packet without flag	Yes	Yes	Yes	Yes
	Teardrop attack protection	Yes	Yes	Yes	Yes
	UDP address sweep	Yes	Yes	Yes	Yes
	UDP flood protection	Yes	Yes	Yes	Yes
	Unknown protocol	Yes	Yes	Yes	Yes
	WinNuke attack protection	Yes	Yes	Yes	Yes
<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	Not supported for SRX5000 line	Not supported for SRX5000 line	Not supported for SRX5000 line	Not supported for SRX5000 line

Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<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	—	—	—	—
	Transmission queues	Yes	Yes	Yes	Yes
<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



Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<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	—	—	—	—
<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	SRX5000 line devices only	Yes	Yes	Yes
	Ethernet, Copper (10-Mbps, 100-Mbps, or 1000-Mbps port)	Yes	Yes	Yes	Yes
	Gigabit Ethernet interface	Yes	Yes	Yes	Yes
	Promiscuous mode on Ethernet interface	Yes	Yes	Yes	Yes

**Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices 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)	Yes	Yes	Yes	Yes
	LACP (port priority) Layer 3 Mode	Yes	Yes	Yes	Yes
	LACP (port priority) Layer 2 Mode	No	No	No	No
	Layer 3 LAG on routed ports	Yes	Yes	Yes	Yes
	Static LAG (routing)	Yes	Yes	Yes	Yes
	Static LAG (switching)	No	No	No	No
	Switching mode	No	No	No	No
<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 and decryption of configuration files	Yes	Yes	Yes	Yes
	Management of account files	Yes	Yes	Yes	Yes

*Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<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 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices 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	Yes	Yes	Yes	Yes
	Express Path	SRX5000 line only	No	No	No
	Flow-based processing	Yes	Yes	Yes	Yes
	Host bound fragmented traffic	Yes	Yes	Yes	Yes
	Network processor bundling	Yes	Yes	Yes	Yes
	Packet-based processing	No	No	No	No
	Selective stateless packet-based services	No	No	No	No
<b>GPRS</b>	GPRS (transparent mode and route mode)	Yes	Yes	No	No

*Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)*

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

*Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)*

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

Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

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

**Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices 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	No	No	No	No
	Flexible Ethernet services	No	No	No	No
	IRB interface	Yes	Yes	Yes	Yes
	LLDP and LLDP-MED	No	No	No	No
	MAC limit (port security)	Yes	Yes	Yes	Yes
	Q-in-Q tunneling	No	No	No	No
	Spanning Tree Protocol	No	No	No	No
	VLAN retagging	Yes	Yes	Yes	Yes
	VLANs	Yes	Yes	Yes	Yes
<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 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

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

Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
	Source IP address translation	Yes	Yes	Yes	Yes
	Static NAT	Yes	Yes	Yes	Yes
<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

*Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)*

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<b>RPM Probe</b>	RPM probe	Yes	Yes	Yes	Yes

*Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)*

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

Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
	Virtual Router Redundancy Protocol (VRRP)				
<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
<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	Yes	Yes	Yes	Yes
<b>Stateless Firewall Filters</b>	Stateless firewall filters (ACLs)	Yes	Yes	Yes	Yes

**Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)**

Category	Feature	Active/Backup	Active/Backup Failover	Active/Active	Active/Active Failover
<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
<b>Transparent Mode</b>	Bridge domain and transparent mode	Yes	Yes	Yes	Yes
	Class of service	Yes	Yes	Yes	Yes

Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)

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

**Table 5: Features Supported on SRX5800, SRX5600, and SRX5400 Devices in a Chassis Cluster (continued)**

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

<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 is 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> IPsec in active/active chassis cluster on SRX5000 line devices has the limitation that Z-mode traffic is not supported. This limitation pertains to Junos OS Release 12.3X48 and later and must be avoided.

## Chassis Cluster-Supported Features (SRX5800, SRX5600, SRX5400, and SRX4600)

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

[Table 6 on page 72](#) lists the chassis cluster features that are supported on SRX5800, SRX5600, SRX5400 and SRX4600 devices.

**Table 6: Chassis Cluster Feature Support on SRX5800, SRX5600, SRX5400, and SRX4600 Devices**

Features	SRX5000 Line & SRX4600
Active/backup Routing Engine group (RG0)	Yes
Active/active data redundancy groups (RGx)	Yes
Aggregate Interfaces (link aggregation)	Yes
Autorecovery of fabric link	Yes



*Table 6: Chassis Cluster Feature Support on SRX5800, SRX5600, SRX5400, and SRX4600 Devices (continued)*

Features	SRX5000 Line & SRX4600
Chassis cluster extended cluster ID	Yes
Chassis cluster formation	Yes
Encrypted control link	Yes
Chassis clusters (active/backup and active/active)	Yes
Control link recovery	Yes
Control plane failover	Yes
Dampening time between back-to-back redundancy group failovers	Yes
Data plane failover	Yes
Dual control links (redundant link for failover)	Yes
Dual fabric links	Yes
IP monitoring	Yes
Flow forwarding	Yes
Graceful restart routing protocols	Yes
Graceful protocol restart for BGP	Yes
Graceful protocol restart for IS-IS	Yes
Graceful protocol restart for OSPF	Yes
Graceful Routing Engine switchover (GRES) (between nodes)	Yes
HA fabric forwarded packet reordering Interface	Yes
HA monitoring	Yes
In-band cluster upgrade (ICU)	No
Junos OS flow-based routing functionality	Yes
LACP support for Layer 3	Yes
Layer 2 Ethernet switching capability	No
Layer 2 transparent mode LAG	Yes

**Table 6: Chassis Cluster Feature Support on SRX5800, SRX5600, SRX5400, and SRX4600 Devices (continued)**

Features	SRX5000 Line & SRX4600
Layer 3 LAG	Yes
Local interface support (non-reth)	Yes
In-service Software Upgrade (ISSU)	Yes
Multicast in HA mode	Yes
Network Time Protocol (NTP) time synchronization	Yes
Point-to-Point Protocol over Ethernet (PPPoE) over redundant Ethernet interface	No
Quality of service (QoS)	Yes
Redundancy group 0 (backup for Routing Engine)	Yes
Redundancy groups 1 through 128	Yes
Redundant Ethernet interfaces	Yes
Redundant Ethernet or aggregate Ethernet interface monitoring	Yes
SPU monitoring	Yes
Synchronization—backup node configuration from primary node	Yes
Synchronization—configuration	Yes
Synchronization—Dynamic Routing Protocol (DRP)	Yes
Synchronization—policies	Yes
Synchronization— session state sync (RTO sync)	Yes
TCP support for DNS	Yes
Upstream device IP address monitoring on a backup interface	Yes
Virtual Router Redundancy Protocol (VRRP) version 3	No
WAN interfaces	No

**Related Documentation** • [Chassis Cluster Overview on page 25](#)

---

## Chassis Cluster License Requirements

---

To configure and use the Junos OS software features that require a license in chassis cluster setup, you must apply the licenses to each member of a chassis cluster setup. For more information, see the following topics:

- [Understanding Chassis Cluster Licensing Requirements on page 75](#)
- [Installing Licenses on the SRX Series Devices in a Chassis Cluster on page 75](#)
- [Verifying Licenses on an SRX Series Device in a Chassis Cluster on page 77](#)

### Understanding Chassis Cluster Licensing Requirements

There is no separate license required for chassis cluster. However, some Junos OS software features require a license to activate the feature. 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. 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 <https://www.juniper.net/in/en/contact-us/>.

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

### Installing Licenses on the SRX Series Devices in a Chassis Cluster

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 Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster” on page 110](#).
- Ensure that your SRX Series device has a connection to the Internet (if particular feature requires Internet or if (automatic) renewal of license through internet is to be used). For instructions on establishing basic connectivity, see the Getting Started Guide or Quick Start Guide for your device.

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

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

```
{primary:node0}
```

```
user@host> show chassis cluster status redundancy-group 0
```

```
Cluster ID: 9
Node          Priority      Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 1
node0         254          primary   no       no
node1         1            secondary no       no
```

Output to this command indicates that node 0 is primary and node 1 is secondary.

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

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

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

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

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

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

4. Verify the installed licenses.

For more details, see [Adding New Licenses \(CLI Procedure\)](#).

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

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


```
{primary:node0}
```


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

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



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

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



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

**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

```

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

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

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

```

Licenses details on node 1.

```

{secondary-hold:node1}
user@host> show system license
License usage:

```

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

```

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

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

```

```
License identifier: JUNOS363685
License version: 2
Valid for device: JN111AB4DAGB
Features:
  services-offload - services offload mode
                    permanent

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

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

- Related Documentation**
- [SRX Series Chassis Cluster Configuration Overview on page 81](#)
  - [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)





## CHAPTER 2

# Setting Up a Chassis Cluster

- [SRX Series Chassis Cluster Configuration Overview on page 81](#)
- [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)
- [Preparing Your Equipment for Chassis Cluster Formation on page 99](#)
- [Connecting SRX Series Devices to Create a Chassis Cluster on page 102](#)
- [Example: Setting the Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster on page 110](#)
- [Chassis Cluster Management Interfaces on page 113](#)
- [Chassis Cluster Fabric Interfaces on page 119](#)
- [Chassis Cluster Control Plane Interfaces on page 129](#)
- [Chassis Cluster Redundancy Groups on page 134](#)
- [Chassis Cluster Redundant Ethernet Interfaces on page 140](#)
- [Configuring Chassis Clustering on an SRX Series Devices on page 150](#)
- [Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Devices in a Chassis Cluster on page 168](#)
- [Conditional Route Advertisement over Redundant Ethernet Interfaces on SRX Series Devices in a Chassis Cluster on page 176](#)

## SRX Series Chassis Cluster Configuration Overview

---

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

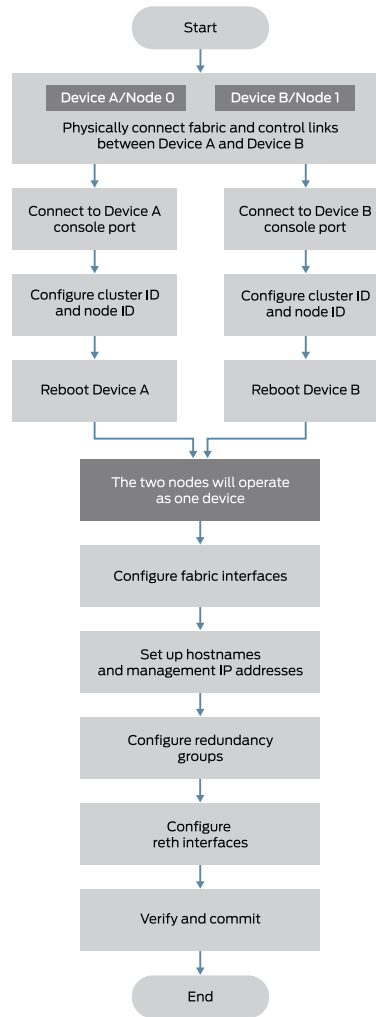
Note the following prerequisites for configuring a chassis cluster:

- On SRX300, SRX320, SRX340, SRX345, and SRX550M, 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 86](#).
- Confirm that hardware and software are the same on both devices.
- Confirm that license keys are the same on both devices.

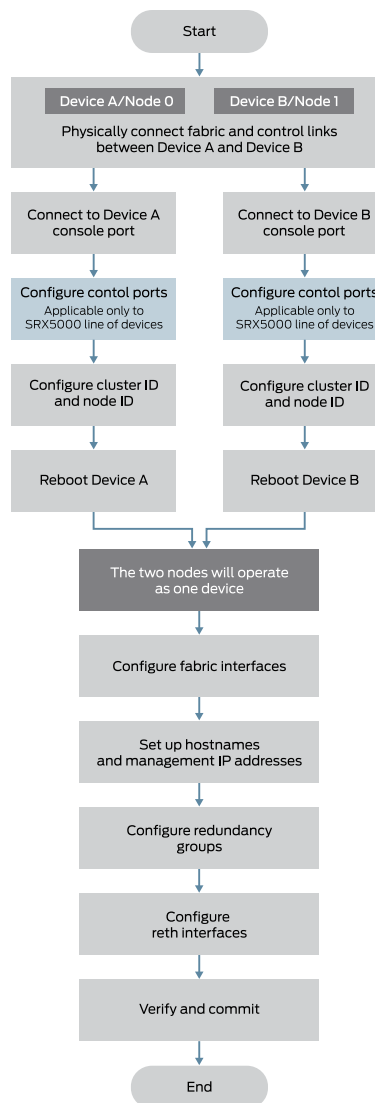
- For SRX300, SRX320, SRX340, SRX345, and SRX550M chassis clusters, the placement and type of GPIMs, XGPIMs, XPIMs, and Mini-PIMs (as applicable) must match in the two devices.
- For SRX5000 line chassis clusters, the placement and type of SPCs must match in the two devices.

Figure 1 on page 83 shows a chassis cluster flow diagram for SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, and SRX4600 devices.

*Figure 1: Chassis Cluster Flow Diagram (SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1500, and SRX4600 Devices)*



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



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

1. Prepare the SRX Series devices to be used in the chassis cluster. For more information, see [“Preparing Your Equipment for Chassis Cluster Formation” on page 99](#).
2. 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 102](#).
  - 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).

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 190](#).

- b. Configure the control ports (SRX5000 line only). See [“Example: Configuring Chassis Cluster Control Ports” on page 131](#).
3. Connect the first device to be initialized in the cluster to the console port. This is the node (node 0) that forms the cluster and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster by giving it the cluster ID.
  - b. Identify the node by giving it its own node ID and then reboot the system.See [“Example: Setting the Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster” on page 110](#). For connection instructions, see the Getting Started Guide for your device
4. Connect to the console port on the other device (node 1) and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.
5. Configure the management interfaces on the cluster. See [“Example: Configuring the Chassis Cluster Management Interface” on page 114](#).
6. Configure the cluster with the CLI. See the following topics:
  - a. *Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster*
  - b. [Example: Configuring the Chassis Cluster Fabric Interfaces on page 125](#)
  - c. [Example: Configuring Chassis Cluster Redundancy Groups on page 138](#)
  - d. [Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring on page 201](#)
  - e. [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)
7. (Optional) Initiate manual failover. See [“Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 255](#).
8. (Optional) Configure conditional route advertisement over redundant Ethernet interfaces. See [“Understanding Conditional Route Advertising in a Chassis Cluster” on page 176](#).
9. Verify the configuration. See [“Viewing a Chassis Cluster Configuration” on page 164](#).

**Note:**

- 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 190](#).
- When using dual control link functionality (SRX5600 and SRX5800 devices only), connect the two pairs of control ports that you will use on each device.

See [“Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster” on page 185](#).

For SRX5600 and SRX5800 devices, control ports must be on corresponding slots in the two devices. [Table 7 on page 86](#) shows the slot numbering offsets:

**Table 7: Slot Numbering Offsets**

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

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

**Related  
Documentation**

- [Chassis Cluster Overview on page 25](#)

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

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



**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 the devices are connected as a cluster, the slot numbering on one device changes and thus the interface numbering will change. The slot number for each slot in both nodes is determined using the following formula:

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

In chassis cluster mode, the interfaces on the secondary node are renumbered internally.

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.

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

- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Devices on page 87](#)
- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX4600 Devices on page 90](#)
- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX4100 and SRX4200 Devices on page 91](#)
- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX5800, SRX5600, and SRX5400 Devices on page 93](#)
- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX3600, SRX3400, and SRX1400 on page 95](#)
- [Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX650, SRX550, SRX240, SRX210, SRX110, and SRX100 on page 97](#)
- [FPC Slot Numbering in SRX Series Device Cards on page 98](#)

### Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 Devices

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

For SRX300, SRX320, SRX340, 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.

SRX1500 devices have 16 GE interfaces and 4 XE ports.

[Table 8 on page 88](#) 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 8: Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming**

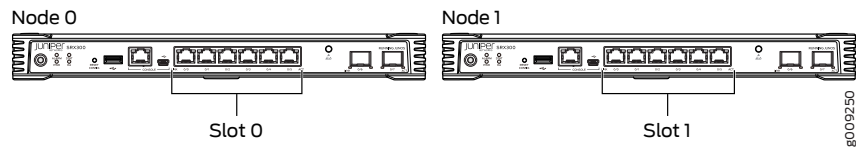
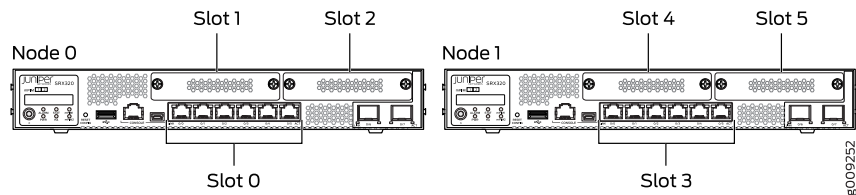
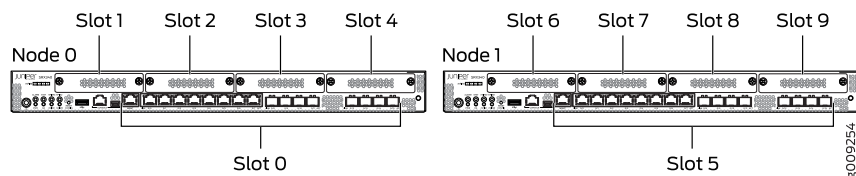
Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
1500	Node 0	1	0	fxp0	Dedicated Control port	Any Ethernet port
					em0	fab0
	Node 1		7	fxp0	Dedicated Control port	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
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

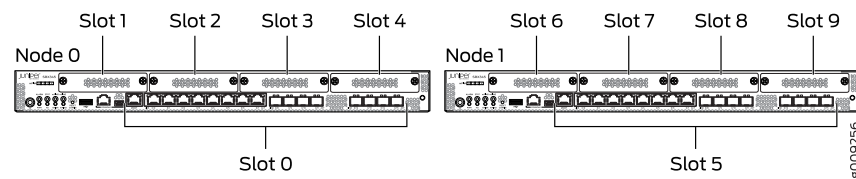
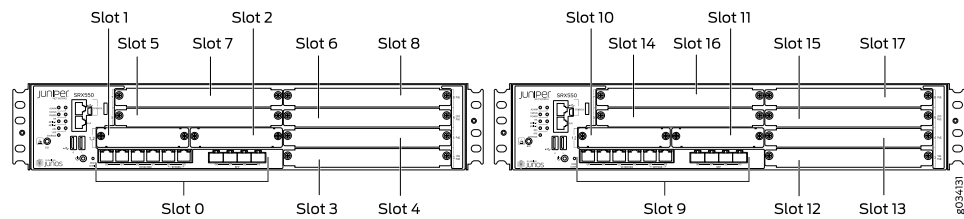
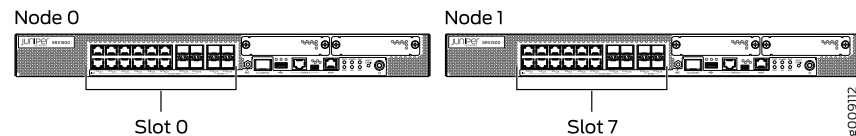


**Table 8: Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming (continued)**

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
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

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many slots. (See [Figure 3 on page 89](#), [Figure 4 on page 89](#), [Figure 5 on page 89](#), [Figure 6 on page 90](#), [Figure 7 on page 90](#), and [Figure 8 on page 90](#).)

**Figure 3: Slot Numbering in SRX300 Chassis Cluster****Figure 4: Slot Numbering in SRX320 Chassis Cluster****Figure 5: Slot Numbering in SRX340 Chassis Cluster**

*Figure 6: Slot Numbering in SRX345 Chassis Cluster**Figure 7: Slot Numbering in SRX550M Chassis Cluster**Figure 8: Slot Numbering in SRX1500 Chassis Cluster*

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

The SRX4600 devices use dedicated HA control and fabric ports. The HA dedicated interface on SRX4600 supports 10-Gigabit Ethernet Interface.

Table 9 on page 90 and Table 10 on page 91 show 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 9: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX4600 Devices*

Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX4600	Node 0	1	0-7	fxp0	Dedicated control port, em0	Dedicated fabric port, fab0
	Node 1		8-15			Dedicated fabric port, (for dual fabric-link), fab1
						40-Gigabit Ethernet port (xe)

*Table 10: Chassis Cluster Interface Renumbering for SRX4600*

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

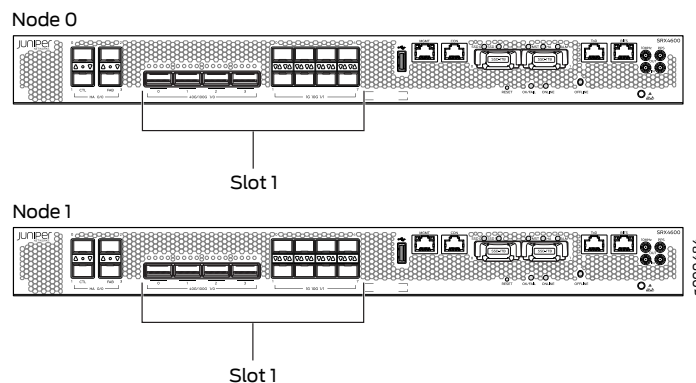
*Table 11: Chassis Cluster Fabric Interface Details for SRX4600*

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



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

Figure 9 on page 91 shows the slot numbering for both of the SRX Series devices that become node 0 and node 1 of the chassis cluster after the cluster is formed.

*Figure 9: Slot Numbering in SRX4600 Chassis Cluster*

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

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

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

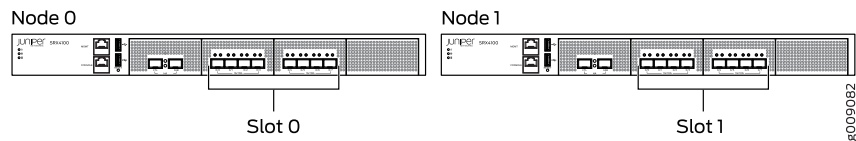
Table 12 on page 92 shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series devices that become node 0 and node 1 of the chassis cluster after the cluster is formed

**Table 12: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX4100 and SRX4200 Devices**

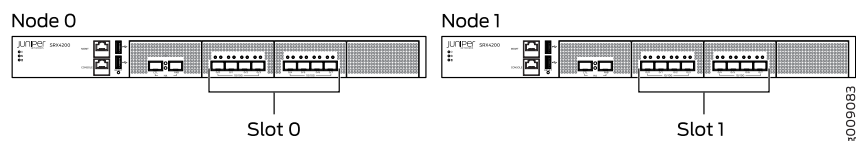
Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX4100	Node 0	1	0	fxp0	Dedicated control port, em0	Dedicated fabric port, any Ethernet port (for dual fabric-link),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

Figure 10 on page 92 and Figure 11 on page 92 shows the slot numbering for both of the SRX Series devices that become node 0 and node 1 of the chassis cluster after the cluster is formed.

**Figure 10: Slot Numbering in SRX4100 Chassis Cluster**



**Figure 11: Slot Numbering in SRX4200 Chassis Cluster**



The node 1 rennumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. For example, see [Table 13 on page 93](#) for interface renumbering on the SRX Series devices (SRX4100 and SRX4200).

**Table 13: Chassis Cluster Interface Renumbering for SRX4100 and SRX4200**

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



**NOTE:** On SRX4100 and SRX4200 devices, when the system comes up as chassis cluster, the xe-0/0/8 and xe-7/0/8 interfaces are automatically set as fabric interfaces links as shown in [Table 10 on page 91](#). You can set up another pair of fabric interfaces using any pair of 10-Gigabit interfaces to serve as the fabric between nodes. Note that, the automatically created fabric interfaces cannot be deleted. However, you can delete the second pair of fabric interfaces (manually configured interfaces).

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

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

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

[Table 14 on page 93](#) 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.

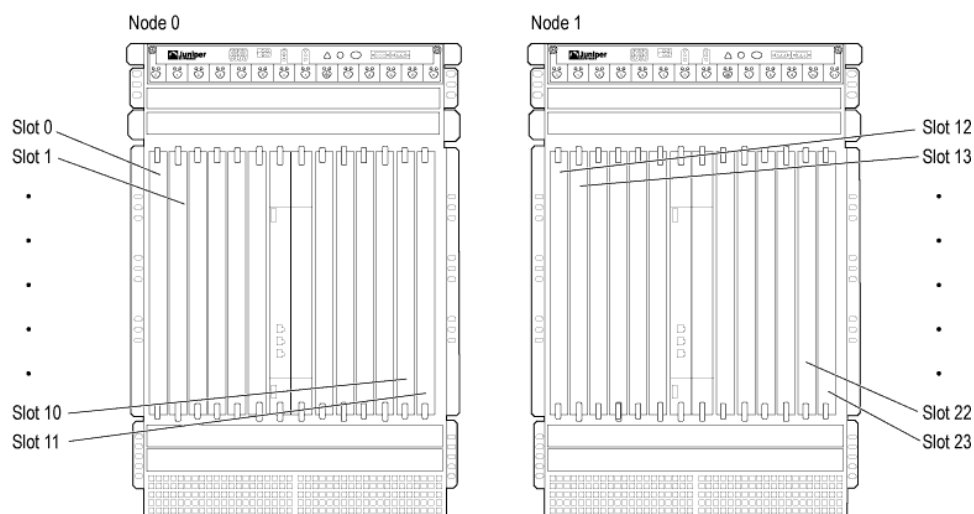
**Table 14: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX5000-Line Devices**

Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
5800	Node 0	12 (FPC slots)	0—11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1		12—23	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1

**Table 14: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX5000-Line Devices (continued)**

Model	Chassis Cluster	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
5600	Node 0	6 (FPC slots)	0—5	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1	6 (FPC slots)	6—11	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1
5400	Node 0	3 (FPC slots)	0—2	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab0
	Node 1	3 (FPC slots)	3—5	Dedicated Gigabit Ethernet port	Control port on an SPC	Any Ethernet port
				fxp0	em0	fab1

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many slots. (See [Figure 12 on page 94](#).)

**Figure 12: Slot Numbering in SRX5800 Chassis Cluster**

## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX3600, SRX3400, and SRX1400

Table 15 on page 95 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 15: Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming for SRX1400, SRX3400, and SRX3600**

Model	Chassis	Maximum Slots Per Node	Slot Numbering in a Cluster	Management Physical Port/Logical Interface	Control Physical Port/Logical Interface	Fabric Physical Port/Logical Interface
SRX3600	Node 0	13 (CFM slots)	0 — 12	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		13 — 25	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1
SRX3400	Node 0	8 (CFM slots)	0 — 7	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		8 — 15	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1
SRX1400	Node 0	4 (FPC slots)	0 — 3	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab0
	Node 1		4 — 7	Dedicated Gigabit Ethernet port	Dedicated Gigabit Ethernet port	Any Ethernet port
				fxp0	em0	fab1

Information about chassis cluster slot numbering is also provided in [Figure 13 on page 96](#), [Figure 14 on page 96](#) and [Figure 15 on page 97](#).

Figure 13: Chassis Cluster Slot Numbering for SRX3600 Devices

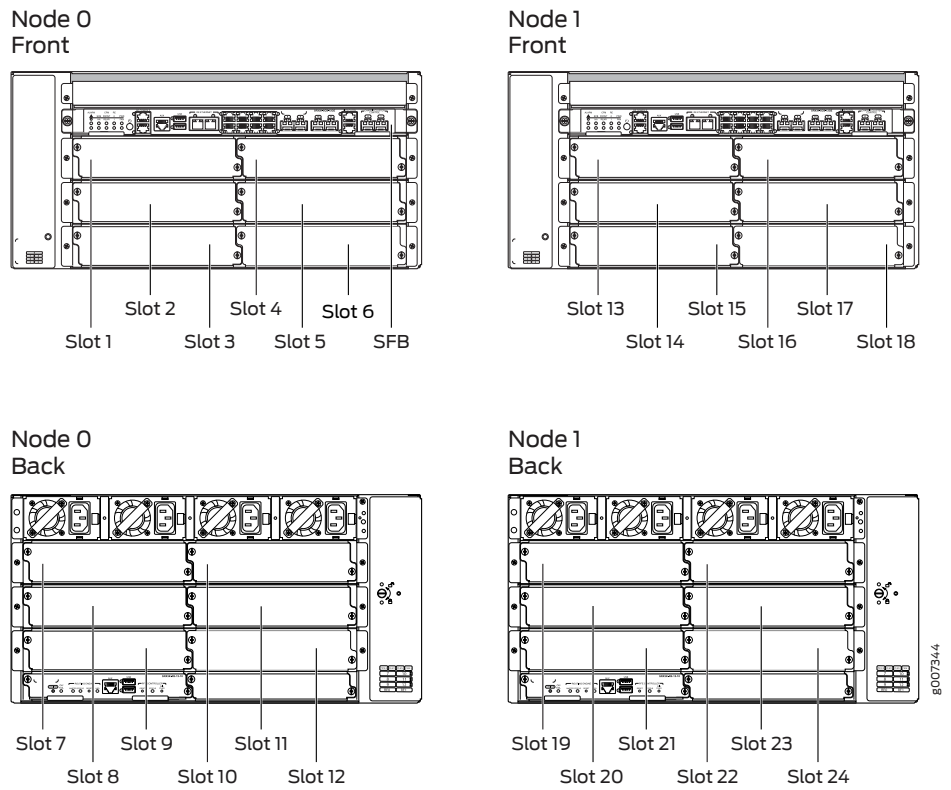


Figure 14: Chassis Cluster Slot Numbering for SRX3400 Devices

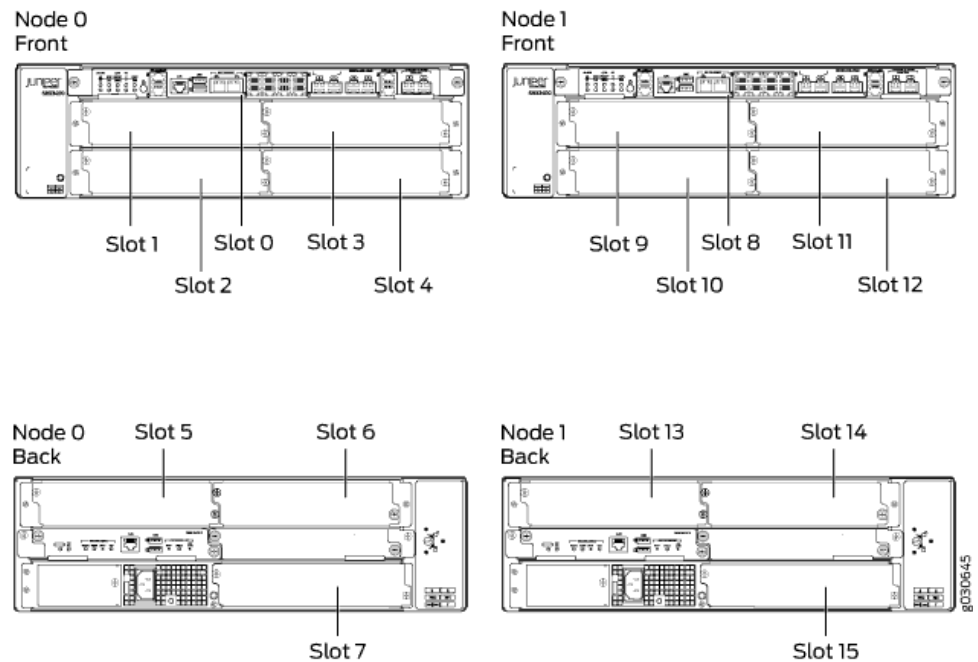
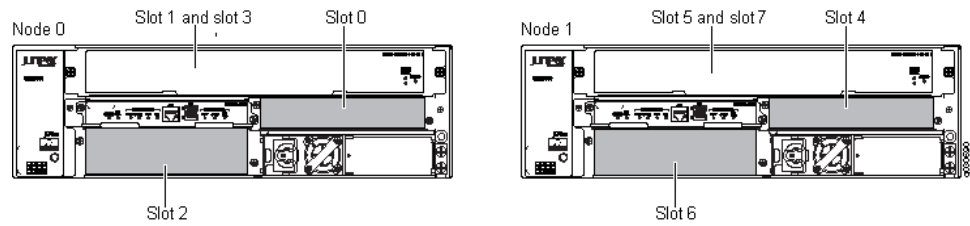




Figure 15: Chassis Cluster Slot Numbering for SRX1400 Devices



## Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming for SRX650, SRX550, SRX240, SRX210, SRX110, and SRX100

Information about chassis cluster slot numbering is also provided in [Figure 16 on page 97](#), [Figure 17 on page 97](#), [Figure 18 on page 97](#), [Figure 19 on page 97](#), [Figure 20 on page 98](#), and [Figure 21 on page 98](#).

Figure 16: Chassis Cluster Slot Numbering for SRX650 Devices

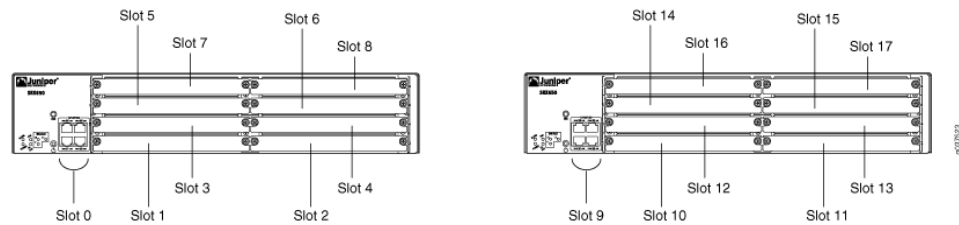


Figure 17: Chassis Cluster Slot Numbering for SRX550 Devices

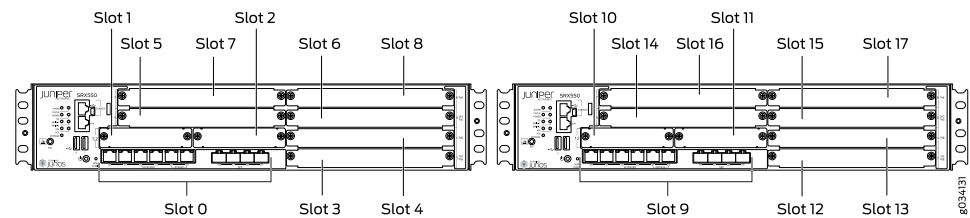


Figure 18: Chassis Cluster Slot Numbering for SRX240 Devices

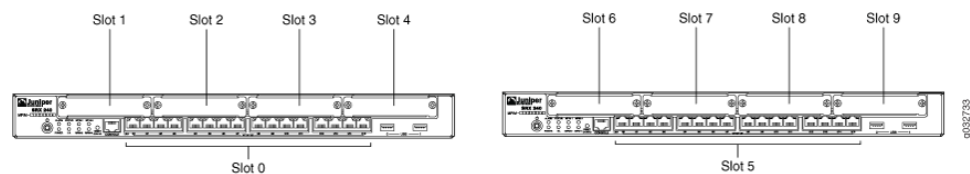


Figure 19: Chassis Cluster Slot Numbering for SRX220 Devices

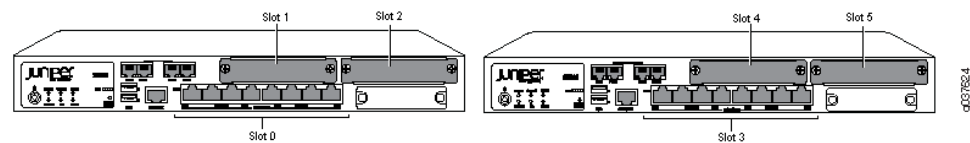


Figure 20: Chassis Cluster Slot Numbering for SRX210 Devices

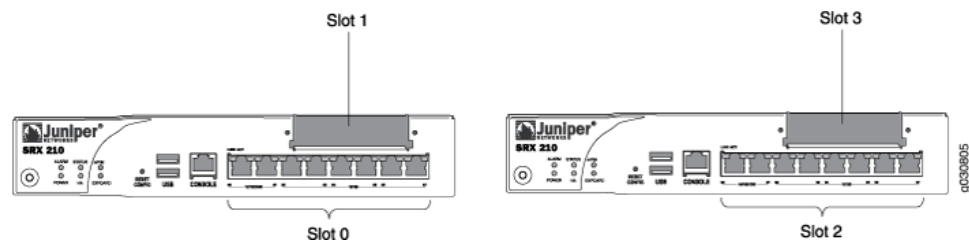
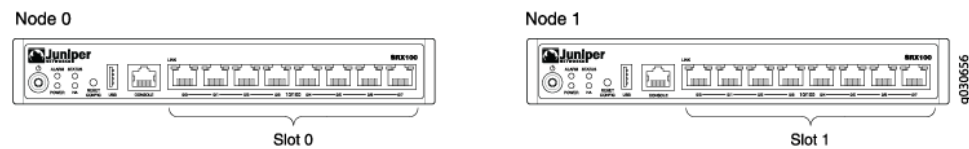


Figure 21: Chassis Cluster Slot Numbering for SRX100 Devices



**CAUTION:** Layer 2 switching must not be enabled on an SRX Series device when chassis clustering is enabled. If you have enabled Layer 2 switching, make sure you disable it before enabling chassis clustering.

The factory default configuration for SRX100, SRX210, and SRX220 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, if you use the factory default configuration for these devices, you must delete the Ethernet switching configuration before you enable chassis clustering. See [Disabling Switching on SRX100, SRX210, and SRX220 Devices Before Enabling Chassis Clustering](#).

For SRX100, SRX210, and SRX220 devices, after you enable chassis clustering and reboot the system, the built-in interface named **fe-0/0/6** is repurposed as the management interface and is automatically renamed **fxp0**.

For SRX240, SRX550, and SRX650, devices, control interfaces are dedicated Gigabit Ethernet ports. For SRX100, SRX210, and SRX220 devices, after you enable chassis clustering and reboot the system, the built-in interface named **fe-0/0/7** is repurposed as the control interface and is automatically renamed **fxp1**.

In chassis cluster mode, the interfaces on the secondary node are renumbered internally. For example, the management interface port on the front panel of each SRX210 device is still labeled **fe-0/0/6**, but internally, the node 1 port is referred to as **fe-2/0/6**.

## FPC Slot Numbering in SRX Series Device Cards

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

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

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

**slot/port module/port**

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

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

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

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

**Related  
Documentation**

- [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)

---

## Preparing Your Equipment for Chassis Cluster Formation

---

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

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

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

- For SRX5400, SRX5600 and SRX5800 chassis clusters, the placement and the type of services processing cards (SPC, SPC2), and input/output cards (IOCs, IOC2, IOC3) must match in two devices. Note that SPC and IOC are not supported on SRX5400.
- For SRX3400 and SRX3600 chassis clusters, the placement and the type of SPCs, input/output cards (IOCs, NPICs), and network processing cards (NPCs) must match in two devices.

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

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

```
user@host> show chassis hardware
node0:
```

```
-----
Hardware inventory:
```

Item	Version	Part number	Serial number	Description
Chassis			JN1251EA1AGB	SRX5600
Midplane	REV 01	760-063936	ACRE2657	Enhanced SRX5600
Midplane				
FPM Board	REV 01	710-024631	CABY3551	Front Panel Display
PEM 0	Rev 03	740-034701	QCS13380901P	PS 1.4-2.6kW; 90-264V
AC in				
PEM 1	Rev 03	740-034701	QCS133809019	PS 1.4-2.6kW; 90-264V
AC in				
Routing Engine 0	REV 02	740-056658	9009210105	SRX5k RE-1800X4
Routing Engine 1	REV 02	740-056658	9013115551	SRX5k RE-1800X4
CB 0	REV 01	750-062257	CADW3663	SRX5k SCB3
CB 1	REV 01	750-062257	CADZ3263	SRX5k SCB3
FPC 0	REV 18	750-054877	CABG6043	SRX5k SPC II
CPU		BUILTIN	BUILTIN	SRX5k DPC PPC
PIC 0		BUILTIN	BUILTIN	SPU Cp
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
FPC 1	REV 01	750-062243	CAEE5918	SRX5k IOC3 24XGE+6XLG
CPU	REV 02	711-062244	CADX8509	RMPC PMB
PIC 0		BUILTIN	BUILTIN	12x 10GE SFP+
Xcvr 0	REV 01	740-031980	273363A01891	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	273363A01915	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	ANA0BK6	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AP407GA	SFP+-10G-SR
Xcvr 9	REV 01	740-021308	MUC20G1	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	12x 10GE SFP+
PIC 2		BUILTIN	BUILTIN	3x 40GE QSFP+
PIC 3		BUILTIN	BUILTIN	3x 40GE QSFP+
WAN MEZZ	REV 15	750-049136	CAEE5845	MPC5E 24XGE OTN Mezz
FPC 3	REV 11	750-043157	CACL7452	SRX5k IOC II
CPU	REV 04	711-043360	CACP1977	SRX5k MPC PMB
MIC 0	REV 04	750-049488	CABL4759	10x 10GE SFP+
PIC 0		BUILTIN	BUILTIN	10x 10GE SFP+
Xcvr 0	REV 01	740-021308	CF36KM0SY	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	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	AN40MVV	SFP+-10G-SR
Xcvr 2	REV 01	740-021308	CF36KM37B	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AD153830DSZ	SFP+-10G-SR
MIC 1	REV 01	750-049487	CABB5961	2x 40GE QSFP+
PIC 2		BUILTIN	BUILTIN	2x 40GE QSFP+
Xcvr 1	REV 01	740-032986	QB160513	QSFP+-40G-SR4
FPC 5	REV 02	750-044175	ZY2569	SRX5k SPC II
CPU		BUILTIN	BUILTIN	SRX5k DPC PPC
PIC 0		BUILTIN	BUILTIN	SPU Flow
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
Fan Tray				Enhanced Fan Tray

- SRX1500—Has dedicated slots for each kind of card that cannot be interchanged.

- SRX4600—Has dedicated slots for each kind of card that cannot be interchanged.
- SRX300, SRX320, SRX340, SRX345, and SRX550M—Although the devices must be of the same type, they can contain different Physical Interface Modules (PIMs).
- The control ports on the respective nodes are connected to form a control plane that synchronizes the configuration and kernel state to facilitate the high availability of interfaces and services.
- The data plane on the respective nodes is connected over the fabric ports to form a unified data plane. The fabric link allows for the management of cross-node flow processing and for the management of session redundancy.



---

**NOTE:**

For SRX210 Services Gateways, the base and enhanced versions of a model can be used to form a cluster. For example:

- SRX210B and SRX210BE
- SRX210H and SRX210HE

However, the following combinations cannot be used to form a cluster:

- SRX210B and SRX210H
  - SRX210B and SRX210HE
  - SRX210BE and SRX210H
  - SRX210BE and SRX210HE
- 

**Related Documentation**

- [Chassis Cluster Overview on page 25](#)
- [Understanding Chassis Cluster Fabric Interfaces on page 119](#)

---

## Connecting SRX Series Devices to Create a Chassis Cluster

---

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

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

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

You must use the following ports to form the control link on the following 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.

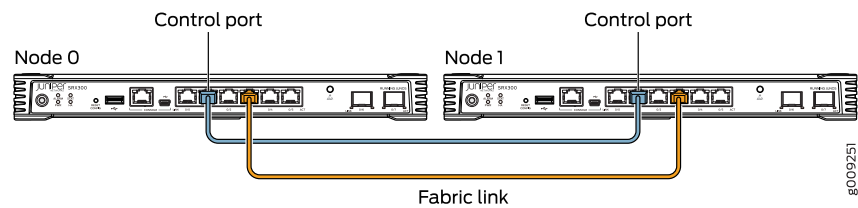
- For SRX340 and SRX345 devices, connect the ge-0/0/1 on node 0 to the ge-5/0/1 on node 1.
- For SRX550M devices, connect the ge-0/0/1 on node 0 to the ge-9/0/1 on node 1.
- "For SRX1500 devices, connect the HA control port on node 0 to the HA control port on node 1

To establish a fabric link:

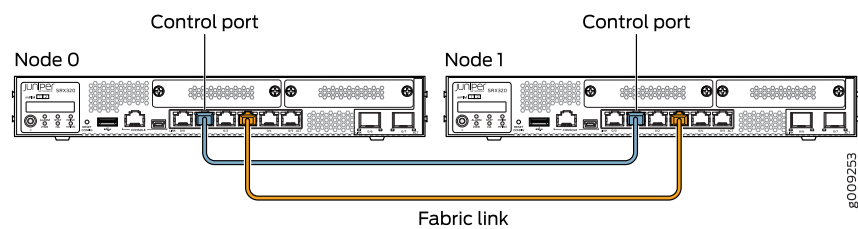
- For SRX300 and SRX320 devices, connect any interface except ge-0/0/0 and ge-0/0/1.
- For SRX340 and SRX345 devices, connect any interface except fxp0 and ge-0/0/1.

[Figure 22 on page 103](#), [Figure 23 on page 103](#), [Figure 24 on page 103](#), [Figure 25 on page 104](#), [Figure 26 on page 104](#), and [Figure 27 on page 104](#) show pairs of SRX Series devices with the fabric links and control links connected.

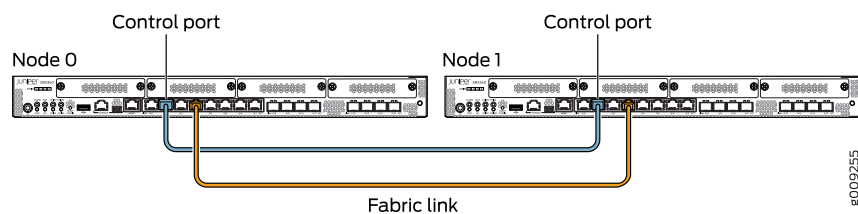
*Figure 22: Connecting SRX300 Devices in a Chassis Cluster*

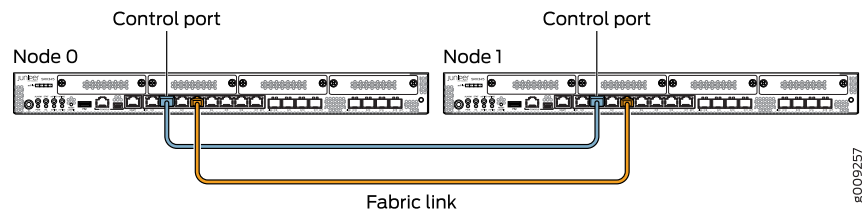


*Figure 23: Connecting SRX320 Devices in a Chassis Cluster*

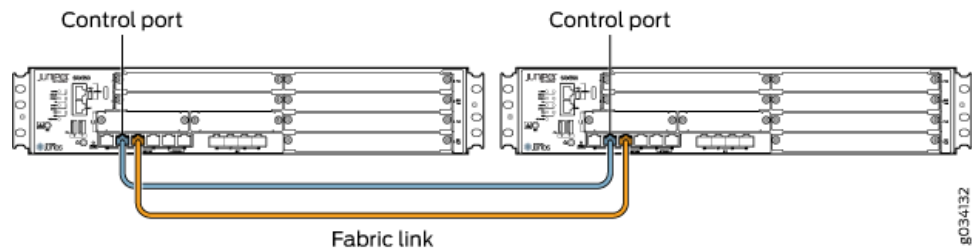


*Figure 24: Connecting SRX340 Devices in a Chassis Cluster*

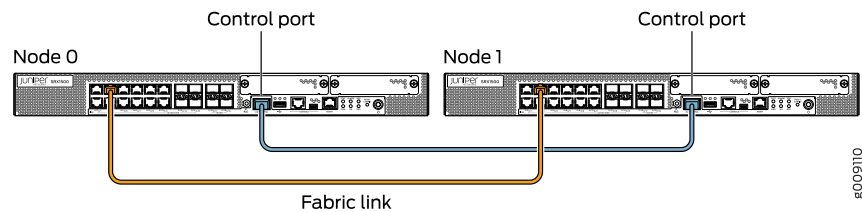


*Figure 25: Connecting SRX345 Devices in a Chassis Cluster*

8009257

*Figure 26: Connecting SRX550M Devices in a Chassis Cluster*

8034132

*Figure 27: Connecting SRX1500 Devices in a Chassis Cluster*

8009110

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



**NOTE:** You can connect two control links (SRX1400, SRX4600, SRX5000 and SRX3000 lines only) and two fabric links between the two devices in the cluster to reduce the chance of control link and fabric link failure. See [“Understanding Chassis Cluster Dual Control Links” on page 183](#) and [“Understanding Chassis Cluster Dual Fabric Links” on page 190](#).

Figure 28 on page 105, Figure 29 on page 105 and Figure 30 on page 105 show pairs of SRX Series devices with the fabric links and control links connected.



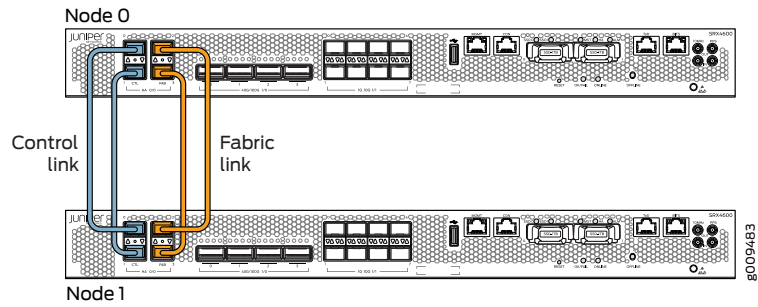
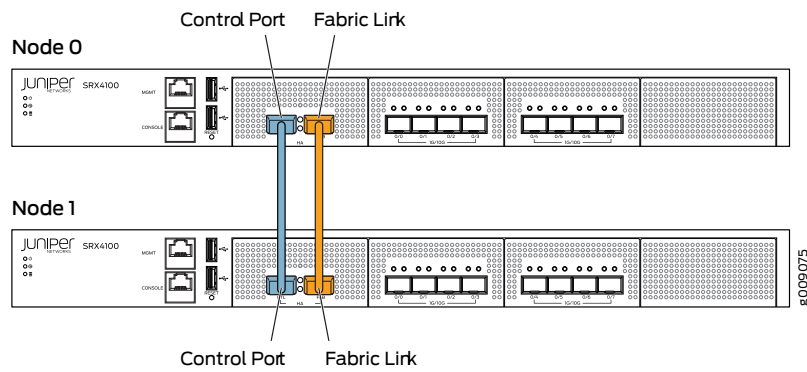
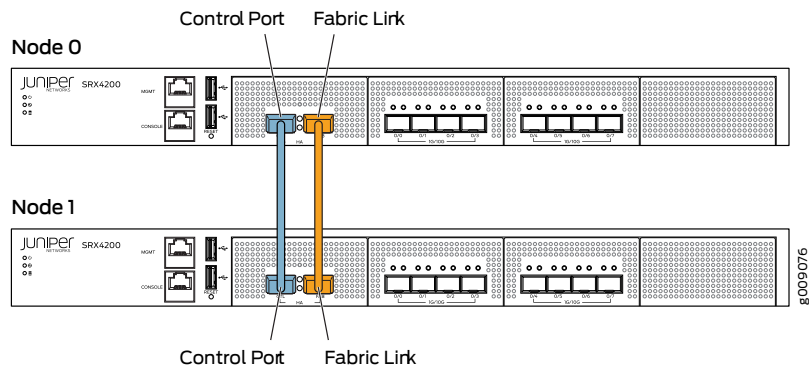
*Figure 28: Connecting SRX4600 Devices in a Chassis Cluster**Figure 29: Connecting SRX4100 Devices in a Chassis Cluster**Figure 30: Connecting SRX4200 Devices in a Chassis Cluster*

Figure 31 on page 106, Figure 32 on page 106, and Figure 33 on page 107 show pairs of SRX Series devices with the fabric links and control links connected.

Figure 31: Connecting SRX5800 Devices in a Chassis Cluster

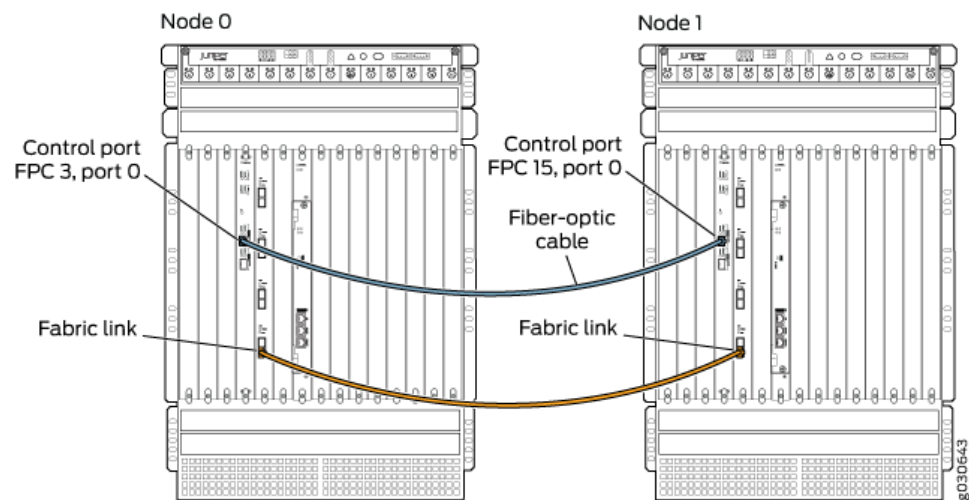
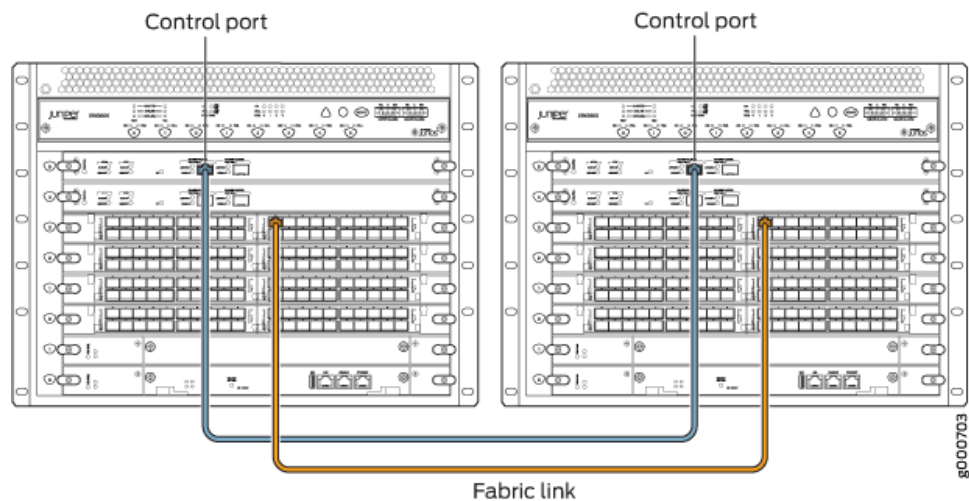


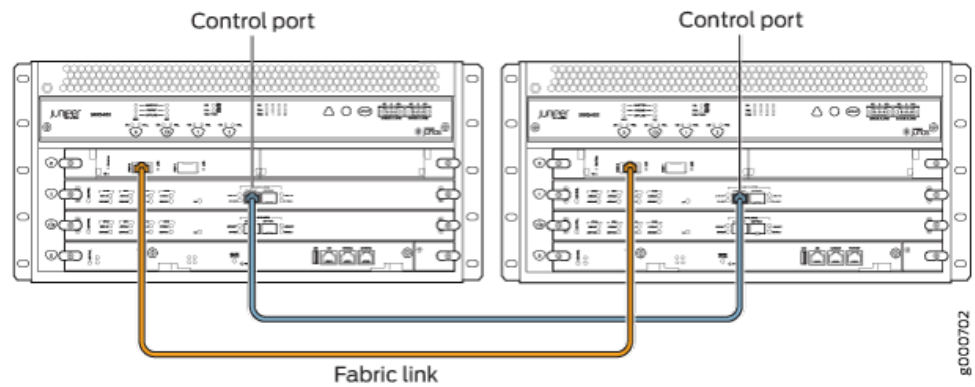
Figure 32: Connecting SRX5600 Devices in a Chassis Cluster



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

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

Figure 33: Connecting SRX5400 Devices in a Chassis Cluster



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

Figure 34 on page 107, Figure 35 on page 107 and Figure 36 on page 108 show pairs of SRX Series devices with the fabric links and control links connected.

Figure 34: Connecting SRX3600 Devices in a Chassis Cluster

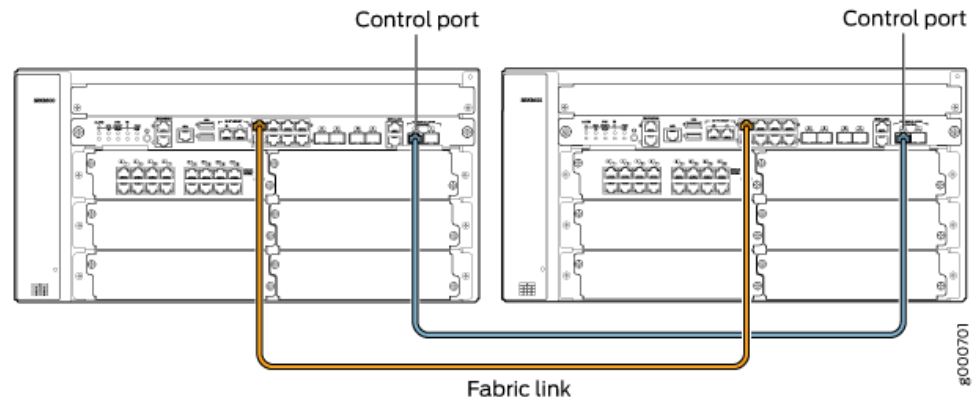
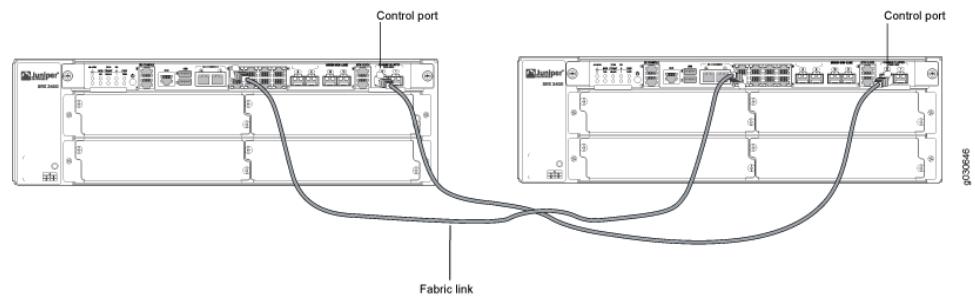


Figure 35: Connecting SRX3400 Devices in a Chassis Cluster





**NOTE:** For dual control links on SRX3000 line devices, the Routing Engine must be in slot 0 and the SRX Clustering Module (SCM) in slot 1. The opposite configuration (SCM in slot 0 and Routing Engine in slot 1) is not supported.

*Figure 36: Connecting SRX1400 Devices in a Chassis Cluster*

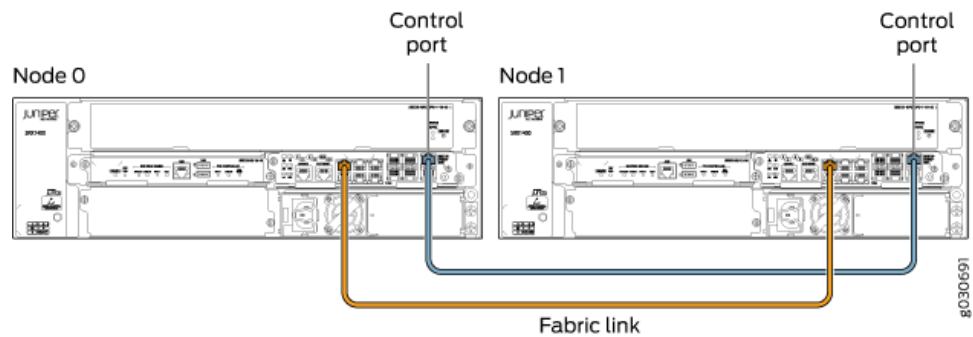
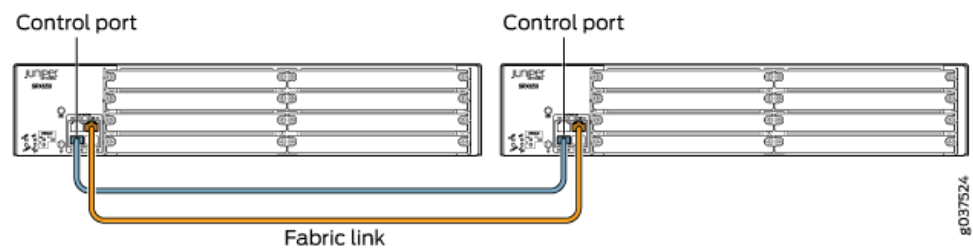
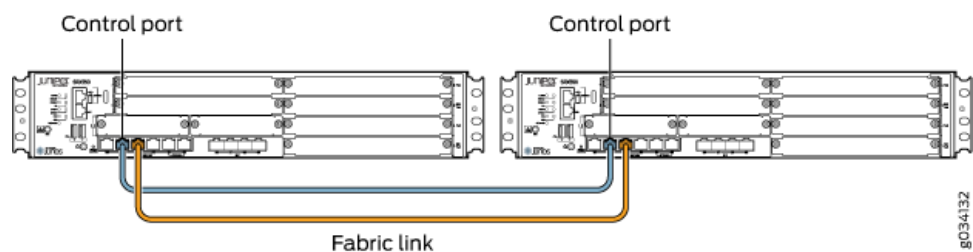


Figure 37 on page 108, Figure 38 on page 108, Figure 39 on page 108, Figure 40 on page 109, Figure 41 on page 109, Figure 42 on page 109 and Figure 43 on page 109 all show pairs of SRX Series devices with the fabric links and control links connected.

*Figure 37: Connecting SRX650 Devices in a Chassis Cluster*



*Figure 38: Connecting SRX550 Devices in a Chassis Cluster*



*Figure 39: Connecting SRX240 Devices in a Chassis Cluster*

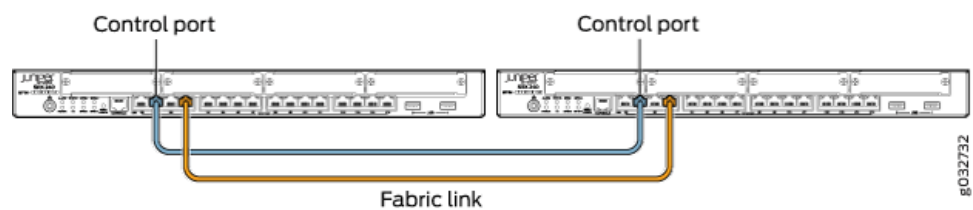


Figure 40: Connecting SRX220 Devices in a Chassis Cluster

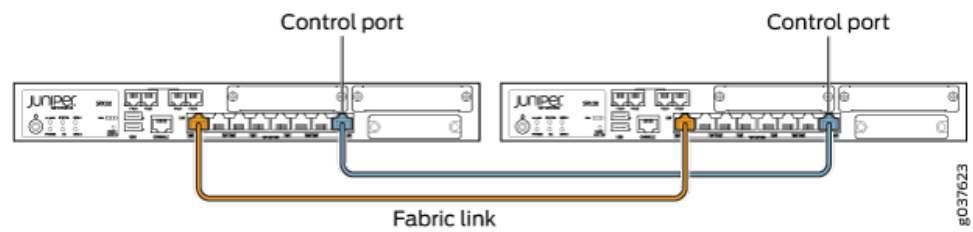


Figure 41: Connecting SRX210 Devices in a Chassis Cluster

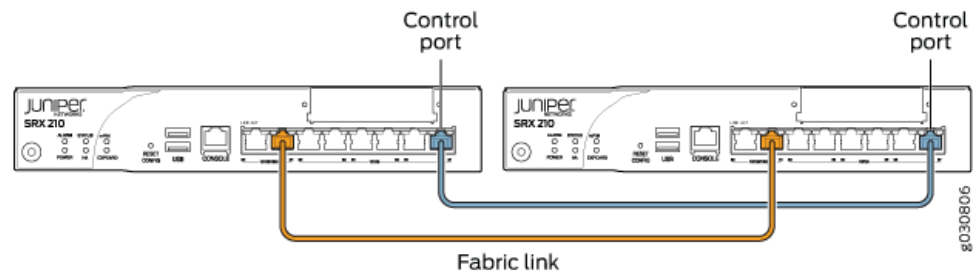


Figure 42: Connecting SRX110 Devices in a Chassis Cluster

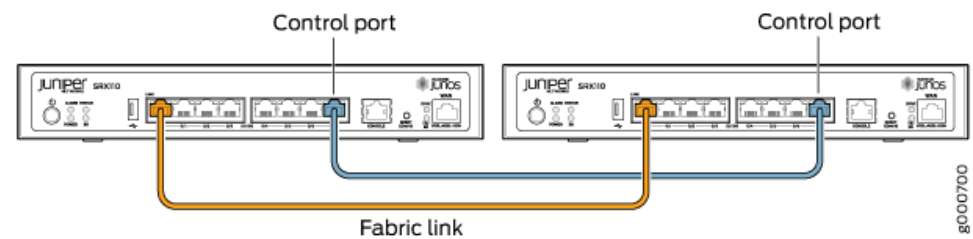
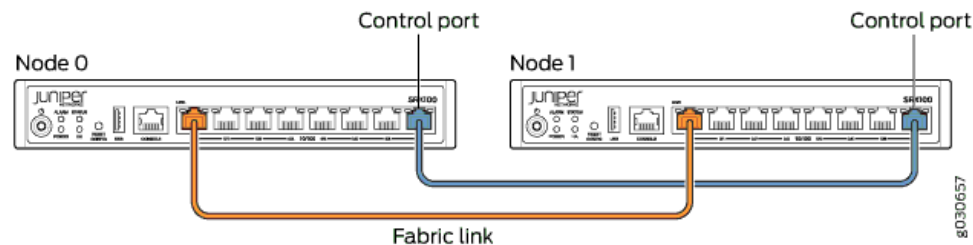


Figure 43: Connecting SRX100 Devices in a Chassis Cluster



The fabric link connection for the SRX100 must be a pair of Fast Ethernet interfaces and for the SRX210 must be a pair of either Fast Ethernet or Gigabit Ethernet interfaces. The fabric link connection must be any pair of either Gigabit Ethernet or 10-Gigabit Ethernet interfaces on all SRX Series devices.

#### Related Documentation

- [SRX Series Chassis Cluster Configuration Overview on page 81](#)
- [Example: Setting the Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster on page 110](#)
- [Example: Configuring the Chassis Cluster Management Interface on page 114](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster](#)

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

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

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

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

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.

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 110](#)
- [Overview on page 111](#)
- [Configuration on page 111](#)
- [Verification on page 112](#)

## Requirements

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

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



**NOTE:** The factory-default configuration of an SRX Series device includes the configuration of the interfaces on the device. Therefore, before enabling chassis clustering on the device, you must remove any existing configuration associated with those interfaces that will be transformed into the control and fabric interfaces. See [“Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming”](#) on page 86 for more information.

## Overview

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

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



**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 166.

## Configuration

### Step-by-Step Procedure

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

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

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

 Successfully enabled chassis cluster. Going to reboot now.
2. Connect to the second device through the console port.  

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

 Successfully enabled chassis cluster. Going to reboot now.



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

To do this, you connect to the console port on the primary device, give it a node ID, and identify the cluster it will belong to, and then reboot the system. You then connect the console port to the other device, give it a node ID, and assign it the same cluster ID you gave to the first node, and then reboot the system. In both instances, you can cause the system to boot automatically by including the `reboot` parameter in the CLI command line. (For further explanation of primary and secondary nodes, see [“Understanding Chassis Cluster Redundancy Groups” on page 134.](#))

## 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
```

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

**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 81](#)
- [Example: Configuring the Chassis Cluster Management Interface on page 114](#)
- [Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster](#)



## Chassis Cluster Management Interfaces

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

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

- [Understanding Management Interface on an Active Chassis Cluster on page 113](#)
- [Example: Configuring the Chassis Cluster Management Interface on page 114](#)

### Understanding Management Interface on an Active Chassis Cluster

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

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

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

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.



**NOTE:** For some SRX Series devices, such as the SRX100 and SRX200 line devices, do not have a dedicated port for fxp0. For SRX100, SRX210, the fxp0 interface is repurposed from a built-in interface.

## Example: Configuring the Chassis Cluster Management Interface

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

- [Requirements on page 114](#)
- [Overview on page 114](#)
- [Configuration on page 114](#)
- [Verification on page 118](#)

---

### Requirements

Before you begin, set the chassis cluster node ID and cluster ID. See [“Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 110](#).

---

### 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

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

#### **CLI Quick Configuration**

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

To configure a chassis cluster management interface for IPv4:

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

**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
```

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

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

```

        family inet {
            address 10.1.1.2/24;
        }
    }
}

```

```

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

```

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

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

**Purpose** Verify the chassis cluster management interface configuration.

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

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

**Meaning** The output displays the management interface information with their status.

*Configuring the Chassis Cluster Management Interface with IPv6 Addresses*

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

To configure a chassis cluster management interface for IPv6:

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

**Step-by-Step Procedure** To configure a chassis cluster management interface for IPv6:

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

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

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

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

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

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

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

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

```

    }
  }
}
}
node1 {
  system {
    host-name node1-router;
  }
  interfaces {
    fxp0 {
      unit 0 {
        family inet {
          address 2001:db8:1::3/32;
        }
      }
    }
  }
}

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

```

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

## Verification

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

**Purpose** Verify the chassis cluster management interface configuration.

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

```

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

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

```

```

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

```

```

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

```

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

fxp0 {
    unit 0 {
        family inet {
            address 2001:db8:1::3/32;
        }
    }
}
```

**Meaning** The output displays the management interface information with their status.

**See Also** • [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)

**Related Documentation** • [Chassis Cluster Fabric Interfaces on page 119](#)  
• [Chassis Cluster Control Plane Interfaces on page 129](#)

## Chassis Cluster Fabric Interfaces

**Supported Platforms** SRX Series, vSRX

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

- [Understanding Chassis Cluster Fabric Interfaces on page 119](#)
- [Example: Configuring the Chassis Cluster Fabric Interfaces on page 125](#)
- [Verifying Chassis Cluster Data Plane Interfaces on page 127](#)
- [Viewing Chassis Cluster Data Plane Statistics on page 127](#)
- [Clearing Chassis Cluster Data Plane Statistics on page 128](#)

## Understanding Chassis Cluster Fabric Interfaces

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

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

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

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

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



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

- [Supported Fabric Interface Types for SRX Series Devices \(SRX300 Series, SRX550M, SRX1500, SRX4100/SRX4200, and SRX5000 Series\) on page 120](#)
- [Supported Fabric Interface Types for SRX Series Devices \(SRX650, SRX550, SRX240, SRX210, and SRX100 Devices\) on page 121](#)
- [Jumbo Frame Support on page 121](#)
- [Understanding Fabric Interfaces on SRX5000 Line Devices for IOC2 and IOC3 on page 121](#)
- [Understanding Session RTOs on page 122](#)
- [Understanding Data Forwarding on page 123](#)
- [Understanding Fabric Data Link Failure and Recovery on page 124](#)

### **Supported Fabric Interface Types for SRX Series Devices (SRX300 Series, SRX550M, SRX1500, SRX4100/SRX4200, and SRX5000 Series)**

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

- For SRX300, SRX320, SRX340, and SRX345 devices, the fabric link can be any pair of Gigabit Ethernet interfaces.
- For SRX Series chassis clusters made up of SRX550M devices, SFP interfaces on Mini-PIMs cannot be used as the fabric link.
- 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.
- Supported fabric interface types for SRX4100 and SRX4200 devices are 10-Gigabit Ethernet (xe) (10-Gigabit Ethernet Interface SFP+ slots).
- Supported fabric interface types for SRX4600 devices are 40-Gigabit Ethernet (xe) (40-Gigabit Ethernet Interface QSFP slots).
- Supported fabric interface types supported for SRX5000 line devices are:



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



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

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 86.

#### Supported Fabric Interface Types for SRX Series Devices (SRX650, SRX550, SRX240, SRX210, and SRX100 Devices)

For SRX100, SRX210, SRX220, SRX240, SRX550, and SRX650 devices, the fabric link can be any pair of Gigabit Ethernet interfaces or Fast Ethernet interfaces (as applicable). Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

Table 16 on page 121 shows the fabric interface types that are supported for SRX Series devices.

**Table 16: Supported Fabric Interface Types for SRX Series Devices**

SRX650 and SRX550	SRX240	SRX220	SRX210	SRX100
Fast Ethernet	Fast Ethernet		Fast Ethernet	Fast Ethernet
Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	

#### Jumbo Frame Support

The fabric data link does not support fragmentation. To accommodate this state, jumbo frame support is enabled by default on the link with an MTU size of 9014 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 Fabric Interfaces on SRX5000 Line Devices for IOC2 and IOC3

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

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

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

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

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

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

**WARNING:**

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

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

---

### Understanding Session RTOs

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

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

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

RTOs for synchronizing a session include:

- Session creation RTOs on the first packet
- Session deletion and age-out RTOs
- Change-related RTOs, including:
  - TCP state changes
  - Timeout synchronization request and response messages
  - RTOs for creating and deleting temporary openings in the firewall (pinholes) and child session pinholes

### Understanding Data Forwarding

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

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

This traversal is required in the following situations:

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

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

## Understanding Fabric Data Link Failure and Recovery

---



**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.

---



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

---

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

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

**NOTE:**

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

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

- See Also**
- [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)
  - [Understanding Chassis Cluster Dual Fabric Links on page 190](#)

## Example: Configuring the Chassis Cluster Fabric Interfaces

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 125](#)
- [Overview on page 125](#)
- [Configuration on page 126](#)
- [Verification on page 127](#)

### Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See [“Example: Setting the Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster” on page 110](#).

### 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

```

## Verifying Chassis Cluster Data Plane Interfaces

**Purpose** Display chassis cluster data plane interface status.

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

```

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

```

## Viewing Chassis Cluster Data Plane Statistics

**Purpose** Display chassis cluster data plane statistics.

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

```

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

```

Services Synchronized:		
Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	0	0
Session create	0	0
Session close	0	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RTSP ALG	0	0

## Clearing Chassis Cluster Data Plane Statistics

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

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

Cleared data-plane statistics

**See Also** • [Configuring Chassis Clustering on an SRX Series Devices on page 150](#)

### Release History Table

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

**Related Documentation** • [SRX Series Chassis Cluster Configuration Overview on page 81](#)  
 • [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)



## Chassis Cluster Control Plane Interfaces

---

SRX Series devices in a chassis cluster use the control plane to synchronize the kernel state between the two Routing Engines. The control interfaces provide the link between the two nodes in the cluster, which are used for by devices' control planes communicate for the session state, the configuration file, and liveness signals across the nodes.

- [Understanding Chassis Cluster Control Plane and Control Links on page 129](#)
- [Example: Configuring Chassis Cluster Control Ports on page 131](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 133](#)
- [Clearing Chassis Cluster Control Plane Statistics on page 134](#)

### Understanding Chassis Cluster Control Plane and Control Links

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.

### 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.



**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.

Control ports supported on SRX Series devices are:

- On SRX5400, SRX5600, and SRX5800 devices, by default, all control ports are disabled. Each SPC in a device has two control ports, and each device can have multiple SPCs plugged into it. To set up the control link in a chassis cluster with SRX5600 or SRX5800 devices, you connect and configure the control ports that you will use on each device (**fpc<n>** and **fpc<n>**) and then initialize the device in cluster mode.
- For SRX4600 devices, dedicated chassis cluster (HA) control ports and fabric ports are available. No control link configuration is needed for SRX4600 devices; however, you need to configure fabric link explicitly for chassis cluster deployments.
- For SRX4100 and SRX4200 devices, there are dedicated chassis cluster (HA) control ports available. No control link configuration is needed for SRX4100 and SRX4200 devices. For more information about all SRX4100 and SRX4200 ports including dedicated control and fabric link ports, see [“Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming”](#) on page 86.



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

- 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 SRX240, SRX550M, and SRX650 devices, the control link uses the **ge-0/0/1** interface.
- For SRX220 devices, the control link uses the **ge-0/0/7** interface.
- For SRX100 and SRX210 devices, the control link uses the **fe-0/0/7** interface.

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 86.

## Example: Configuring Chassis Cluster Control Ports

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

- [Requirements on page 131](#)
- [Overview on page 131](#)
- [Configuration on page 131](#)
- [Verification on page 132](#)

### Requirements

---

Before you begin:

- Understand chassis cluster control links. See [“Understanding Chassis Cluster Control Plane and Control Links”](#) on page 129.
- Physically connect the control ports on the devices. See [“Connecting SRX Series Devices to Create a Chassis Cluster”](#) on page 102.

### Overview

---

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

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

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

### Configuration

---

#### CLI Quick Configuration

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

```
{primary:node0}[edit]
```

```
set chassis cluster control-ports fpc 4 port 0
set chassis cluster control-ports fpc 10 port 0
```

**Step-by-Step Procedure** To configure control ports for use as the control link for the chassis cluster:

- Specify the control ports.
 

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

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

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

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

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

## Verification

### Verifying the Chassis Cluster Status

**Purpose** Verify the chassis cluster status.

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

```
{primary:node0}
user@host> show chassis cluster status
Cluster ID: 1
```

Node	Priority	Status	Preempt	Manual failover
Redundancy group: 0 , Failover count: 1				
node0	100	primary	no	no
node1	1	secondary	no	no
Redundancy group: 1 , Failover count: 1				
node0	0	primary	no	no
node1	0	secondary	no	no

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

**See Also**

- [Connecting SRX Series Devices to Create a Chassis Cluster on page 102](#)
- [Chassis Cluster Dual Control Links on page 183](#)

## Verifying Chassis Cluster Control Plane Statistics

**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
```

**See Also**

- [Configuring Chassis Clustering on an SRX Series Devices on page 150](#)

## Clearing Chassis Cluster Control Plane Statistics

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

- [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)
- [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)

---

## Chassis Cluster Redundancy Groups

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

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

- [Understanding Chassis Cluster Redundancy Groups on page 134](#)
- [Example: Configuring Chassis Cluster Redundancy Groups on page 138](#)

## Understanding Chassis Cluster Redundancy Groups

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 138.)

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



**CAUTION:** Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine

failover, in which case all processes running on the primary node are killed and then spawned on the new master Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

---

### Understanding Chassis Cluster Redundancy Groups 1 Through 128

---

You can configure one or more redundancy groups numbered 1 through 128, referred to as redundancy group *x*. The maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure (see “[Maximum Number of Redundant Ethernet Interfaces Allowed \(SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500\)](#)” on page 142). Each redundancy group *x* acts as an independent unit of failover and is primary on only one node at a time.

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

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

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

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

The traffic for a redundancy group is processed on the node where the redundancy group is active. Because more than one redundancy group can be configured, it is possible that the traffic from some redundancy groups is processed on one node while the traffic for other redundancy groups is processed on the other node (depending on where the redundancy group is active). Multiple redundancy groups make it possible for traffic to arrive over an ingress interface of one redundancy group and over an egress interface that belongs to another redundancy group. In this situation, the ingress and egress



interfaces might not be active on the same node. When this happens, the traffic is forwarded over the fabric link to the appropriate node.

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

Table 17 on page 137 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 17: Example of Redundancy Groups in a Chassis Cluster**

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

As the example for a chassis cluster in Table 17 on page 137 shows:

- The Routing Engine on node 0 is active because redundancy group 0 is primary on node 0. (The Routing Engine on node 1 is passive, serving as backup.)
- Redundancy group 1 is primary on node 0. Interfaces ge-1/0/0 and ge-1/3/0 belonging to redundant Ethernet interface 0 and redundant Ethernet interface 1 are active and handling traffic.
- Redundancy group 2 is primary on node 1. Interfaces ge-6/0/0 and ge-6/3/0 belonging to redundant Ethernet interface 2 and redundant Ethernet interface 3 are active and handling traffic.
- Redundancy group 3 is primary on node 0. Interfaces ge-3/0/0 and ge-3/3/0 belonging to redundant Ethernet interface 4 and redundant Ethernet interface 5 are active and handling traffic.

## Example: Configuring Chassis Cluster Redundancy Groups

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

- [Requirements on page 138](#)
- [Overview on page 138](#)
- [Configuration on page 139](#)
- [Verification on page 140](#)

---

### Requirements

Before you begin:

1. Set the chassis cluster node ID and cluster ID. See [“Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 110](#).
2. Configure the chassis cluster management interface. See [“Example: Configuring the Chassis Cluster Management Interface” on page 114](#).
3. Configure the chassis cluster fabric. See [“Example: Configuring the Chassis Cluster Fabric Interfaces” on page 125](#).

---

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

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

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

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

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

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

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

```
chassis {
  cluster {
    redundancy-group 0 {
      node 0 priority 100;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 100;
      node 1 priority 1;
      preempt;
      gratuitous-arp-count 4;
    }
  }
}
```

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

---

### Verification

#### *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
```

---

## Chassis Cluster Redundant Ethernet Interfaces

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

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

- [Understanding Chassis Cluster Redundant Ethernet Interfaces on page 140](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 145](#)

### Understanding Chassis Cluster Redundant Ethernet Interfaces

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, SRX1500, SRX4100, and SRX4200 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.



**NOTE:** For SRX5800, SRX5600, SRX5400, and SRX4600 devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a chassis cluster deployment is 4,096.



**NOTE:** For SRX100, SRX210, SRX220, SRX240, SRX550M, and SRX650 devices, the total number of logical interfaces that you can configure across all the redundant Ethernet (reth) interfaces in a chassis cluster deployment is 1,024.



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

A redundant Ethernet interface must contain, at minimum, a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces that are referred to as child interfaces of the redundant Ethernet interface (the redundant parent). If two or more child interfaces from each node are assigned to the redundant Ethernet interface, a redundant Ethernet interface link aggregation group can be formed. A single redundant Ethernet interface might include a Fast Ethernet interface from node 0 and a Fast Ethernet interface from node 1 or a Gigabit Ethernet interface from node 0 and a Gigabit Ethernet interface from node 1.

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



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

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

The maximum number of redundant Ethernet interfaces that you can configure varies, depending on the device type you are using, as shown in [Table 18 on page 142](#) and [Table 19 on page 142](#). Note that the number of redundant Ethernet interfaces configured determines the number of redundancy groups that can be configured.

**Table 18: Maximum Number of Redundant Ethernet Interfaces Allowed (SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500)**

Device	Maximum Number of reth Interfaces
SRX4600	128
SRX4100, SRX4200	128
SRX5400, SRX5600, SRX5800	128
SRX300, SRX320, SRX340, SRX345	128
SRX550M	58
SRX1500	128

**Table 19: Maximum Number of Redundant Ethernet Interfaces Allowed (SRX100, SRX210, SRX220, SRX240, SRX550, and SRX650)**

Device	Maximum Number of reth Interfaces
SRX100	8
SRX210	8
SRX220	8
SRX240	24
SRX550	58
SRX650	68

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 SRX100, SRX210, SRX220, SRX240, SRX550, SRX650, 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.

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 on SRX300, SRX320, SRX340, SRX345, and SRX550M 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 {  
    gigether-options {  
      redundant-parent reth2;  
    }  
  }  
  reth2 {  
    redundant-ether-options {  
      redundancy-group 1;  
    }  
    unit 0 {  
      family inet {  
        address 1.1.1/24;  
      }  
    }  
  }  
}
```



**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.

---

**See Also**   • [Understanding Link Aggregation Groups in a Chassis Cluster on page 261](#)



## Example: Configuring Chassis Cluster Redundant Ethernet Interfaces

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 145](#)
- [Overview on page 145](#)
- [Configuration on page 146](#)
- [Verification on page 149](#)

---

### Requirements

Before you begin:

- Understand how to set the chassis cluster node ID and cluster ID. See [“Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 110](#).
- 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 125](#).
- Understand how to set the chassis cluster node redundancy groups. See [“Example: Configuring Chassis Cluster Redundancy Groups” on page 138](#).

---

### 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

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

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

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

**Step-by-Step Procedure** To configure redundant Ethernet interfaces for IPv4:

1. Bind redundant child physical interfaces to reth1.

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

2. Bind redundant child physical interfaces to reth2.

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

3. Add reth1 to redundancy group 1.

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

4. Set the MTU size.

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



**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
```

- Associate reth1.0 to the trust security zone.

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

### *Configuring Chassis Cluster Redundant Ethernet Interfaces for IPv6 Addresses*

#### **CLI Quick Configuration**

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

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

#### **Step-by-Step Procedure**

To configure redundant Ethernet interfaces for IPv6:

- 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
```

- 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
```

- Add reth2 to redundancy group 1.

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

- Set the MTU size.

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

- Assign an IP address to reth2.

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

- Associate reth2.0 to the trust security zone.

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

**Step-by-Step  
Procedure**

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

1. Specify the number of redundant Ethernet interfaces:

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

**Results**

From configuration mode, confirm your configuration by entering the **show interfaces reth0** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

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

```
{primary:node0}[edit]
user@host# show interfaces
interfaces {
    ...
    fe-1/0/0 {
        fastether-options {
            redundant-parent reth2;
        }
    }
    fe-8/0/0 {
        fastether-options {
            redundant-parent reth2;
        }
    }
    ge-0/0/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    ge-7/0/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
    }
    unit 0 {
        family inet {
            mtu 1500;
            address 10.1.1.3/24;
        }
    }
}
```

```

    }
  }
  reth2 {
    redundant-ether-options {
      redundancy-group 1;
    }
    unit 0 {
      family inet6 {
        mtu 1500;
        address 2010:2010:201::2/64;
      }
    }
  }
  ...
}

```

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

### Verification

Confirm that the configuration is working properly.

- [Verifying Chassis Cluster Redundant Ethernet Interfaces on page 149](#)
- [Verifying Chassis Cluster Control Links on page 149](#)

#### *Verifying Chassis Cluster Redundant Ethernet Interfaces*

**Purpose** Verify the configuration of the chassis cluster redundant Ethernet interfaces.

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

```

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

e-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
[...Output truncated...]

```

#### *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

**See Also** • [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 145](#)

#### Release History Table

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

#### Related Documentation

- [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)
- [Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Devices in a Chassis Cluster on page 168](#)

## Configuring Chassis Clustering on an SRX Series Devices

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

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

- [Example: Configuring Chassis Clustering on an SRX Series Devices on page 151](#)
- [Viewing a Chassis Cluster Configuration on page 164](#)
- [Viewing Chassis Cluster Statistics on page 164](#)
- [Clearing Chassis Cluster Statistics on page 166](#)

- [Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 166](#)
- [Verifying Chassis Cluster Configuration Synchronization Status on page 167](#)

## Example: Configuring Chassis Clustering on an SRX Series Devices

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

- [Requirements on page 151](#)
- [Overview on page 152](#)
- [Configuration on page 153](#)
- [Verification on page 160](#)

### 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 86 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 rennumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. See [Table 20 on page 152](#) for interface renumbering on the SRX Series device.

**Table 20: 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



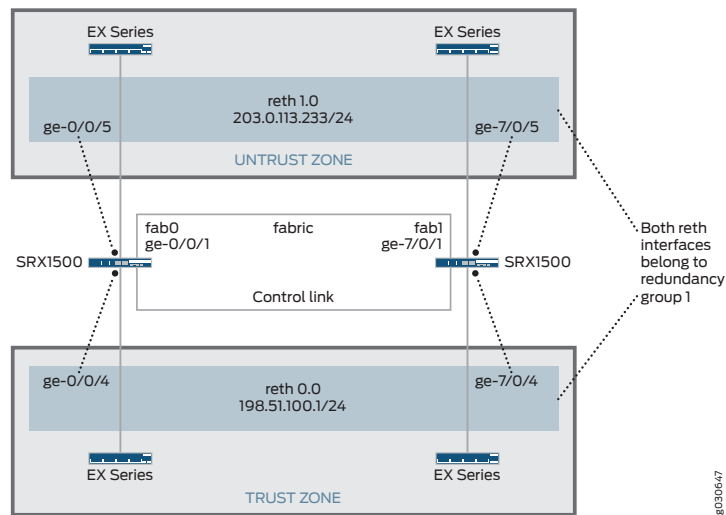
Table 20: SRX Series Services Gateways Interface Renumbering (continued)

SRX Series Services Gateway	Renumbering Constant	Node 0 Interface Name	Node 1 Interface Name
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 44 on page 153 shows the topology used in this example.

Figure 44: SRX Series Devices (SRX1500) In Chassis Cluster



## 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 gigether-options redundant-parent reth1
set interfaces ge-7/0/2 gigether-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 gigether-options redundant-parent reth0
set interfaces ge-7/0/3 gigether-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 SRX300, SRX320, SRX340, SRX345, and SRX550M device, see [Table 21 on page 154](#) for command and interface settings for your device and substitute these commands into your CLI.

**Table 21: SRX Series Services Gateways 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 21: SRX Series Services Gateways 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

Table 22: SRX Series Services Gateways Interface Settings (SRX100, SRX210, SRX220, SRx240, SRX550)

Command	SRX100	SRX210	SRX220	SRX240	SRX550
set interfaces fab0 fabric-options member-interfaces	fe-0/0/1	ge-0/0/1	ge-0/0/0 to ge-0/0/5	ge-0/0/2	ge-0/0/2
set interfaces fab1 fabric-options member-interfaces	fe-1/0/1	ge-2/0/1	ge-3/0/0 to ge-3/0/5	ge-5/0/2	ge-9/0/2
set chassis cluster redundancy-group 1 interface-monitor	fe-0/0/0 weight 255	fe-0/0/3 weight 255	ge-0/0/0 weight 255	ge-0/0/5 weight 255	ge-1/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-0/0/2 weight 255	fe-0/0/2 weight 255	ge-3/0/0 weight 255	ge-5/0/5 weight 255	ge-10/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor	fe-1/0/0 weight 255	fe-2/0/3 weight 255	ge-0/0/1 weight 255	ge-0/0/6 weight 255	ge-1/0/1 weight 255

**Table 22: SRX Series Services Gateways Interface Settings (SRX100, SRX210, SRX220, SRx240, SRX550) (continued)**

Command	SRX100	SRX210	SRX220	SRX240	SRX550
set chassis cluster redundancy-group 1 interface-monitor	fe-1/0/2 weight 255	fe-2/0/2 weight 255	ge-3/0/1 weight 255	ge-5/0/6 weight 255	ge-10/0/1 weight 255
set interfaces	fe-0/0/2 fastether-options redundant-parent reth1	fe-0/0/2 fastether-options redundant-parent reth1	ge-0/0/2 fastether-options redundant-parent reth0	ge-0/0/5 gigether-options redundant-parent reth1	ge-1/0/0 gigether-options redundant-parent reth1
set interfaces	fe-1/0/2 fastether-options redundant-parent reth1	fe-2/0/2 fastether-options redundant-parent reth1	ge-0/0/3 fastether-options redundant-parent reth1	ge-5/0/5 gigether-options redundant-parent reth1	ge-10/0/0 gigether-options redundant-parent reth1
set interfaces	fe-0/0/0 fastether-options redundant-parent reth0	fe-0/0/3 fastether-options redundant-parent reth0	ge-3/0/2 fastether-options redundant-parent reth0	ge-0/0/6 gigether-options redundant-parent reth0	ge-1/0/1 gigether-options redundant-parent reth0
set interfaces	fe-1/0/0 fastether-options redundant-parent reth0	fe-2/0/3 fastether-options redundant-parent reth0	ge-3/0/3 fastether-options redundant-parent reth1	ge-5/0/6 gigether-options redundant-parent reth0	ge-10/0/1 gigether-options redundant-parent reth0

### 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 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.

- 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 {
    gigether-options {
      redundant-parent reth1;
    }
    unit 0 {
      family inet {
        address 2.2.2.2/30;
      }
    }
  }
  ge-0/0/3 {
    gigether-options {
      redundant-parent reth0;
    }
  }
  ge-7/0/2 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  ge-7/0/3 {
    gigether-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 161](#)
- [Verifying Chassis Cluster Interfaces on page 161](#)
- [Verifying Chassis Cluster Statistics on page 161](#)



- [Verifying Chassis Cluster Control Plane Statistics on page 162](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 163](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 163](#)
- [Troubleshooting with Logs on page 164](#)

### ***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                                RT0s sent  RT0s received
  Translation context                          0           0
  Incoming NAT                                0           0
  Resource manager                             6           0
  Session create                             161          0
  Session close                               148          0
  Session change                              0           0
  Gate create                                 0           0
  Session ageout refresh requests              0           0
  Session ageout refresh replies              0           0
  IPSec VPN                                   0           0
  Firewall user authentication                 0           0
  MGCP ALG                                    0           0
  H323 ALG                                    0           0
  SIP ALG                                     0           0
  SCCP ALG                                    0           0
  PPTP ALG                                    0           0
  RPC ALG                                     0           0
  RTSP ALG                                    0           0
  RAS ALG                                     0           0
  MAC address learning                         0           0
  GPRS GTP                                    0           0
```

### *Verifying Chassis Cluster Control Plane Statistics*

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

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

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 2294
    Heartbeat packets received: 2298
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
```

Probes sent: 2290  
Probes received: 615

### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

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

Services Synchronized:

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

### *Verifying Chassis Cluster Redundancy Group Status*

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

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

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

Cluster ID: 1

Node	Priority	Status	Preempt	Manual failover
Redundancy group: 1, Failover count: 1				
node0	100	primary	no	no
node1	50	secondary	no	no

### *Troubleshooting with Logs*

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

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

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

**See Also** • [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)

## Viewing a Chassis Cluster Configuration

**Purpose** Display chassis cluster verification options.

**Action** From the CLI, enter the **show chassis cluster ?** command:

```
{primary:node1}
user@host> show chassis cluster ?
Possible completions:
  interfaces      Display chassis-cluster interfaces
  statistics      Display chassis-cluster traffic statistics
  status          Display chassis-cluster status
```

## Viewing Chassis Cluster Statistics

**Purpose** Display information about chassis cluster services and interfaces.

**Action** From the CLI, enter the **show chassis cluster statistics** command:

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

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 798
    Heartbeat packets received: 784
Fabric link statistics:
  Child link 0
    Probes sent: 793
    Probes received: 0
Services Synchronized:
  Service name          RT0s sent  RT0s received
  Translation context    0          0
  Incoming NAT           0          0
```

Resource manager	0	0
Session create	0	0
Session close	0	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RTSP ALG	0	0

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

Control link statistics:

```
Control link 0:
  Heartbeat packets sent: 258689
  Heartbeat packets received: 258684
Control link 1:
  Heartbeat packets sent: 258689
  Heartbeat packets received: 258684
```

Fabric link statistics:

```
Child link 0
  Probes sent: 258681
  Probes received: 258681
Child link 1
  Probes sent: 258501
  Probes received: 258501
```

Services Synchronized:

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	0	0
Session create	1	0
Session close	1	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

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

```
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 82371
    Heartbeat packets received: 82321
  Control link 1:
    Heartbeat packets sent: 0
    Heartbeat packets received: 0
```

## Clearing Chassis Cluster Statistics

To clear displayed information about chassis cluster services and interfaces, enter the **clear chassis cluster statistics** command from the CLI:

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

```
Cleared control-plane statistics
Cleared data-plane statistics
```

## Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes

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

**See Also** • [NTP Time Synchronization on SRX Series Devices on page 291](#)

## Verifying Chassis Cluster Configuration Synchronization Status

**Purpose** Display the configuration synchronization status of a chassis “[Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes](#)” on [page 166](#)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
```

**See Also** • [show chassis cluster information configuration-synchronization on page 526](#)

**Related Documentation**

- [Preparing Your Equipment for Chassis Cluster Formation on page 99](#)
- [Connecting SRX Series Devices to Create a Chassis Cluster on page 102](#)
- [SRX Series Chassis Cluster Configuration Overview on page 81](#)

---

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

---

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

This example shows how to enable eight-queue CoS on redundant Ethernet interfaces on SRX Series devices in a chassis cluster. This example is applicable to SRX5800, SRX5600, SRX5400, SRX4200, and SRX4100.

- [Requirements on page 168](#)
- [Overview on page 168](#)
- [Configuration on page 169](#)
- [Verification on page 175](#)

### Requirements

This example uses the following hardware and software components:

- Two SRX5600 Service Gateways in a chassis cluster
- Junos OS Release 11.4R4 or later for SRX Series Services Gateways

Before you begin:

- Understand chassis cluster configuration. See [“Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Devices” on page 299](#).
- Understand chassis cluster redundant interface configuration. See [“Example: Configuring Chassis Cluster Redundant Ethernet Interfaces” on page 145](#).

### Overview

The SRX Series devices support eight queues, but only four queues are enabled by default. Use the **set chassis fpc x pic y max-queues-per-interface 8** command to enable eight queues explicitly at the chassis level. The values of *x* and *y* depends on the location of the IOC and the PIC number where the interface is located on the device on which CoS needs to be implemented. To find the IOC location use the **show chassis fpc pic-status** or **show chassis hardware** commands.

You must restart the chassis control for the configuration to take effect.

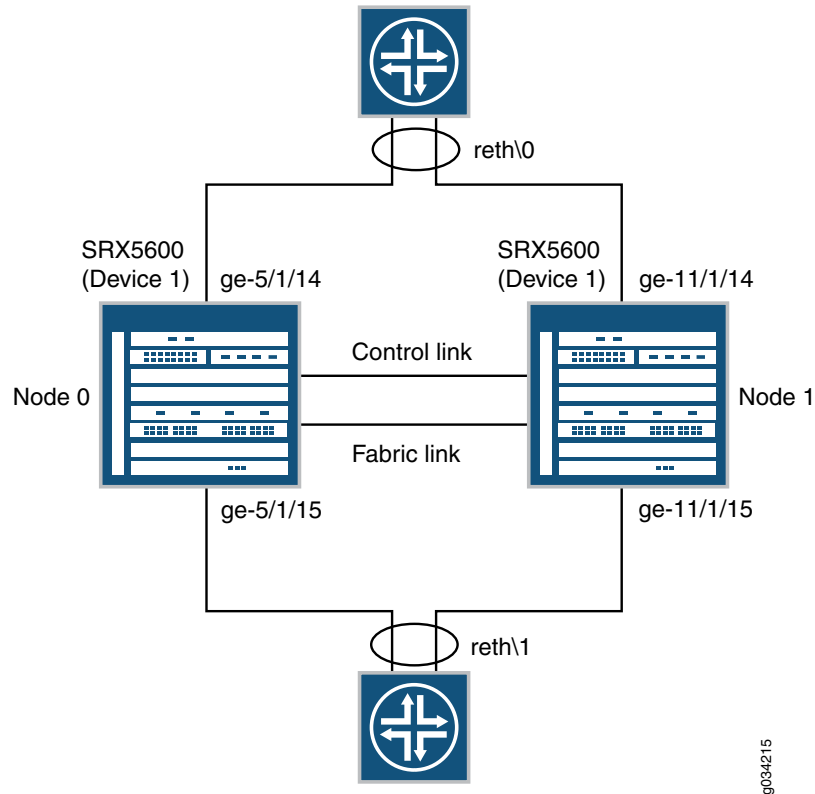




**NOTE:** On SRX Series devices, eight QoS queues are supported per ae interface.

Figure 45 on page 169 shows how to configure eight-queue CoS on redundant Ethernet interfaces on SRX Series devices in a chassis cluster.

*Figure 45: Eight-Queue CoS on Redundant Ethernet Interfaces*



## Configuration

### CLI Quick Configuration

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

```
set chassis fpc 5 pic 1 max-queues-per-interface 8
set chassis fpc 5 pic 1 max-queues-per-interface 8
set chassis cluster reth-count 2
set chassis cluster control-ports fpc 4 port 0
set chassis cluster control-ports fpc 10 port 0
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
set interfaces ge-5/1/14 together-options redundant-parent reth0
```

```
set interfaces ge-5/1/15 gigether-options redundant-parent reth1
set interfaces ge-11/1/14 gigether-options redundant-parent reth0
set interfaces ge-11/1/15 gigether-options redundant-parent reth1
set interfaces reth0 vlan-tagging
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 vlan-id 1350
set interfaces reth0 unit 0 family inet address 192.0.2.1/24
set interfaces reth1 hierarchical-scheduler
set interfaces reth1 vlan-tagging
set interfaces reth1 redundant-ether-options redundancy-group 2
set interfaces reth1 unit 0 vlan-id 1351
set interfaces reth1 unit 0 family inet address 192.0.2.2/24
set interfaces reth1 unit 1 vlan-id 1352
set interfaces reth1 unit 1 family inet address 192.0.2.3/24
set interfaces reth1 unit 2 vlan-id 1353
set interfaces reth1 unit 2 family inet address 192.0.2.4/24
set interfaces reth1 unit 3 vlan-id 1354
set interfaces reth1 unit 3 family inet address 192.0.2.5/24
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q0
  loss-priority low code-points 000
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q2
  loss-priority low code-points 010
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q3
  loss-priority low code-points 011
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q1 loss-priority
  low code-points 001
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q4
  loss-priority low code-points 100
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q5
  loss-priority low code-points 101
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q6
  loss-priority low code-points 110
set class-of-service classifiers inet-precedence inet_prec_4 forwarding-class q7
  loss-priority low code-points 111
set class-of-service forwarding-classes queue 0 q0
set class-of-service forwarding-classes queue 1 q1
set class-of-service forwarding-classes queue 2 q2
set class-of-service forwarding-classes queue 3 q3
set class-of-service forwarding-classes queue 4 q4
set class-of-service forwarding-classes queue 5 q5
set class-of-service forwarding-classes queue 6 q6
set class-of-service forwarding-classes queue 7 q7
set class-of-service traffic-control-profiles 1 scheduler-map sched_map
set class-of-service traffic-control-profiles 1 shaping-rate 200m
set class-of-service interfaces reth0 unit 0 classifiers inet-precedence inet_prec_4
set class-of-service interfaces reth1 unit 0 output-traffic-control-profile 1
set class-of-service scheduler-maps sched_map forwarding-class q0 scheduler S0
set class-of-service scheduler-maps sched_map forwarding-class q1 scheduler S1
set class-of-service scheduler-maps sched_map forwarding-class q2 scheduler S2
set class-of-service scheduler-maps sched_map forwarding-class q3 scheduler S3
set class-of-service scheduler-maps sched_map forwarding-class q4 scheduler S4
set class-of-service scheduler-maps sched_map forwarding-class q5 scheduler S5
set class-of-service scheduler-maps sched_map forwarding-class q6 scheduler S6
set class-of-service scheduler-maps sched_map forwarding-class q7 scheduler S7
set class-of-service schedulers S0 transmit-rate percent 20
set class-of-service schedulers S1 transmit-rate percent 5
```

```

set class-of-service schedulers S2 transmit-rate percent 5
set class-of-service schedulers S3 transmit-rate percent 10
set class-of-service schedulers S4 transmit-rate percent 10
set class-of-service schedulers S5 transmit-rate percent 10
set class-of-service schedulers S6 transmit-rate percent 10
set class-of-service schedulers S7 transmit-rate percent 30

```

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

To enable eight-queue CoS on redundant Ethernet interfaces:

1. Configure a maximum of eight queues on the interfaces on Node 0 and Node 1.

```

[edit chassis]
user@host# set fpc 5 pic 1 max-queues-per-interface 8

```



**NOTE:** In addition to configuring eight queues at the [edit chassis] hierarchy level, the configuration at the [edit class-of-service] hierarchy level must support eight queues per interface.

2. Specify the number of redundant Ethernet interfaces.

```

[edit chassis cluster]
user@host# set reth-count 2

```

3. Configure the control ports.

```

[edit chassis cluster]
user@host# set control-ports fpc 4 port 0
user@host# set control-ports fpc 10 port 0

```

4. Configure redundancy groups.

```

[edit chassis cluster]
user@host# set redundancy-group 0 node 0 priority 254
user@host# set redundancy-group 0 node 1 priority 1
user@host# set redundancy-group 1 node 0 priority 200
user@host# set redundancy-group 1 node 1 priority 100

```

5. Configure the redundant Ethernet interfaces.

```

[edit interfaces]
user@host# set ge-5/1/14 gigether-options redundant-parent reth0
user@host# set ge-11/1/14 gigether-options redundant-parent reth0
user@host# set ge-5/1/15 gigether-options redundant-parent reth1
user@host# set ge-11/1/15 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth0 vlan-tagging
user@host# set reth0 unit 0 vlan-id 1350

```

```
user@host# set reth0 unit 0 family inet address 192.0.2.1/24
user@host# set reth1 hierarchical-scheduler
user@host# set reth1 vlan-tagging
user@host# set reth1 redundant-ether-options redundancy-group 2
user@host# set reth1 unit 0 vlan-id 1351
user@host# set reth1 unit 0 family inet address 192.0.2.2/24
user@host# set reth1 unit 1 vlan-id 1352
user@host# set reth1 unit 1 family inet address 192.0.2.3/24
user@host# set reth1 unit 2 vlan-id 1353
user@host# set reth1 unit 2 family inet address 192.0.2.4/24
user@host# set reth1 unit 3 vlan-id 1354
user@host# set reth1 unit 3 family inet address 192.0.2.5/24
```

6. Define a classifier and apply it to a logical interface.

```
[edit class-of-service]
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q0
  loss-priority low code-points 000
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q2
  loss-priority low code-points 010
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q3
  loss-priority low code-points 011
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q1
  loss-priority low code-points 001
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q4
  loss-priority low code-points 100
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q5
  loss-priority low code-points 101
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q6
  loss-priority low code-points 110
user@host# set classifiers inet-precedence inet_prec_4 forwarding-class q7
  loss-priority low code-points 111
```

7. Map forwarding classes to CoS queues.

```
[edit class-of-service]
user@host# set forwarding-classes queue 0 q0
user@host# set forwarding-classes queue 1 q1
user@host# set forwarding-classes queue 2 q2
user@host# set forwarding-classes queue 3 q3
user@host# set forwarding-classes queue 4 q4
user@host# set forwarding-classes queue 5 q5
user@host# set forwarding-classes queue 6 q6
user@host# set forwarding-classes queue 7 q7
```

8. Configure traffic control profiles.

```
[edit class-of-service]
user@host# set traffic-control-profiles 1 scheduler-map sched_map
user@host# set traffic-control-profiles 1 shaping-rate 200m
```

9. Define packet flow through the CoS elements.

```
[edit class-of-service]
user@host# set interfaces reth0 unit 0 classifiers inet-precedence inet_prec_4
```

10. Apply a traffic scheduling profile to the interface.

```
[edit class-of-service]
user@host# set interfaces reth1 unit 0 output-traffic-control-profile 1
```

11. Configure the CoS schedulers.

```
[edit class-of-service]
user@host# set scheduler-maps sched_map forwarding-class q0 scheduler S0
user@host# set scheduler-maps sched_map forwarding-class q1 scheduler S1
user@host# set scheduler-maps sched_map forwarding-class q2 scheduler S2
user@host# set scheduler-maps sched_map forwarding-class q3 scheduler S3
user@host# set scheduler-maps sched_map forwarding-class q4 scheduler S4
user@host# set scheduler-maps sched_map forwarding-class q5 scheduler S5
user@host# set scheduler-maps sched_map forwarding-class q6 scheduler S6
user@host# set scheduler-maps sched_map forwarding-class q7 scheduler S7
user@host# set schedulers S0 transmit-rate percent 20
user@host# set schedulers S1 transmit-rate percent 5
user@host# set schedulers S2 transmit-rate percent 5
user@host# set schedulers S3 transmit-rate percent 10
user@host# set schedulers S4 transmit-rate percent 10
user@host# set schedulers S5 transmit-rate percent 10
user@host# set schedulers S6 transmit-rate percent 10
user@host# set schedulers S7 transmit-rate percent 30
```

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

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

```
[edit]
user@host# show class-of-service
classifiers {
  inet-precedence inet_prec_4 {
    forwarding-class q0 {
      loss-priority low code-points 000;
    }
    forwarding-class q2 {
      loss-priority low code-points 010;
    }
    forwarding-class q3 {
      loss-priority low code-points 011;
    }
    forwarding-class q1 {
      loss-priority low code-points 001;
    }
    forwarding-class q4 {
      loss-priority low code-points 100;
    }
    forwarding-class q5 {
      loss-priority low code-points 101;
    }
  }
}
```

```
    }
    forwarding-class q6 {
        loss-priority low code-points 110;
    }
    forwarding-class q7 {
        loss-priority low code-points 111;
    }
}
forwarding-classes {
    queue 0 q0;
    queue 1 q1;
    queue 2 q2;
    queue 3 q3;
    queue 4 q4;
    queue 5 q5;
    queue 6 q6;
    queue 7 q7;
}
traffic-control-profiles {
    1 {
        scheduler-map sched_map;
        shaping-rate 200m;
    }
}
interfaces {
    reth0 {
        unit 0 {
            classifiers {
                inet-precedence inet_prec_4;
            }
        }
    }
    reth1 {
        unit 0 {
            output-traffic-control-profile 1;
        }
    }
}
scheduler-maps {
    sched_map {
        forwarding-class q0 scheduler S0;
        forwarding-class q1 scheduler S1;
        forwarding-class q2 scheduler S2;
        forwarding-class q3 scheduler S3;
        forwarding-class q4 scheduler S4;
        forwarding-class q5 scheduler S5;
        forwarding-class q6 scheduler S6;
        forwarding-class q7 scheduler S7;
    }
}
schedulers {
    S0 {
        transmit-rate percent 20;
    }
    S1 {
```

```

        transmit-rate percent 5;
    }
    S2 {
        transmit-rate percent 5;
    }
    S3 {
        transmit-rate percent 10;
    }
    S4 {
        transmit-rate percent 10;
    }
    S5 {
        transmit-rate percent 10;
    }
    S6 {
        transmit-rate percent 10;
    }
    S7 {
        transmit-rate percent 30;
    }
}

```

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

To restart chassis control, enter **restart chassis-control** command from operational mode.



**NOTE:** When you execute the **restart chassis-control** command all the FRU cards on the box are reset, thus impacting traffic. Changing the number of queues must be executed during a scheduled downtime. It takes 5-10 minutes for the cards to come online after the **restart chassis-control** command is executed.

## Verification

### Verifying the Eight-Queue COS Configuration

**Purpose** Verify that eight-queue CoS is enabled properly.

**Action** From the operational mode, enter the following commands:

- **show interfaces ge-5/1/14 extensive**
- **show interfaces queue ge-5/1/14**
- **show class-of-service forwarding-class**
- **show class-of-service interface ge-5/1/14**

**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 81](#)

- [Chassis Cluster Redundant Ethernet Interfaces on page 140](#)

## Conditional Route Advertisement over Redundant Ethernet Interfaces on SRX Series Devices in a Chassis Cluster

---

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

Conditional route advertising allows you to add criteria on route advertisements before they are installed in the route table or advertised to peers and neighbors. The goal of conditional route advertisement in a chassis cluster is to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface. For more information, see the following topics:

- [Understanding Conditional Route Advertising in a Chassis Cluster on page 176](#)
- [Example: Configuring Conditional Route Advertising in a Chassis Cluster on page 178](#)

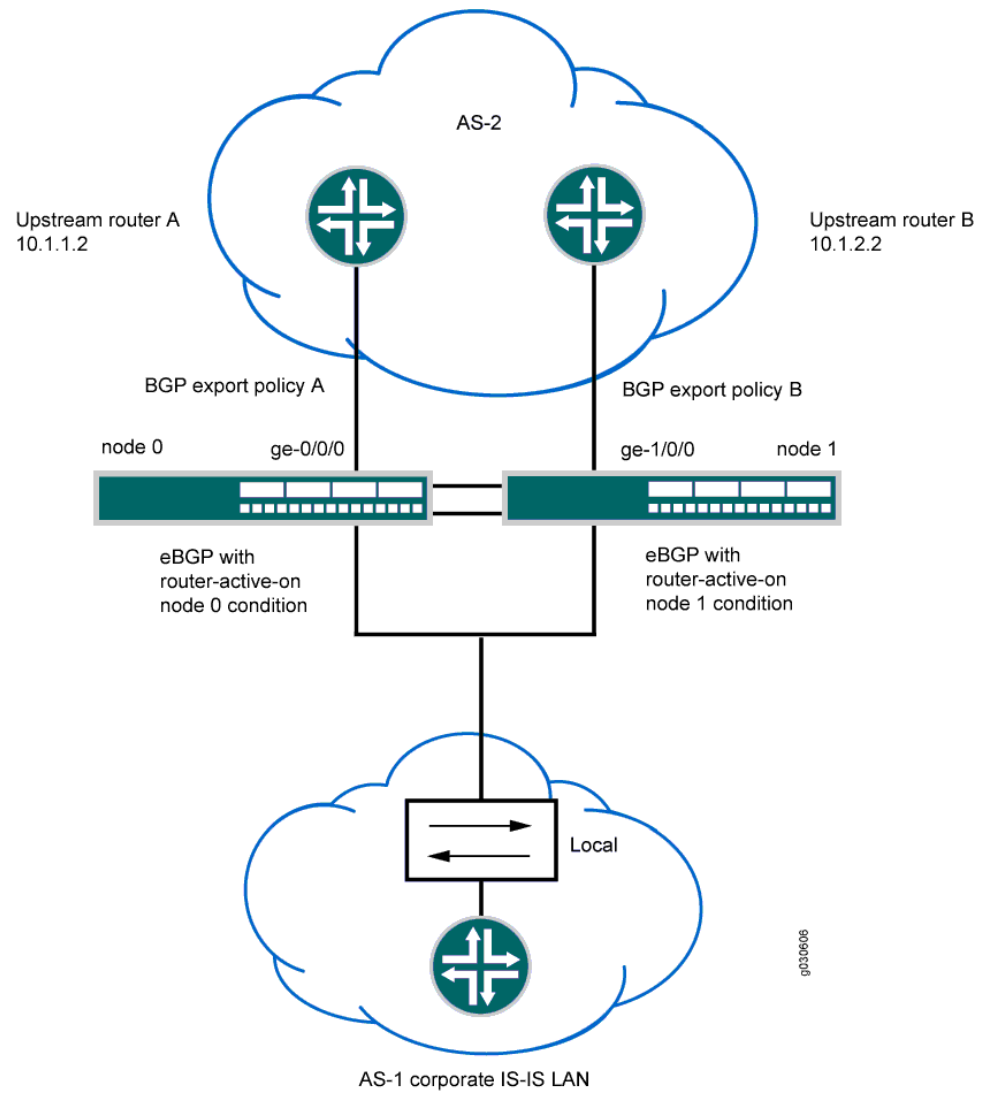
### Understanding Conditional Route Advertising in a Chassis Cluster

Route advertisement over redundant Ethernet interfaces in a chassis cluster is complicated by the fact that the active node in the cluster can change dynamically. Conditional route advertisement enables you to advertise routes in such a way that incoming traffic from the core network is attracted to the Border Gateway Protocol (BGP) interface that exists on the same node as the currently active redundant Ethernet interface. In this way, traffic is processed by the active node and does not traverse the fabric interface between nodes. You do this by manipulating the BGP attribute at the time routes are advertised by BGP.

The goal of conditional route advertisement in a chassis cluster is to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface. To understand how this works, keep in mind that in a chassis cluster, each node has its own set of interfaces. [Figure 46 on page 177](#) shows a typical scenario, with a redundant Ethernet interface connecting the corporate LAN, through a chassis cluster, to an external network segment.



Figure 46: Conditional Route Advertising



- See Also**
- [Example: Configuring Conditional Route Advertising in a Chassis Cluster on page 178](#)
  - [Viewing a Chassis Cluster Configuration on page 164](#)
  - [Viewing Chassis Cluster Statistics on page 164](#)

## Example: Configuring Conditional Route Advertising in a Chassis Cluster

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 178](#)
- [Overview on page 178](#)
- [Configuration on page 180](#)

---

### Requirements

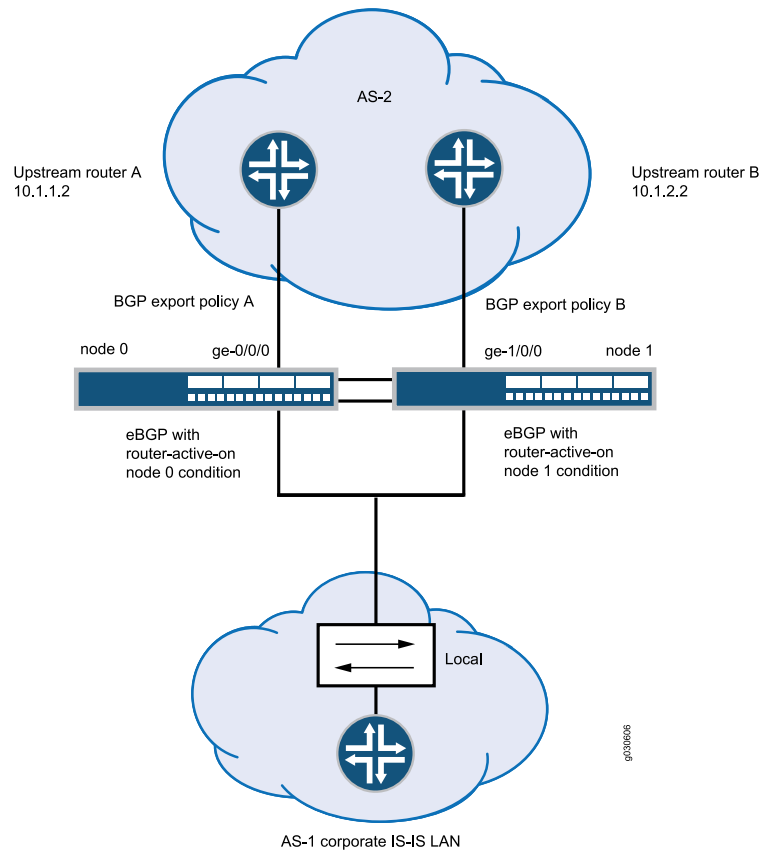
Before you begin, understand conditional route advertising in a chassis cluster. See [“Understanding Conditional Route Advertising in a Chassis Cluster” on page 176](#).

---

### Overview

As illustrated in [Figure 47 on page 179](#), 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 47: Conditional Route Advertising on SRX Series Devices in a Chassis Cluster



To achieve this behavior, you apply a policy to BGP before exporting routes. An additional term in the policy match condition determines the current active redundant Ethernet interface child interface of the next hop before making the routing decision. When the active status of a child redundant Ethernet interface changes, BGP reevaluates the export policy for all routes affected.

The condition statement in this configuration works as follows. The command states that any routes evaluated against this condition will pass only if:

- The routes have a redundant Ethernet interface as their next-hop interface.

- The current child interface of the redundant Ethernet interface is active at node 0 (as specified by the **route-active-on node0** keyword).

```
{primary:node0}[edit]
user@host# set policy-options condition reth-nh-active-on-0 route-active-on node0
```

Note that a route might have multiple equal-cost next hops, and those next hops might be redundant Ethernet interfaces, regular interfaces, or a combination of both. The route still satisfies the requirement that it has a redundant Ethernet interface as its next hop.

If you use the BGP export policy set for node 0 in the previous example command, only OSPF routes that satisfy the following requirements will be advertised through the session:

- The OSPF routes have a redundant Ethernet interface as their next hop.
- The current child interface of the redundant Ethernet interface is currently active at node 0.

You must also create and apply a separate policy statement for the other BGP session by using this same process.

In addition to the BGP MED attribute, you can define additional BGP attributes, such as origin-code, as-path, and community.

---

### Configuration

#### 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.

- See Also**
- [Understanding Conditional Route Advertising in a Chassis Cluster on page 176](#)
  - [Viewing a Chassis Cluster Configuration on page 164](#)
  - [Viewing Chassis Cluster Statistics on page 164](#)



## CHAPTER 3

# Configuring Redundancy and Failover in a Chassis Cluster

- [Chassis Cluster Dual Control Links on page 183](#)
- [Chassis Cluster Dual Fabric Links on page 190](#)
- [Monitoring of Global-Level Objects in a Chassis Cluster on page 196](#)
- [Monitoring Chassis Cluster Interfaces on page 200](#)
- [Monitoring IP Addresses on a Chassis Cluster on page 229](#)
- [Configuring Cluster Failover Parameters on page 245](#)
- [Chassis Cluster Redundancy Group Failover on page 249](#)

## Chassis Cluster Dual Control Links

---

**Supported Platforms** [SRX4600, SRX5600, SRX5800](#)

Dual control links provide a redundant link for control traffic. For more information, see the following topics:

- [Understanding Chassis Cluster Dual Control Links on page 183](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 185](#)
- [Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices on page 186](#)
- [Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 188](#)

## Understanding Chassis Cluster Dual Control Links

The control link connects two SRX Series devices together and it is responsible for sending high-availability control data between the two SRX Series devices including heartbeats and configuration synchronization. If this link goes down, the secondary SRX Series is disabled from the cluster. In dual control links, two pairs of control link interfaces are connected between each device in a cluster. Having two control links helps to avoid a possible single point of failure. Dual control links, provide a redundant link for control traffic. Unlike dual fabric links, only one control link is used at any one time.

Dual control links are supported for the SRX4600, SRX5600 and SRX5800 Services Gateways.



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



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

---

### Benefits of Dual Control Links

- Provides a redundant link for control traffic. In the link-level redundancy, if one link fails, the other can take over and restore traffic forwarding that had been previously sent over the failed link.
- Prevents the possibility of single point of failure.

---

### Dual Control Links Functionality Requirements

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



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

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

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



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





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



**NOTE:** For the SRX3000 line, dual control link functionality requires an SRX Clustering Module (SCM) to be installed on each device in the cluster. Although the SCM fits in the Routing Engine slot, it is not a Routing Engine. SRX3000 line devices do not support a second Routing Engine. The purpose of the SCM is to initialize the second control link.

**See Also** • [Chassis Cluster Control Plane Interfaces on page 129](#)

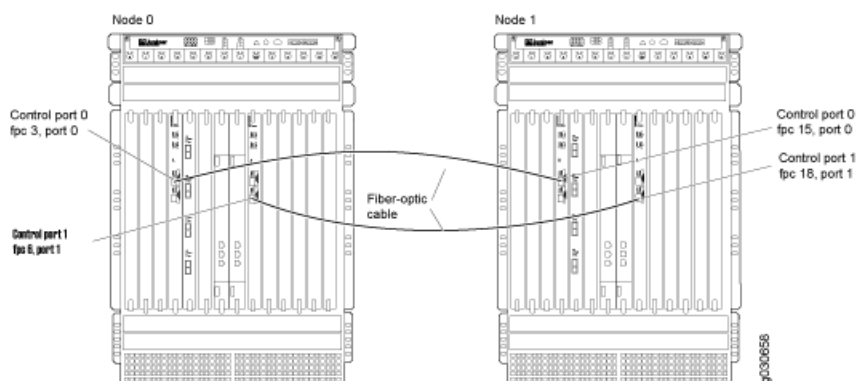
## Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster

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

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

For SRX5600 and SRX5800 devices, connect two pairs of the same type of Ethernet ports. For each device, you can use ports on the same Services Processing Card (SPC), but we recommend that they be on two different SPCs to provide high availability. [Figure 48 on page 185](#) shows a pair of SRX5800 devices with dual control links connected. In this example, control port 0 and control port 1 are connected on different SPCs.

**Figure 48: Connecting Dual Control Links (SRX5800 Devices)**





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

- See Also**
- [Connecting SRX Series Devices to Create a Chassis Cluster on page 102](#)
  - [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)

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

For SRX5600 and SRX5800 devices, a second Routing Engine is required for each device in a cluster if you are using dual control links. The second Routing Engine does not provide backup functionality; its purpose is only to initialize the switch on the Switch Control Board (SCB). The second Routing Engine must be running Junos OS Release 12.1X47-D35, 12.3X48-D30, 15.1X49-D40 or later. For more information, see knowledge base article [KB30371](#).



**NOTE:** On SRX5600 and SRX5800 devices, starting from Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, you can use the `show chassis hardware` command to see the serial number and the hardware version details of the second Routing Engine. To use this functionality, ensure that the second Routing Engine is running Junos OS Release 15.1X49-D70 and later releases or Junos OS Release 17.3R1 or later releases.



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

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

To upgrade the software image on the second Routing Engine:

1. Use FTP to copy the installation media into the `/var/tmp` directory of the master Routing Engine.
2. Insert a USB storage device into the USB port on the master Routing Engine.
3. In the UNIX shell, navigate to the `/var/tmp` directory:

```
start shell
```

```
cd /var/tmp
```

4. Log in as root or superuser:

```
su [enter]
password: [enter SU password]
```

5. Issue the following command:

```
dd if=installMedia of=/dev/externalDrive bs=1m
```

where

- *externalDrive*—Refers to the removable media name. For example, the removable media name on an SRX5000 line device is da0 for both Routing Engines.
- *installMedia*—Refers to the installation media downloaded into the /var/tmp directory. For example, install-media-srx5000-10.1R1-domestic.tgz.

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

```
dd if=install-media-srx5000-10.1R1-domestic.tgz of=/dev/da0 bs=1m
```

6. Log out as root or superuser:

```
exit
```

7. After the software image is written to the USB storage device, remove the device and insert it into the USB port on the second Routing Engine.
8. Move the console connection from the master Routing Engine to the second Routing Engine, if you do not already have a connection.
9. Reboot the second Routing Engine. Issue the following command (for Junos OS Release 15.1X49-D65 and earlier):

```
# reboot
```

Starting with Junos OS Release 15.1X49-D70, issue the following command:

```
login : root
root % reboot
```

- When the following system output appears, press y:

```
WARNING: The installation will erase the contents of your disks.
Do you wish to continue (y/n)?
```

- When the following system output appears, remove the USB storage device and press Enter:

```
Eject the installation media and hit [Enter] to reboot?
```

## Example: Configuring Chassis Cluster Control Ports for Dual Control Links

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



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

- [Requirements on page 188](#)
- [Overview on page 188](#)
- [Configuration on page 188](#)
- [Verification on page 189](#)

---

### Requirements

Before you begin:

- Understand chassis cluster control links. See [“Understanding Chassis Cluster Control Plane and Control Links” on page 129](#).
- Physically connect the control ports on the devices. See [“Connecting SRX Series Devices to Create a Chassis Cluster” on page 102](#).

---

### Overview

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

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

- FPC 4, port 0
- FPC 10, port 0
- FPC 6, port 1
- FPC 12, port 1

---

### Configuration

#### CLI Quick Configuration

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

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

```
set chassis cluster control-ports fpc 6 port 1
set chassis cluster control-ports fpc 12 port 1
```

**Step-by-Step Procedure** To configure control ports for use as dual control links for the chassis cluster:

- Specify the control ports.
 

```
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 4 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 10 port 0
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 6 port 1
{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 12 port 1
```

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

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

```
{primary:node0}[edit]
user@host# show chassis cluster
...
control-ports {
  fpc 4 port 0;
  fpc 6 port 1;
  fpc 10 port 0;
  fpc 12 port 1;
}
...
```

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

## Verification

### Verifying the Chassis Cluster Status

**Purpose** Verify the chassis cluster status.

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

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

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

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

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

**Related Documentation**

- [Connecting SRX Series Devices to Create a Chassis Cluster on page 102](#)
- [SRX Series Chassis Cluster Configuration Overview on page 81](#)

---

## Chassis Cluster Dual Fabric Links

Dual fabric links remove single point of failure in a chassis cluster setup. If one fabric link fails and one remains functional, all sessions are maintained between the two nodes and the chassis cluster status is preserved. For more information, see the following topics:

- [Understanding Chassis Cluster Dual Fabric Links on page 190](#)
- [Example: Configuring the Chassis Cluster Dual Fabric Links with Matching Slots and Ports on page 191](#)
- [Example: Configuring Chassis Cluster Dual Fabric Links with Different Slots and Ports on page 193](#)

### Understanding Chassis Cluster Dual Fabric Links

You can connect two fabric links between each device in a cluster, which provides a redundant fabric link between the members of a cluster. Having two fabric links helps to avoid a possible single point of failure.

When you use dual fabric links, the RTOs and probes are sent on one link and the fabric-forwarded and flow-forwarded packets are sent on the other link. If one fabric link fails, the other fabric link handles the RTOs and probes, as well as the data forwarding. The system selects the physical interface with the lowest slot, PIC, or port number on each node for the RTOs and probes.

For all SRX Series 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.

**See Also** • [Understanding Chassis Cluster Fabric Interfaces on page 119](#)

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

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 191](#)
- [Overview on page 191](#)
- [Configuration on page 192](#)
- [Verification on page 193](#)

### Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See [“Example: Setting the Chassis Cluster Node ID and Cluster ID” on page 110](#).

### 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

```

**See Also**

- [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)

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

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 193](#)
- [Overview on page 194](#)
- [Configuration on page 194](#)
- [Verification on page 195](#)

#### Requirements

Before you begin, set the chassis cluster ID and chassis cluster node ID. See “[Example: Setting the Chassis Cluster Node ID and Cluster ID](#)” on page 110.

## 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 maximum transmission unit (MTU) size supported is 9014. 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**
- [Connecting SRX Series Devices to Create a Chassis Cluster on page 102](#)
  - [SRX Series Chassis Cluster Configuration Overview on page 81](#)

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

**Supported Platforms** [SRX4100, SRX4200, SRX5400, SRX5600, SRX5800, vSRX](#)

There are various types of objects to monitor as you work with devices configured as chassis clusters, including global-level objects and objects that are specific to redundancy groups. This section describes the monitoring of global-level objects.

The SRX5000 lines have one or more Services Processing Units (SPUs) that run on a Services Processing Card (SPC). All flow-based services run on the SPU. Other SRX Series devices have a flow-based forwarding process, *flowd*, which forwards packets through the device.

- [Understanding SPU Monitoring on page 196](#)
- [Understanding flowd Monitoring on page 197](#)
- [Understanding Cold-Sync Monitoring on page 198](#)

### Understanding SPU Monitoring

SPU monitoring tracks the health of the SPUs and of the central point (CP). The chassis manager on each SPC monitors the SPUs and the central point, and also maintains the heartbeat with the Routing Engine chassisd. In this hierarchical monitoring system, chassisd is the center for hardware failure detection. SPU monitoring is enabled by default.



**NOTE:** SPU monitoring is supported on SRX4600 and SRX5000 line devices.

Persistent SPU and central point failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups *x* to 0.

- A central point failure triggers failover to the secondary node. The failed node's PFE, which includes all SPCs and all I/O cards (IOCs), is automatically restarted. If the secondary central point has failed as well, the cluster is unable to come up because there is no primary device. Only the data plane (redundancy group *x*) is failed over.

- A single, failed SPU causes failover of redundancy group *x* to the secondary node. All IOCs and SPCs on the failed node are restarted and redundancy group *x* is failed over to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group *x*. The interval for dead SPU detection is 30 seconds.



**NOTE:** On SRX5400, SRX5600, and SRX5800 SPCs, the Routing Engine monitors the health of the chassis manager. The chassis manager sends a heartbeat message to the Routing Engine chassisd every second. When the Routing Engine chassisd detects a heartbeat loss, it initiates a power cycle for the entire SPC. If multiple recoveries fail within a certain timeframe, the Routing Engine powers off the SPC to prevent it from affecting the entire system.

This event triggers an alarm, indicating that a new field-replaceable unit (FRU) is needed.

The following list describes the limitations for inserting an SPC on SRX5400, SRX5600, and SRX5800 devices in chassis cluster mode:

- The chassis cluster must be in active/passive mode before and during the SPC insert procedure.
- A different number of SPCs cannot be inserted in two different nodes.
- A new SPC must be inserted in a slot that is higher than the central point slot.

The existing combo central point cannot be changed to a full central point after the new SPC is inserted.

- During an SPC insert procedure, the IKE and IPsec configurations cannot be modified.



**NOTE:** An SPC is not hot-insertable. Before inserting an SPC, the device must be taken offline. After inserting an SPC, the device must be rebooted.

- Users cannot specify the SPU and the IKE instance to anchor a tunnel.
- After a new SPC is inserted, the existing tunnels cannot use the processing power of the new SPC and redistribute it to the new SPC.

## Understanding flowd Monitoring

Flowd monitoring tracks the health of the flowd process. Flowd monitoring is enabled by default.

Persistent flowd failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups *x* to 0.

A failed flowd process causes failover of redundancy group  $x$  to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group  $x$ .

During SPC and flowd monitoring failures on a local node, the data plane redundancy group RG1+ fails over to the other node that is in a good state. However, the control plane RG0 does not fail over and remains primary on the same node as it was before the failure.

## Understanding Cold-Sync Monitoring

The process of synchronizing the data plane runtime objects (RTOs) on the startup of the SPUs or flowd is called *cold sync*. When all the RTOs are synchronized, the cold-sync process is complete, and the SPU or flowd on the node is ready to take over for the primary node, if needed. The process of monitoring the cold-sync state of all the SPUs or flowd on a node is called *cold-sync monitoring*. Keep in mind that when preempt is enabled, cold-sync monitoring prevents the node from taking over the mastership until the cold-sync process is completed for the SPUs or flowd on the node. Cold-sync monitoring is enabled by default.

When the node is rebooted, or when the SPUs or flowd come back up from failure, the priority for all the redundancy groups  $1+$  is 0. When an SPU or flowd comes up, it tries to start the cold-sync process with its mirror SPU or flowd on the other node.

If this is the only node in the cluster, the priorities for all the redundancy groups  $1+$  stay at 0 until a new node joins the cluster. Although the priority is at 0, the device can still receive and send traffic over its interfaces. A priority of 0 implies that it cannot fail over in case of a failure. When a new node joins the cluster, all the SPUs or flowd, as they come up, will start the cold-sync process with the mirror SPUs or flowd of the existing node.

When the SPU or flowd of a node that is already up detects the cold-sync request from the SPU or flowd of the peer node, it posts a message to the system indicating that the cold-sync process is complete. The SPUs or flowd of the newly joined node posts a similar message. However, they post this message only after all the RTOs are learned and cold-sync is complete. On receipt of completion messages from all the SPUs or flowd, the priority for redundancy groups  $1+$  moves to the configured priority on each node if there are no other failures of monitored components, such as interfaces. This action ensures that the existing primary node for redundancy  $1+$  groups always moves to the configured priority first. The node joining the cluster later moves to its configured priorities only after all its SPUs or flowd have completed their cold-sync process. This action in turn guarantees that the newly added node is ready with all the RTOs before it takes over mastership.

### Understanding Cold-Sync Monitoring with SPU Replacement or Expansion

---

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



**NOTE:** For SRX5400 devices, only SPC2 is supported.

The following events take place during this scenario:

- When the SPC2 is to be installed on a node (for example, on node 1, the secondary node), node 1 is shut down so the SPC2 can be installed.
- Once node 1 is powered up and rejoins the cluster, the number of SPUs on node 1 will be higher than the number of SPUs on node 0, the primary node. Now, one node (node 0) still has an old SPC while the other node has the new SPC2; SPC2s have four SPUs per card, and the older SPCs have two SPUs per card.

The cold-sync process is based on node 0 total SPU number. Once those SPUs in node 1 corresponding to node 0 SPUs have completed the cold-sync, the node 1 will declare cold-sync completed. Since the additional SPUs in node 1 do not have the corresponding node 0 SPUs, there is nothing to be synchronized and failover from node 0 to node 1 does not cause any issue.

SPU monitoring functionality monitors all SPUs and reports if there are any SPU failure.

For example assume that both nodes originally have 2 existing SPCs and you have replaced both SPCs with SPC2 on node 1. Now we have 4 SPUs in node 0 and 8 SPUs in node 1. The SPU monitoring function monitors the 4 SPUs on node 0 and 8 SPUs on node 1. If any of those 8 SPUs failed in node 1, the SPU monitoring will still report to the Juniper Services Redundancy Protocol (jsrpd) process that there is an SPU failure. The jsrpd process controls chassis clustering.

- Once node 1 is ready to failover, you can initiate all redundancy group failover manually to node 1. Node 0 will be shut down to replace its SPC with the SPC2. After the replacement, node 0 and node 1 will have exactly the same hardware setup.

Once node 0 is powered up and rejoins the cluster, the system will operate as a normal chassis cluster.



**NOTE:** Starting from Junos OS Release 15.1X49-D120, when the cold-sync process is still in progress on SRX Series device in chassis cluster, and if the control link is down, a delay (of 30 seconds) is expected before the node takes transition from the secondary state to the primary state.

#### Related Documentation

- [Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 200](#)
- [Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring on page 201](#)
- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 232](#)
- [Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 235](#)

## Monitoring Chassis Cluster Interfaces

---

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

Interface monitoring monitors the state of an interface by checking if the interface is in an up or down state. When one or more monitored interfaces fail, the redundancy group fails over to the other node in the cluster. For more information, see the following topics:

- [Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 200](#)
- [Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring on page 201](#)

## Understanding Chassis Cluster Redundancy Group Interface Monitoring

For a redundancy group to automatically failover to another node, its interfaces must be monitored. When you configure a redundancy group, you can specify a set of interfaces that the redundancy group is to monitor for status (or “health”) to determine whether the interface is up or down. A monitored interface can be a child interface of any of its redundant Ethernet interfaces. When you configure an interface for a redundancy group to monitor, you give it a weight.

Every redundancy group has a threshold tolerance value initially set to 255. When an interface monitored by a redundancy group becomes unavailable, its weight is subtracted from the redundancy group's threshold. When a redundancy group's threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

To check the interface weight, use the following commands:

- `show chassis cluster information`
- `show chassis cluster interfaces`



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

---



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

---



A redundancy group failover occurs because the cumulative weight of the redundancy group's monitored interfaces has brought its threshold value to 0. When the monitored interfaces of a redundancy group on both nodes reach their thresholds at the same time, the redundancy group is primary on the node with the lower node ID, in this case node 0.

**NOTE:**

- If you want to dampen the failovers occurring because of interface monitoring failures, use the `hold-down-interval` statement.
- If a failover occurs on redundancy group 0 (RGO), the interface monitoring on the RGO secondary is disabled for 30 seconds. This prevents failover of other redundancy groups along with RGO failover.

---

### Benefits of Monitoring Chassis Cluster Redundancy Group Interfaces

- Helps to determine the status of a specific interface in a chassis cluster setup by a specific redundancy group.
- Enables automatic failover of an interface to another node if the interface is down.

**See Also**   • [Understanding Chassis Cluster Redundancy Groups on page 134](#)

### Example: Configuring Chassis Cluster Redundancy Group Interface Monitoring

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 201](#)
- [Overview on page 201](#)
- [Configuration on page 203](#)
- [Verification on page 207](#)

---

#### Requirements

Before you begin, create a redundancy group. See “[Example: Configuring Chassis Cluster Redundancy Groups](#)” on page 138.

---

#### Overview

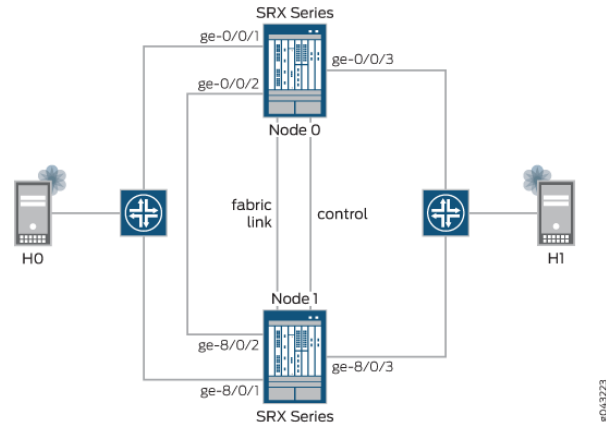
To retrieve the remaining redundancy group threshold after a monitoring interface is down, you can configure your system to monitor the health of the interfaces belonging to a redundancy group. When you assign a weight to an interface to be monitored, the system monitors the interface for availability. 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 49 on page 203 illustrates the network topology used in this example.

Figure 49: 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 208](#)
- [Verifying Chassis Cluster Interfaces on page 208](#)
- [Verifying Chassis Cluster Information on page 209](#)
- [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 211](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 on page 211](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 on page 212](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130 on page 213](#)
- [Verifying Interface ge-0/0/2 Is Disabled on page 214](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/2 on page 215](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/2 on page 215](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/2 on page 216](#)
- [Verifying Interface Status After Disabling ge-0/0/3 on page 218](#)
- [Verifying Chassis Cluster Status After Disabling Interface ge-0/0/3 on page 218](#)
- [Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/3 on page 219](#)
- [Verifying Chassis Cluster Information After Disabling Interface ge-0/0/3 on page 220](#)
- [Verifying That Interface ge-0/0/2 Is Enabled on page 221](#)
- [Verifying Chassis Cluster Status After Enabling Interface ge-0/0/2 on page 222](#)
- [Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/2 on page 222](#)
- [Verifying Chassis Cluster Information After Enabling Interface ge-0/0/2 on page 223](#)
- [Verifying Chassis Cluster RG2 Preempt on page 225](#)
- [Verifying Chassis Cluster Status After Preempting RG2 on page 225](#)

- [Verifying That Interface ge-0/0/3 Is Enabled on page 226](#)
- [Verifying Chassis Cluster Status After Enabling Interface ge-0/0/3 on page 226](#)
- [Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/3 on page 227](#)
- [Verifying Chassis Cluster Information After Enabling Interface ge-0/0/3 on page 228](#)

### ***Verifying Chassis Cluster Status***

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

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

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

Monitor Failure codes:

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

```
Cluster ID: 2
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  254      primary      no      no      None
node1  1        secondary   no      no      None

Redundancy group: 1 , Failover count: 1
node0  200      primary      no      no      None
node1  100      secondary   no      no      None

Redundancy group: 2 , Failover count: 1
node0  200      primary      no      no      None
node1  100      secondary   no      no      None
```

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

### ***Verifying Chassis Cluster Interfaces***

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

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

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



Control link status: Up

Control interfaces:

Index	Interface	Monitored-Status	Internal-SA
0	em0	Up	Disabled
1	em1	Down	Disabled

Fabric link status: Up

Fabric interfaces:

Name	Child-interface	Status (Physical/Monitored)
fab0	ge-0/0/0	Up / Up
fab0		
fab1	ge-8/0/0	Up / Up
fab1		

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1
reth2	Up	2

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-8/0/2	120	Up	1
ge-8/0/1	150	Up	1
ge-0/0/2	140	Up	1
ge-0/0/1	130	Up	1
ge-8/0/3	255	Up	2
ge-0/0/3	255	Up	2

**Meaning** The sample output confirms that monitoring interfaces are up and that the weight of each interface being monitored is displayed correctly as configured. These values do not change if the interface goes up or down. The weights only change for the redundant group and can be viewed when you use the **show chassis cluster information** command.

### *Verifying Chassis Cluster Information*

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

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

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

```
node0:
```

```
-----
Redundancy Group Information:
```

Redundancy Group 0 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:27	hold	secondary	Hold timer expired
Feb 24 22:56:34	secondary	primary	Better priority (254/1)

Redundancy Group 1 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:12	secondary	primary	Remote yield (0/0)

Redundancy Group 2 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:13	secondary	primary	Remote yield (0/0)

Chassis cluster LED information:

Current LED color: Green  
Last LED change reason: No failures

node1:

-----  
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:34	hold	secondary	Hold timer expired

Redundancy Group 1 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired

Redundancy Group 2 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired

Chassis cluster LED information:

Current LED color: Green  
Last LED change reason: No failures

**Meaning** The sample output confirms that node 0 and node 1 are healthy, and the green LED on the device indicates that there are no failures. Also, the default weight of the redundancy group (255) is displayed. The default weight is deducted whenever an interface mapped to the corresponding redundancy group goes down.

Refer to subsequent verification sections to see how the redundancy group value varies when a monitoring interface goes down or comes up.

### *Verifying Interface ge-0/0/1 Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130*

**Purpose** Verify that the interface ge-0/0/1 is disabled on node 0.

**Action** From configuration mode, enter the **set interface ge-0/0/1 disable** command.

```
{primary:node0}
user@host# set interface ge-0/0/1 disable
user@host# commit

node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

{primary:node0}
user@host# show interfaces ge-0/0/1
disable;
gigether-options {
    redundant-parent reth0;
}
```

**Meaning** The sample output confirms that interface ge-0/0/1 is disabled.

### *Verifying Chassis Cluster Status After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130*

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

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

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring            SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary no no None
node1 1 secondary no no None

Redundancy group: 1 , Failover count: 1
```

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

Redundancy group: 2 , Failover count: 1

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

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

### *Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130*

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

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

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

Control interfaces:

Index	Interface	Monitored-Status	Internal-SA
0	em0	Up	Disabled
1	em1	Down	Disabled

Fabric link status: Up

Fabric interfaces:

Name	Child-interface	Status (Physical/Monitored)
fab0	ge-0/0/0	Up / Up
fab0		
fab1	ge-8/0/0	Up / Up
fab1		

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Down	1
reth1	Up	1
reth2	Up	2

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-8/0/2	120	Up	1
ge-8/0/1	150	Up	1
ge-0/0/2	140	Up	1
ge-0/0/1	130	Down	1

ge-8/0/3	255	Up	2
ge-0/0/3	255	Up	2

**Meaning** The sample output confirms that monitoring interface ge-0/0/1 is down.

### *Verifying Chassis Cluster Information After Disabling Interface ge-0/0/1 of RG1 in Node 0 with a Weight of 130*

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

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

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

node0:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: primary, Weight: 255

    Time           From           To           Reason
    Feb 24 22:56:27 hold           secondary    Hold timer expired
    Feb 24 22:56:34 secondary    primary      Better priority (254/1)

Redundancy Group 1 , Current State: primary, Weight: 125

    Time           From           To           Reason
    Feb 24 23:16:12 hold           secondary    Hold timer expired
    Feb 24 23:16:12 secondary    primary      Remote yield (0/0)

Redundancy Group 2 , Current State: primary, Weight: 255

    Time           From           To           Reason
    Feb 24 23:16:12 hold           secondary    Hold timer expired
    Feb 24 23:16:13 secondary    primary      Remote yield (0/0)

Chassis cluster LED information:
Current LED color: Green
Last LED change reason: No failures

Failure Information:

Interface Monitoring Failure Information:
Redundancy Group 1, Monitoring status: Unhealthy
    Interface           Status
    ge-0/0/1             Down

node1:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255
```

Time	From	To	Reason
Feb 24 22:56:34	hold	secondary	Hold timer expired

Redundancy Group 1 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired

Redundancy Group 2 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

**Meaning** The sample output confirms that in node 0, the RG1 weight is reduced to 125 (that is, 255 minus 130) because monitoring interface ge-0/0/1 (weight of 130) went down. The monitoring status is unhealthy, the device LED is amber, and the interface status of ge-0/0/1 is down.



**NOTE:** If interface ge-0/0/1 is brought back up, the weight of RG1 in node 0 becomes 255. Conversely, if interface ge-0/0/2 is also disabled, the weight of RG1 in node 0 becomes 0 or less (in this example, 125 minus 140 = -15) and triggers failover, as indicated in the next verification section.

### Verifying Interface ge-0/0/2 Is Disabled

**Purpose** Verify that interface ge-0/0/2 is disabled on node 0.

**Action** From configuration mode, enter the **set interface ge-0/0/2 disable** command.

```
{primary:node0}
user@host# set interface ge-0/0/2 disable
user@host# commit
```

```
node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete
```

```
{primary:node0}
user@host# show interfaces ge-0/0/2
disable;
gigether-options {
    redundant-parent reth1;
}
```

**Meaning** The sample output confirms that interface ge-0/0/2 is disabled.

### *Verifying Chassis Cluster Status After Disabling Interface ge-0/0/2*

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

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

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary      no    no    None
node1 1 secondary    no    no    None

Redundancy group: 1 , Failover count: 2
node0 0 secondary    no    no    IF
node1 100 primary     no    no    None

Redundancy group: 2 , Failover count: 1
node0 200 primary     no    no    None
node1 100 secondary   no    no    None
```

**Meaning** Use the **show chassis cluster status** command to confirm that devices in the chassis cluster are communicating properly, with one device functioning as the primary node and the other as the secondary node. On RG1, you see interface failure, because both interfaces mapped to RG1 on node 0 failed during interface monitoring.

### *Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/2*

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

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

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

Control interfaces:
  Index  Interface  Monitored-Status  Internal-SA
```

0	em0	Up	Disabled
1	em1	Down	Disabled

Fabric link status: Up

Fabric interfaces:

Name	Child-interface	Status (Physical/Monitored)
fab0	ge-0/0/0	Up / Up
fab0		
fab1	ge-8/0/0	Up / Up
fab1		

Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1
reth2	Up	2

Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	0

Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-8/0/2	120	Up	1
ge-8/0/1	150	Up	1
ge-0/0/2	140	Down	1
ge-0/0/1	130	Down	1
ge-8/0/3	255	Up	2
ge-0/0/3	255	Up	2

**Meaning** The sample output confirms that monitoring interfaces ge-0/0/1 and ge-0/0/2 are down.

#### *Verifying Chassis Cluster Information After Disabling Interface ge-0/0/2*

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

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

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

node0:

-----  
Redundancy Group Information:

Redundancy Group 0 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:27	hold	secondary	Hold timer expired
Feb 24 22:56:34	secondary	primary	Better priority (254/1)

Redundancy Group 1 , Current State: secondary, Weight: -15



Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:12	secondary	primary	Remote yield (0/0)
Feb 24 23:31:36	primary	secondary-hold	Monitor failed: IF
Feb 24 23:31:37	secondary-hold	secondary	Ready to become secondary

Redundancy Group 2 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:12	hold	secondary	Hold timer expired
Feb 24 23:16:13	secondary	primary	Remote yield (0/0)

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

Failure Information:

Interface Monitoring Failure Information:

Redundancy Group 1, Monitoring status: Failed

Interface	Status
ge-0/0/2	Down
ge-0/0/1	Down

node1:

-----  
Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 22:56:34	hold	secondary	Hold timer expired

Redundancy Group 1 , Current State: primary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired
Feb 24 23:31:36	secondary	primary	Remote is in secondary hold

Redundancy Group 2 , Current State: secondary, Weight: 255

Time	From	To	Reason
Feb 24 23:16:10	hold	secondary	Hold timer expired

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

**Meaning** The sample output confirms that in node 0, monitoring interfaces ge-0/0/1 and ge-0/0/2 are down. The weight of RG1 on node 0 reached zero value, which triggered RG1 failover during use of the **show chassis cluster status** command.



**NOTE:** For RG2, the default weight of 255 is set for redundant Ethernet interface 2 (reth2). When interface monitoring is required, we recommend that you use the default weight when you do not have backup links like those in RG1. That is, if interface ge-0/0/3 is disabled, it immediately triggers failover because the weight becomes 0 (255 minus 225), as indicated in the next verification section.

### *Verifying Interface Status After Disabling ge-0/0/3*

**Purpose** Verify that interface ge-0/0/3 is disabled on node 0.

**Action** From configuration mode, enter the **set interface ge-0/0/3 disable** command.

```
{primary:node0}
user@host# set interface ge-0/0/3 disable
user@host# commit
```

```
node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete
```

```
{primary:node0}
user@host# show interfaces ge-0/0/3
disable;
gigether-options {
    redundant-parent reth2;
}
```

**Meaning** The sample output confirms that interface ge-0/0/3 is disabled.

### *Verifying Chassis Cluster Status After Disabling Interface ge-0/0/3*

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

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

```
{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring     IP IP monitoring
  LB Loopback monitoring      MB Mbuf monitoring
  NH Nexthop monitoring       NP NPC monitoring
  SP SPU monitoring           SM Schedule monitoring
  CF Config Sync monitoring
```

```

Cluster ID: 2
Node   Priority Status           Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  254      primary          no    no    None
node1  1        secondary       no    no    None

Redundancy group: 1 , Failover count: 2
node0  0        secondary       no    no    IF
node1  100      primary        no    no    None

Redundancy group: 2 , Failover count: 2
node0  0        secondary       no    no    IF
node1  100      primary        no    no    None

```

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

### *Verifying Chassis Cluster Interfaces After Disabling Interface ge-0/0/3*

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

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

```

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

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

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
              (Physical/Monitored)
  fab0    ge-0/0/0         Up / Up
  fab0
  fab1    ge-8/0/0         Up / Up
  fab1

Redundant-ethernet Information:
  Name    Status  Redundancy-group
  reth0    Up      1
  reth1    Up      1
  reth2    Up      2

Redundant-pseudo-interface Information:
  Name    Status  Redundancy-group
  lo0     Up      0

Interface Monitoring:

```

Interface	Weight	Status	Redundancy-group
ge-8/0/2	120	Up	1
ge-8/0/1	150	Up	1
ge-0/0/2	140	Down	1
ge-0/0/1	130	Down	1
ge-8/0/3	255	Up	2
ge-0/0/3	255	Down	2

**Meaning** The sample output confirms that monitoring interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 are down.

### *Verifying Chassis Cluster Information After Disabling Interface ge-0/0/3*

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitoring interfaces in the cluster.

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

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

node0:
-----
Redundancy Group Information:

Redundancy Group 0 , Current State: primary, Weight: 255

    Time           From           To           Reason
    Feb 24 22:56:27 hold           secondary    Hold timer expired
    Feb 24 22:56:34 secondary    primary      Better priority (254/1)

Redundancy Group 1 , Current State: secondary, Weight: -15

    Time           From           To           Reason
    Feb 24 23:16:12 hold           secondary    Hold timer expired
    Feb 24 23:16:12 secondary    primary      Remote yield (0/0)
    Feb 24 23:31:36 primary      secondary-hold Monitor failed: IF
    Feb 24 23:31:37 secondary-hold secondary      Ready to become secondary

Redundancy Group 2 , Current State: secondary, Weight: 0

    Time           From           To           Reason
    Feb 24 23:16:12 hold           secondary    Hold timer expired
    Feb 24 23:16:13 secondary    primary      Remote yield (0/0)
    Feb 24 23:35:57 primary      secondary-hold Monitor failed: IF
    Feb 24 23:35:58 secondary-hold secondary      Ready to become secondary

Chassis cluster LED information:
  Current LED color: Amber
  Last LED change reason: Monitored objects are down

Failure Information:

  Interface Monitoring Failure Information:
    Redundancy Group 1, Monitoring status: Failed
```

```

Interface          Status
ge-0/0/2           Down
ge-0/0/1           Down
Redundancy Group 2, Monitoring status: Failed
Interface          Status
ge-0/0/3           Down

```

node1:

-----  
 Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

```

Time      From      To      Reason
Feb 24 22:56:34 hold      secondary Hold timer expired

```

Redundancy Group 1 , Current State: primary, Weight: 255

```

Time      From      To      Reason
Feb 24 23:16:10 hold      secondary Hold timer expired
Feb 24 23:31:36 secondary primary Remote is in secondary hold

```

Redundancy Group 2 , Current State: primary, Weight: 255

```

Time      From      To      Reason
Feb 24 23:16:10 hold      secondary Hold timer expired
Feb 24 23:35:57 secondary primary Remote is in secondary hold

```

Chassis cluster LED information:

Current LED color: Amber

Last LED change reason: Monitored objects are down

**Meaning** The sample output confirms that in node 0, monitoring interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 are down.



**NOTE:** In regard to RG1, allowing any interface in node 0 go up triggers a failover only if the **preempt** option is enabled. In the example, **preempt** is not enabled. Therefore the node should return to normal, with no monitor failure showing for RG1.

### *Verifying That Interface ge-0/0/2 Is Enabled*

**Purpose** Verify that interface ge-0/0/2 is enabled on node 0.

**Action** From configuration mode, enter the **delete interfaces ge-0/0/2 disable** command.

```

{primary:node0}
user@host# delete interfaces ge-0/0/2 disable
user@host# commit

```

```

node0:
configuration check succeeds
node1:
commit complete
node0:
commit complete

```

**Meaning** The sample output confirms that interface ge-0/0/2 disable is deleted.

### *Verifying Chassis Cluster Status After Enabling Interface ge-0/0/2*

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

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

```

{primary:node0}
user@host> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring      FL Fabric Connection monitoring
  GR GRES monitoring          HW Hardware monitoring
  IF Interface monitoring      IP IP monitoring
  LB Loopback monitoring       MB Mbuf monitoring
  NH Nexthop monitoring        NP NPC monitoring
  SP SPU monitoring            SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 2
Node  Priority Status      Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 254 primary      no    no    None
node1 1  secondary    no    no    None

Redundancy group: 1 , Failover count: 2
node0 200 secondary    no    no    None
node1 100 primary      no    no    None

Redundancy group: 2 , Failover count: 2
node0 0  secondary    no    no    IF
node1 100 primary      no    no    None

```

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

### *Verifying Chassis Cluster Interfaces After Enabling Interface ge-0/0/2*

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

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

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

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

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status
              (Physical/Monitored)
  fab0    ge-0/0/0        Up / Up
  fab0    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     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.

**See Also**

- [Example: Configuring Chassis Cluster Redundancy Groups on page 138](#)

**Related Documentation**

- [Monitoring IP Addresses on a Chassis Cluster on page 229](#)
- [Configuring Cluster Failover Parameters on page 245](#)
- [Chassis Cluster Redundancy Group Failover on page 249](#)

## Monitoring IP Addresses on a Chassis Cluster

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

Redundancy group IP address monitoring checks end-to-end connectivity and allows a redundancy group to fail over if reth interface fails to reach a configured IP address. Redundancy groups on both devices in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable. For more information, see the following topics:

- [IP Monitoring Overview on page 230](#)
- [Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 232](#)
- [Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring on page 235](#)
- [Example: Configuring IP Monitoring on SRX5000 Line Devices for IOC2 and IOC3 on page 238](#)

## IP Monitoring Overview

IP monitoring checks the end-to-end connectivity of configured IP addresses and allows a redundancy group to automatically fail over when the monitored IP address is not reachable through the redundant Ethernet (reth) interface. Both the primary and secondary nodes in the chassis cluster monitor specific IP addresses to determine whether an upstream device in the network is reachable.

IP monitoring allows for failover based upon end to-end reachability of a configured monitored IP address. On SRX Series devices, the reachability test is done by sending a ping to the monitored IP address from both the primary node and the secondary node through the reth interface and checking if a response is returned. The monitored IP address can be on a directly connected host in the same subnet as the reth interface or on a remote device reachable through a next-hop router.

The reachability states of the monitored IP address are reachable, unreachable, and unknown. The status is "unknown" if Packet Forwarding Engines are not yet up and running. The status changes to either "reachable" or "unreachable," depending on the corresponding message from the Packet Forwarding Engine.



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

[Table 23 on page 230](#) provides details of different combinations of monitored results from both the primary and secondary nodes, and the corresponding actions by the Juniper Services Redundancy Protocol (jsrpd) process.

**Table 23: IP Monitoring Results and Failover Action**

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

**Table 23: IP Monitoring Results and Failover Action (continued)**

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

**NOTE:**

- You can configure up to 64 IP addresses for IP monitoring on SRX5000 line devices.
- The minimum interval of IP monitoring is 1 second and the maximum is 30 seconds. Default interval is 1 second.
- The minimum threshold of IP monitoring is 5 requests and the maximum is 15 requests. If the IP monitoring request does not receive a response for consecutive requests (exceeding the threshold value), IP monitoring reports that the monitored IP is unreachable. Default value for the threshold is 5.
- Reth interface not associated with Redundancy Group (RG) in IP monitoring CLI configuration is supported.

Table 24 on page 231 provides details on multiple interface combinations of IOC2 and IOC3 with maximum MAC numbers.

**Table 24: Maximum MACs Supported for IP Monitoring on IOC2 and IOC3**

Cards	Interfaces	Maximum MACs Supported for IP Monitoring
IOC2 (SRX5K-MPC)	10XGE	10
	20GE	20
	2X40GE	2
	1X100GE	1
IOC3 (SRX5K-MPC3-40G10G or SRX5K-MPC3-100G10G)	24x10GE	24
	6x40GE	6
	2x100GE + 4x10GE	6

Note the following limitations for IP monitoring support on SRX5000 line IOC2 and IOC3:

- IP monitoring is supported through the reth or the RLAG interface. If your configuration does not specify either of these interfaces, the route lookup returns a non-reth/RLAG interface, which results in a failure report.
- Equal-cost multipath (ECMP) routing is not supported in IP monitoring.

### Benefits of Monitoring IP Addresses in a Chassis Cluster

---

- Helps determine the status of a specific IP address in a Chassis Cluster setup as unknown, reachable or unreachable.
- Initiates failover based upon end-to-end reachability of a configured monitored IP address. If the monitored IP address becomes unreachable, the redundancy group can fail over to its backup to maintain service.

- See Also**
- [SRX5400, SRX5600, and SRX5800 Services Gateway Card Overview](#)
  - [Chassis Cluster Redundancy Groups on page 134](#)

## Understanding Chassis Cluster Redundancy Group IP Address Monitoring

Redundancy group IP address monitoring checks end-to-end connectivity and allows a redundancy group to fail over because of the inability of a redundant Ethernet interface (known as a *reth*) to reach a configured IP address. Redundancy groups on both devices in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable. The redundancy group can be configured such that if the monitored IP address becomes unreachable, the redundancy group will fail over to its backup to maintain service. The primary difference between this monitoring feature and interface monitoring is that IP address monitoring allows for failover when the interface is still up but the network device it is connected to is not reachable for some reason. It may be possible under those circumstances for the other node in the cluster to route traffic around the problem.



**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 fallback unless the preempt option has been enabled.



When configured, the IP address monitoring failover value (global-weight) is considered along with interface monitoring—if set—and built-in failover monitoring, including SPU monitoring, cold-sync monitoring, and NPC monitoring (on supported platforms). The main IP addresses that should be monitored are router gateway addresses to ensure that valid traffic coming into the services gateway can be forwarded to the appropriate network router.

Starting in Junos OS Release 12.1X46-D35 and Junos OS Release 17.3R1, for all SRX Series 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.



**NOTE:** IP address monitoring is not supported on SRX5000 line devices if the redundant Ethernet interface is configured for a VPN routing and forwarding (VRF) instance.

The default interval to check the reachability of a monitored IP address is once per second. The interval can be adjusted using the **retry-interval** command. The default number of permitted consecutive failed ping attempts is 5. The number of allowed consecutive failed ping attempts can be adjusted using the **retry-count** command. After failing to reach a monitored IP address for the configured number of consecutive attempts, the IP address is determined to be unreachable and its failover value is deducted from the redundancy group's global-threshold.



**NOTE:** On SRX5600 and SRX5800 devices, only two of the 10 ports on each PIC of 40-port 1-Gigabit Ethernet I/O cards (IOCs) can simultaneously enable IP address monitoring. Because there are four PICs per IOC, this permits a total of eight ports per IOC to be monitored. If more than two ports per PIC on 40-port 1-Gigabit Ethernet IOCs are configured for IP address monitoring, the commit will succeed but a log entry will be generated, and the accuracy and stability of IP address monitoring cannot be ensured. This limitation does not apply to any other IOCs or devices.

Once the IP address is determined to be unreachable, its weight is deducted from the global-threshold. If the recalculated global-threshold value is not 0, the IP address is marked unreachable, but the global-weight is not deducted from the redundancy group's

threshold. If the redundancy group IP monitoring global-threshold reaches 0 and there are unreachable IP addresses, the redundancy group will continuously fail over and fail back between the nodes until either an unreachable IP address becomes reachable or a configuration change removes unreachable IP addresses from monitoring. Note that both default and configured hold-down-interval failover dampening is still in effect.

Every redundancy group *x* has a threshold tolerance value initially set to 255. When an IP address monitored by redundancy group *x* becomes unavailable, its weight is subtracted from the redundancy group *x*'s threshold. When redundancy group *x*'s threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

A redundancy group *x* failover occurs because the cumulative weight of the redundancy group *x*'s monitored IP addresses and other monitoring has brought its threshold value to 0. When the monitored IP addresses of redundancy group *x* on both nodes reach their thresholds at the same time, redundancy group *x* is primary on the node with the lower node ID, which is typically node 0.



**NOTE:** Upstream device failure detection for the chassis cluster feature is supported on SRX Series devices.

---

Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, configuring Address Resolution Protocol (ARP) request throttling is supported on SRX5000 line devices. This feature allows you to bypass the previously hard-coded ARP request throttling time default (10 seconds per SPU for each IP address) and set the time to a greater value (10 through 100 seconds). Setting the throttling time to a greater value reduces the high utilization of the Routing Engine, allowing it to work more efficiently. You can configure the ARP request throttling time using the **set forwarding-options next-hop arp-throttle <seconds>** command.

Monitoring can be accomplished only if the IP address is reachable on a redundant Ethernet interface (known as a reth in CLI commands and interface listings), and IP addresses cannot be monitored over a tunnel. For an IP address to be monitored through a redundant Ethernet interface on a secondary cluster node, the interface must have a secondary IP address configured. IP address monitoring cannot be used on a chassis cluster running in transparent mode. The maximum number of monitoring IP addresses that can be configured per cluster is 64 for the SRX5000 line and 32 for the SRX1400 device and the SRX3000 line.



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

---

## Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring

This example shows how to configure redundancy group IP address monitoring for an SRX Series device in a chassis cluster.

- [Requirements on page 235](#)
- [Overview on page 235](#)
- [Configuration on page 236](#)
- [Verification on page 237](#)

### Requirements

Before you begin:

- Set the chassis cluster node ID and cluster ID. See “[Example: Setting the Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster](#)” on page 110
- Configure the chassis cluster management interface. See “[Example: Configuring the Chassis Cluster Management Interface](#)” on page 114.
- Configure the chassis cluster fabric. See “[Example: Configuring the Chassis Cluster Fabric Interfaces](#)” on page 125.

### 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

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

This example shows how to monitor IP address on a an SRX5000 line device with chassis cluster enabled.

- [Requirements on page 238](#)
- [Overview on page 238](#)
- [Configuration on page 239](#)
- [Verification on page 244](#)

---

### Requirements

This example uses the following hardware and software:

- Two SRX5400 Services Gateways with MIC (SRX-MIC-10XG-SFPP [IOC2]), and one Ethernet switch
- Junos OS Release 15.1X49-D30

The procedure mentioned in this example are also applicable to IOC3 also.

Before you begin:

- Physically connect the two SRX5400 devices (back-to-back for the fabric and control ports).
- Configure the two devices to operate in a chassis cluster.

---

### Overview

IP address monitoring checks end-to-end reachability of the configured IP address and allows a redundancy group to automatically fail over when it is not reachable through the child link of redundant Ethernet (reth) interface. Redundancy groups on both devices, or nodes, in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable.

### Topology

In this example, two SRX5400 devices in a chassis cluster are connected to an Ethernet switch. The example shows how the redundancy groups can be configured to monitor key upstream resources reachable through redundant Ethernet interfaces on either node in a cluster.

You set the system to send pings every second, with 10 losses required to declare unreachability to peer. You also set up a secondary IP address to allow testing from the secondary node.

In this example, you configure the following settings for redundancy group 1:

- IP address to be monitored—192.0.2.2, 198.51.100.2, 203.0.113.2
- IP monitoring global-weight—255
- IP monitoring global-threshold—240
- IP monitoring retry-interval—3 seconds
- IP monitoring retry-count—10
- Weight for monitored IP address—80
- Secondary IP addresses— 192.0.2.12, 198.51.100.12, 203.0.113.12

### Configuration

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

```
set chassis cluster reth-count 10
set chassis cluster control-ports fpc 3 port 0
set chassis cluster control-ports fpc 0 port 0
set chassis cluster redundancy-group 0 node 0 priority 254
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 199
set chassis cluster redundancy-group 1 ip-monitoring global-weight 255
set chassis cluster redundancy-group 1 ip-monitoring global-threshold 240
set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
set chassis cluster redundancy-group 1 ip-monitoring family inet 192.0.2.2 weight 80
set chassis cluster redundancy-group 1 ip-monitoring family inet 192.0.2.2
interface reth0.0 secondary-ip-address 192.0.2.12
set chassis cluster redundancy-group 1 ip-monitoring family inet 198.51.100.2
weight 80
set chassis cluster redundancy-group 1 ip-monitoring family inet 198.51.100.2
interface reth1.0 secondary-ip-address 198.51.100.12
set chassis cluster redundancy-group 1 ip-monitoring family inet 203.0.113.1
weight 80
set chassis cluster redundancy-group 1 ip-monitoring family inet 203.0.113.1
interface reth2.0 secondary-ip-address 203.0.113.12
set interfaces xe-1/2/1 gigether-options redundant-parent reth0
set interfaces xe-1/2/2 gigether-options redundant-parent reth2
set interfaces xe-1/2/3 gigether-options redundant-parent reth1
set interfaces xe-4/2/1 gigether-options redundant-parent reth0
set interfaces xe-4/2/2 gigether-options redundant-parent reth2
set interfaces xe-4/2/3 gigether-options redundant-parent reth1
set interfaces fab0 fabric-options member-interfaces xe-1/2/0
set interfaces fab1 fabric-options member-interfaces xe-4/2/0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.0.2.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 198.51.100.1/24
set interfaces reth2 redundant-ether-options redundancy-group 1
```

```

set interfaces reth2 unit 0 family inet address 203.0.113.1/24
set security zones security-zone HOST host-inbound-traffic system-services
any-service
set security zones security-zone HOST host-inbound-traffic protocols all
set security zones security-zone HOST interfaces all

```

### *Configuring IP Monitoring on a 10x10GE SFP+ MIC*

#### **Step-by-Step Procedure**

To configure IP monitoring on a 10x10GE SFP+ MIC:

1. Specify the number of redundant Ethernet interfaces.  

```

{primary:node0}[edit]
user@host# set chassis cluster reth-count 10

```
2. Configure the control ports.  

```

{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 3 port 0
user@host# set chassis cluster control-ports fpc 0 port 0

```
3. Configure fabric interfaces.  

```

{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces xe-1/2/0
user@host# set interfaces fab1 fabric-options member-interfaces xe-4/2/0

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

```

{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 200
user@host# set chassis cluster redundancy-group 1 node 1 priority 199

```
5. Configure IP monitoring under redundancy-group 1 with global weight, global threshold, retry interval and retry count.  

```

{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-weight
255
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-threshold
240
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-count 10

```
6. Configure the redundant Ethernet interfaces to redundancy-group 1. Assign a weight to the IP address to be monitored, and configure a secondary IP address that will be used to send packets from the secondary node to track the IP address being monitored.  

```

{primary:node0}[edit]

```



```

user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet
192.0.2.2 weight 80
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet
192.0.2.2 interface reth0.0 secondary-ip-address 192.0.2.12
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet
198.51.100.2 weight 80
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet
198.51.100.2 interface reth1.0 secondary-ip-address 198.51.100.12
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet
203.0.113.2 weight 80
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet
203.0.113.2 interface reth2.0 secondary-ip-address 203.0.113.12

```

7. Assign child interfaces for the redundant Ethernet interfaces from node 0, node 1, and node 2.

```

{primary:node0}[edit]
user@host# set interfaces xe-1/2/1 gigether-options redundant-parent reth0
user@host# set interfaces xe-1/2/2 gigether-options redundant-parent reth2
user@host# set interfaces xe-1/2/3 gigether-options redundant-parent reth1
user@host# set interfaces xe-4/2/1 gigether-options redundant-parent reth0
user@host# set interfaces xe-4/2/2 gigether-options redundant-parent reth2
user@host# set interfaces xe-4/2/3 gigether-options redundant-parent reth1

```

8. Configure the redundant Ethernet interfaces to redundancy-group 1.

```

{primary:node0}[edit]
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 192.0.2.1/24
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 198.51.100.1/24
user@host# set interfaces reth2 redundant-ether-options redundancy-group 1
user@host# set interfaces reth2 unit 0 family inet address 203.0.113.1/24

```

9. Create security zone and assign interfaces to zone.

```

user@host# set security zones security-zone HOST host-inbound-traffic
system-services any-service
user@host# set security zones security-zone HOST host-inbound-traffic protocols
all
user@host# set security zones security-zone HOST interfaces all

```

**Results** From configuration mode, confirm your configuration by entering the **show security chassis cluster** and **show interfaces** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

chassis {
  cluster {
    reth-count 10;
    redundancy-group 0 {
      node 0 priority 254;
      node 1 priority 1;
    }
  }
}

```

Copyright © 2018, Juniper Networks, Inc.

```

        redundant-parent reth1;
    }
}
fab0 {
    fabric-options {
        member-interfaces {
            xe-1/2/0;
        }
    }
}
fab1 {
    fabric-options {
        member-interfaces {
            xe-4/2/0;
        }
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 198.51.100.1/24;
        }
    }
}
reth2 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family inet {
            address 203.0.113.1/24;
        }
    }
}
}

```

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

## Verification

---

Confirm the configuration is working properly.

- [Verifying IP Monitoring Status on page 244](#)

### *Verifying IP Monitoring Status*

**Purpose** Verify the IP status being monitored from both nodes and the failure count for both nodes.

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

**show chassis cluster ip-monitoring status**

node0:

-----  
Redundancy group: 1  
Global weight: 255  
Global threshold: 240  
Current threshold: 240

IP address	Status	Failure count	Weight	Reason
203.0.113.2	reachable	1	80	n/a
198.51.100.2	reachable	1	80	n/a
192.0.2.2	reachable	1	80	n/a

node1:

-----  
Redundancy group: 1  
Global weight: 255  
Global threshold: 240  
Current threshold: 240

IP address	Status	Failure count	Weight	Reason
203.0.113.2	reachable	2	80	n/a
198.51.100.2	reachable	1	80	n/a
192.0.2.2	reachable	2	80	n/a

**Meaning** All the monitored IP addresses are reachable.

**Release History Table**

Release	Description
15.1X49-D60	Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, configuring Address Resolution Protocol (ARP) request throttling is supported on SRX5000 line devices.
12.1X46-D35	Starting in Junos OS Release 12.1X46-D35 and Junos OS Release 17.3R1, for all SRX Series devices, the reth interface supports proxy ARP.

**Related Documentation**

- [Chassis Cluster Redundancy Groups on page 134](#)
- [Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Devices on page 299](#)
- [Monitoring Chassis Cluster Interfaces on page 200](#)

## Configuring Cluster Failover Parameters

---

**Supported Platforms** [SRX5600, SRX5800, vSRX](#)

SRX Series devices in a chassis cluster uses heartbeat transmissions to determine the “health” of the control link. If the number of missed heartbeats has reached the configured threshold, the system assesses whether a failure condition exists. For more information, see the following topics:

- [Understanding Chassis Cluster Control Link Heartbeats, Failure, and Recovery on page 245](#)
- [Example: Configuring Chassis Cluster Control Link Recovery on page 248](#)

## Understanding Chassis Cluster Control Link Heartbeats, Failure, and Recovery

### Understanding Chassis Cluster Control Link Heartbeats

---

You specify the heartbeat threshold and heartbeat interval when you configure the chassis cluster.

The system monitors the control link's status by default.

For dual control links, which are supported on SRX5600 and SRX5800 lines, the Juniper Services Redundancy Protocol process (jsrpd) sends and receives the control heartbeat messages on both control links. As long as heartbeats are received on one of the control links, Junos OS considers the other node to be alive.

The product of the **heartbeat-threshold** option and the **heartbeat-interval** option defines the wait time before failover is triggered. The default values of these options produce a wait time of 3 seconds. A heartbeat-threshold of 5 and a heartbeat-interval of 1000 milliseconds would yield a wait time of 5 seconds. Setting the heartbeat-threshold to 4 and the heartbeat-interval to 1250 milliseconds would also yield a wait time of 5 seconds.

In a chassis cluster environment, if more than 1000 logical interfaces are used, the cluster heartbeat timers are recommended to be increased from the default of 3 seconds. At

maximum capacity on an SRX4600, SRX5400, SRX5600 or an SRX5800 device, we recommend that you increase the configured time before failover to at least 5 seconds. In a large chassis cluster configuration on an SRX3400 or SRX3600 device, we recommend increasing the wait to 8 seconds.

### Understanding Chassis Cluster Control Link Failure and Recovery

If the control link fails, Junos OS changes the operating state of the secondary node to ineligible for a 180-second countdown. If the fabric link also fails during the 180 seconds, Junos OS changes the secondary node to primary; otherwise, after 180 seconds the secondary node state changes to disabled.

When the control link is down, a system log message is generated.

A control link failure is defined as not receiving heartbeats over the control link while heartbeats are still being received over the fabric link.

In the event of a legitimate control link failure, redundancy group 0 remains primary on the node on which it is currently primary, inactive redundancy groups x on the primary node become active, and the secondary node enters a disabled state.



**NOTE:** When the secondary node is disabled, you can still log in to the management port and run diagnostics.

---

To determine if a legitimate control link failure has occurred, the system relies on redundant liveliness signals sent across both the control link and the fabric link.

The system periodically transmits probes over the fabric link and heartbeat signals over the control link. Probes and heartbeat signals share a common sequence number that maps them to a unique time event. Junos OS identifies a legitimate control link failure if the following two conditions exist:

- The threshold number of heartbeats were lost.
- At least one probe with a sequence number corresponding to that of a missing heartbeat signal was received on the fabric link.

If the control link fails, the 180-second countdown begins and the secondary node state is ineligible. If the fabric link fails before the 180-second countdown reaches zero, the secondary node becomes primary because the loss of both links is interpreted by the system to indicate that the other node is dead. Because concurrent loss of both control and fabric links means that the nodes are no longer synchronizing states nor comparing priorities, both nodes might thus temporarily become primary, which is not a stable operating state. However, once the control link is reestablished, the node with the higher priority value automatically becomes primary, the other node becomes secondary, and the cluster returns to normal operation.

When a legitimate control link failure occurs, the following conditions apply:

- Redundancy group 0 remains primary on the node on which it is currently primary (and thus its Routing Engine remains active), and all redundancy groups x on the node become primary.

If the system cannot determine which Routing Engine is primary, the node with the higher priority value for redundancy group 0 is primary and its Routing Engine is active. (You configure the priority for each node when you configure the **redundancy-group** statement for redundancy group 0.)

- The system disables the secondary node.

To recover a device from the disabled mode, you must reboot the device. When you reboot the disabled node, the node synchronizes its dynamic state with the primary node.



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

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

When you use dual control links (supported on SRX5600 and SRX5800 devices), note the following conditions:

- Host inbound or outbound traffic can be impacted for up to 3 seconds during a control link failure. For example, consider a case where redundancy group 0 is primary on node 0 and there is a Telnet session to the Routing Engine through a network interface port on node 1. If the currently active control link fails, the Telnet session will lose packets for 3 seconds, until this failure is detected.
- A control link failure that occurs while the commit process is running across two nodes might lead to commit failure. In this situation, run the **commit** command again after 3 seconds.



**NOTE:** For SRX5600 and SRX5800 devices, dual control links require a second Routing Engine on each node of the chassis cluster.

You can specify that control link recovery be done automatically by the system by setting the **control-link-recovery** statement. In this case, once the system determines that the control link is healthy, it issues an automatic reboot on the disabled node. When the disabled node reboots, the node joins the cluster again.

## Example: Configuring Chassis Cluster Control Link Recovery

This example shows how to enable control link recovery, which allows the system to automatically take over after the control link recovers from a failure.

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

---

### Requirements

Before you begin:

- Understand chassis cluster control links. See “[Understanding Chassis Cluster Control Plane and Control Links](#)” on page 129.
- Understand chassis cluster dual control links. See “[Understanding Chassis Cluster Dual Control Links](#)” on page 183.
- Connect dual control links in a chassis cluster. See “[Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster](#)” on page 185.

---

### Overview

You can enable the system to perform control link recovery automatically. After the control link recovers, the system takes the following actions:

- It checks whether it receives at least 30 consecutive heartbeats on the control link or, in the case of dual control links (SRX5600 and SRX5800 devices only), on either control link. This is to ensure that the control link is not flapping and is healthy.
- After it determines that the control link is healthy, the system issues an automatic reboot on the node that was disabled when the control link failed. When the disabled node reboots, it can rejoin the cluster. There is no need for any manual intervention.

In this example, you enable chassis cluster control link recovery.

---

### Configuration

#### Step-by-Step Procedure

To enable chassis cluster control-link-recovery:

1. Enable control link recovery.  

```
{primary:node0}[edit]  
user@host# set chassis cluster control-link-recovery
```
2. If you are done configuring the device, commit the configuration.  

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



- Related Documentation**
- [Chassis Cluster Dual Control Links on page 183](#)
  - [Example: Configuring Chassis Cluster Control Ports for Dual Control Links on page 188](#)

---

## Chassis Cluster Redundancy Group Failover

---

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

A redundancy group (RG) includes and manages a collection of objects on both nodes of a cluster to provide high-availability. Each redundancy group acts as an independent unit of failover and is primary on only one node at a time. For more information, see the following topics:

- [Understanding Chassis Cluster Redundancy Group Failover on page 249](#)
- [Understanding Chassis Cluster Redundancy Group Manual Failover on page 253](#)
- [Initiating a Chassis Cluster Manual Redundancy Group Failover on page 255](#)
- [Example: Configuring a Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers on page 257](#)
- [Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 258](#)
- [Verifying Chassis Cluster Failover Status on page 259](#)
- [Clearing Chassis Cluster Failover Status on page 260](#)

### Understanding Chassis Cluster Redundancy Group Failover

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.

Configuring the hold-down interval prevents back-to-back failovers from occurring within the duration of hold-down interval.

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

The default dampening time for a redundancy group 0 is 300 seconds (5 minutes) and is configurable to up to 1800 seconds with the **hold-down-interval** statement. For some configurations, such as those with a large number of routes or logical interfaces, the default interval or the user-configured interval might not be sufficient. In such cases, the system automatically extends the dampening time in increments of 60 seconds until the system is ready for failover.

Redundancy groups x (redundancy groups numbered 1 through 128) have a default dampening time of 1 second, with a range from 0 through 1800 seconds.

On SRX Series 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).

---

### Preemptive Failover Delay Timer

---

A redundancy group is in the primary state (active) on one node and in the secondary state (backup) on the other node at any given time.

You can enable the preemptive behavior on both nodes in a redundancy group and assign a priority value for each node in the redundancy group. The node in the redundancy group with the higher configured priority is initially designated as the primary in the group, and the other node is initially designated as the secondary in the redundancy group.

When a redundancy group swaps the state of its nodes between primary and secondary, there is a possibility that a subsequent state swap of its nodes can happen again soon after the first state swap. This rapid change in states results in flapping of the primary and secondary systems.

Starting with Junos OS Release 17.4R1, a failover delay timer is introduced on SRX Series devices in a chassis cluster to limit the flapping of redundancy group state between the secondary and the primary nodes in a preemptive failover.

To prevent the flapping, you can configure the following parameters:

- **Preemptive delay**—The preemptive delay time is the amount of time a redundancy group in a secondary state waits when the primary state is down in a preemptive failover before switching to the primary state. This delay timer delays the immediate failover for a configured period of time—between 1 and 21,600 seconds.
- **Preemptive limit**—The preemptive limit restricts the number of preemptive failovers (between 1 to 50) during a configured preemptive period, when **preemption** is enabled for a redundancy group.
- **Preemptive period**—Time period (1 to 1440 seconds) during which the preemptive limit is applied, that is, number of configured preemptive failovers are applied when preempt is enabled for a redundancy group.

Consider the following scenario where you have configured a preemptive period as 300 seconds and preemptive limit as 50.

When the preemptive limit is configured as 50, the count starts at 0 and increments with a first preemptive failover; this process continues until the count reaches the configured preemptive limit, that is 50, before the preemptive period expires. When the preemptive limit (50) is exceeded, you must manually reset the preempt count to allow preemptive failovers to occur again.

When you have configured the preemptive period as 300 seconds, and if the time difference between the first preemptive failover and the current failover has already exceeded 300 seconds, and the preemptive limit (50) is not yet reached, then the preemptive period will be reset. After resetting, the last failover is considered as the first preemptive failover of the new preemptive period and the process starts all over again.



**NOTE:** The preemptive delay can be configured independent of the failover limit. Configuring the preemptive delay timer does not change the existing preemptive behavior.

---

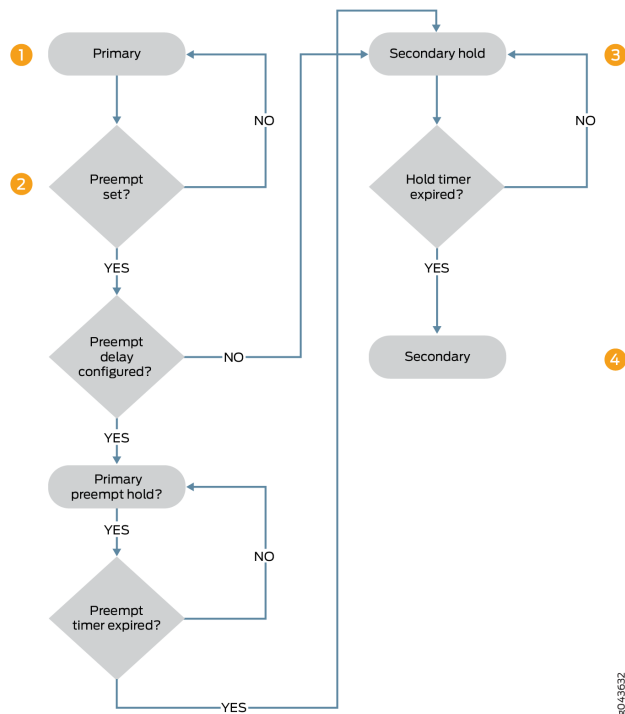
This enhancement enables the administrator to introduce a failover delay, which can reduce the number of failovers and result in a more stable network state due to the reduction in active /standby flapping within the redundancy group.

#### ***Understanding Transition from Primary State to Secondary State with Preemptive Delay***

Consider the following example, where a redundancy group, that is primary on the node 0 is ready for preemptive transition to the secondary state during a failover. Priority is assigned to each node and the **preemptive** option is also enabled for the nodes.

[Figure 50 on page 252](#) illustrates the sequence of steps in transition from the primary state to the secondary state when a preemptive delay timer is configured.

Figure 50: Transition from Primary State to Secondary State with Preemptive Delay



1. The node in the primary state is ready for preemptive transition to secondary state if the **preemptive** option is configured, and the node in secondary state has the priority over the node in primary state. If the preemptive delay is configured, the node in the primary state transitions to primary-preempt-hold state. If preemptive delay is not configured, then instant transition to the secondary state happens.
2. The node is in primary-preempt-hold state waiting for the preemptive delay timer to expire. The preemptive delay timer is checked and transition is held until the timer expires. The primary node stays in the primary-preempt-hold state until the timer expires, before transitioning to the secondary state.
3. The node transitions from primary-preempt-hold state into secondary-hold state and then to the secondary state.
4. The node stays in the secondary-hold state for the default time (1 second) or the configured time (a minimum of 300 seconds), and then the node transitions to the secondary state.



**CAUTION:** If your chassis cluster setup experiences an abnormal number of flaps, you must check your link and monitoring timers to make sure they are set correctly. Be careful when while setting timers in high latency networks to avoid getting false positives.

### Configuring Preemptive Delay Timer

This topic explains how to configure the delay timer on SRX Series devices in a chassis cluster. Back-to-back redundancy group failovers that occur too quickly can cause a chassis cluster to exhibit unpredictable behavior. Configuring the delay timer and failover rate limit delays immediate failover for a configured period of time.

To configure the preemptive delay timer and failover rate limit between redundancy group failovers:

1. Enable preemptive failover for a redundancy group.

You can set the delay timer between 1 and 21,600 seconds. Default value is 1 second.

```
{primary:node1}
[edit chassis cluster redundancy-group number preempt]
user@host# set delay interval
```

2. Set up a limit for preemptive failover.

You can set maximum number of preemptive failovers between 1 to 50 and time period during which the limit is applied between 1 to 1440 seconds.

```
{primary:node1}[edit chassis cluster redundancy-group number preempt]
user@host# set limit limit period period
```

In the following example, you are setting the preemptive delay timer to 300 seconds, and the preemptive limit to 10 for a preemptive period of 600 seconds. That is, this configuration delays immediate failover for 300 seconds, and it limits a maximum of 10 preemptive failovers in a duration of 600 seconds.

```
{primary:node1}[edit chassis cluster redundancy-group 1 preempt]
user@host# set delay 300 limit 10 period 600
```

You can use the **clear chassis clusters preempt-count** command to clear the preempt failover counter for all redundancy groups. When a preempt limit is configured, the counter starts with a first preemptive failover and the count is reduced; this process continues until the count reaches zero before the timer expires. You can use this command to clear the preempt failover counter and reset it to start again.

**See Also** • [Chassis Cluster Redundancy Groups on page 134](#)

## Understanding Chassis Cluster Redundancy Group Manual Failover

You can initiate a redundancy group *x* (redundancy groups numbered 1 through 128) failover manually. A manual failover applies until a failback event occurs.

For example, suppose that you manually do a redundancy group 1 failover from node 0 to node 1. Then an interface that redundancy group 1 is monitoring fails, dropping the threshold value of the new primary redundancy group to zero. This event is considered a failback event, and the system returns control to the original redundancy group.

You can also initiate a redundancy group 0 failover manually if you want to change the primary node for redundancy group 0. You cannot enable preemption for redundancy group 0.



**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 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.



**NOTE:** Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.

Keep in mind that during an in-service software upgrade (ISSU), the transitions described here cannot happen. Instead, the other (primary) node transitions directly to the secondary state because Juniper Networks releases earlier than 10.0 do not interpret the secondary-hold state. While you start an ISSU, if one of the nodes has one or more redundancy groups in the secondary-hold state, you must wait for them to move to the secondary state before you can do manual failovers to make all the redundancy groups be primary on one node.



**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.

## Initiating a Chassis Cluster Manual Redundancy Group Failover

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 138](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 145](#)



**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.



**WARNING:** Unplugging the power cord and holding the power button to initiate a chassis cluster redundancy group failover might result in unpredictable behavior.



**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).

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

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 257](#)
- [Overview on page 258](#)
- [Configuration on page 258](#)

#### Requirements

Before you begin:

- Understand redundancy group failover. See “[Understanding Chassis Cluster Redundancy Group Failover](#)” on page 249.
- Understand redundancy group manual failover. See “[Understanding Chassis Cluster Redundancy Group Manual Failover](#)” on page 253.

## 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
```

## Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover

Chassis clustering supports SNMP traps, which are triggered whenever there is a redundancy group failover.

The trap message can help you troubleshoot failovers. It contains the following information:

- The cluster ID and node ID
- The reason for the failover
- The redundancy group that is involved in the failover
- The redundancy group's previous state and current state

These are the different states that a cluster can be in at any given instant: hold, primary, secondary-hold, secondary, ineligible, and disabled. Traps are generated for the following state transitions (only a transition from a hold state does not trigger a trap):

- primary <—> secondary
- primary —> secondary-hold
- secondary-hold —> secondary
- secondary —> ineligible
- ineligible —> disabled
- ineligible —> primary
- secondary —> disabled

A transition can be triggered because of any event, such as interface monitoring, SPU monitoring, failures, and manual failovers.

The trap is forwarded over the control link if the outgoing interface is on a node different from the node on the Routing Engine that generates the trap.

You can specify that a trace log be generated by setting the **traceoptions flag snmp** statement.

## Verifying Chassis Cluster Failover Status

**Purpose** Display the failover status of a chassis cluster.

**Action** From the CLI, enter the **show chassis cluster status** command:

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 3
Node name          Priority    Status    Preempt  Manual failover

Redundancy-group: 0, Failover count: 1
node0              254       primary   no       no
node1              2        secondary no       no
```

```
Redundancy-group: 1, Failover count: 1
node0              254       primary   no       no
node1              1        secondary no       no
```

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
Node              Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 5
node0             200       primary   no       no
node1             0        lost      n/a      n/a

Redundancy group: 1 , Failover count: 41
node0             101       primary   no       no
node1             0        lost      n/a      n/a
```

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
Node              Priority    Status    Preempt  Manual failover

Redundancy group: 0 , Failover count: 5
node0             200       primary   no       no
node1             0        unavailable n/a      n/a

Redundancy group: 1 , Failover count: 41
node0             101       primary   no       no
node1             0        unavailable n/a      n/a
```

## Clearing Chassis Cluster Failover Status

To clear the failover status of a chassis cluster, enter the **clear chassis cluster failover-count** command from the CLI:

```
{primary:node1}  
user@host> clear chassis cluster failover-count  
Cleared failover-count for all redundancy-groups
```

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, a failover delay timer is introduced on SRX Series devices in a chassis cluster to limit the flapping of redundancy group state between the secondary and the primary nodes in a preemptive failover.
12.1X47-D10	Starting with Junos OS Release 12.1X47-D10 and Junos OS Release 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.
12.1X46-D20	Starting with Junos OS Release 12.1X46-D20 and Junos OS Release 17.3R1, fabric monitoring is enabled by default. With this enabling, the node transitions directly to the ineligible state in case of fabric link failures.

### Related Documentation

- [Monitoring of Global-Level Objects in a Chassis Cluster on page 196](#)
- [Monitoring Chassis Cluster Interfaces on page 200](#)
- [Monitoring IP Addresses on a Chassis Cluster on page 229](#)

## CHAPTER 4

# Chassis Cluster Operations

- [Aggregated Ethernet Interfaces in a Chassis Cluster on page 261](#)
- [NTP Time Synchronization on Chassis Cluster on page 291](#)
- [Active/Passive Chassis Cluster Deployments on page 298](#)
- [Multicast Routing and Asymmetric Routing on Chassis Cluster on page 348](#)
- [Ethernet Switching on Chassis Cluster on page 362](#)
- [Media Access Control Security \(MACsec\) on Chassis Cluster on page 370](#)
- [Understanding SCTP Behavior in Chassis Cluster on page 384](#)

## Aggregated Ethernet Interfaces in a Chassis Cluster

---

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

IEEE 802.3ad link aggregation enables you to group Ethernet interfaces to form a single link layer interface, also known as a link aggregation group (LAG) or bundle. Reth LAG interfaces combine characteristics of reth interfaces and LAG interfaces. For more information, see the following topics:

- [Understanding Link Aggregation Groups in a Chassis Cluster on page 261](#)
- [Example: Configuring Link Aggregation Groups in a Chassis Cluster on page 263](#)
- [Understanding Link Aggregation Group Failover in a Chassis Cluster on page 267](#)
- [Understanding LACP on Chassis Clusters on page 269](#)
- [Example: Configuring LACP on Chassis Clusters on page 271](#)
- [Example: Configuring Chassis Cluster Minimum Links on page 274](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on an SRX5000 Line Device with IOC2 or IOC3 on page 275](#)
- [Understanding VRRP on SRX Series Devices on page 279](#)
- [Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces on page 282](#)

## Understanding Link Aggregation Groups in a Chassis Cluster

Support for Ethernet link aggregation groups (LAGs) based on IEEE 802.3ad makes it possible to aggregate physical interfaces on a standalone device. LAGs on standalone devices provide increased interface bandwidth and link availability. Aggregation of links

in a chassis cluster allows a redundant Ethernet interface to add more than two physical child interfaces thereby creating a redundant Ethernet interface LAG. 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 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*.

**See Also** • [Understanding Link Aggregation Control Protocol](#)

## Example: Configuring Link Aggregation Groups in a Chassis Cluster

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 264](#)
- [Overview on page 264](#)
- [Configuration on page 265](#)
- [Verification on page 266](#)

---

## Requirements

Before you begin:

- Configure chassis cluster redundant interfaces. See [“Example: Configuring Chassis Cluster Redundant Ethernet Interfaces” on page 145](#).
- Understand chassis cluster redundant Ethernet interface link aggregation groups. See [“Understanding Link Aggregation Groups in a Chassis Cluster” on page 261](#).

---

## Overview



**NOTE:** For aggregation to take place, the switch used to connect the nodes in the cluster must enable IEEE 802.3ad link aggregation for the redundant Ethernet interface physical child links on each node. Because most switches support IEEE 802.3ad and are also LACP capable, we recommend that you enable LACP on SRX Series devices. In cases where LACP is not available on the switch, you must not enable LACP on SRX Series devices.

---

In this example, you assign six Ethernet interfaces to reth1 to form the Ethernet interface link aggregation group:

- ge-1/0/1—reth1
  - ge-1/0/2—reth1
  - ge-1/0/3—reth1
  - ge-12/0/1—reth1
  - ge-12/0/2—reth1
  - ge-12/0/3—reth1
- 



**NOTE:** A maximum of eight physical interfaces per node in a cluster, for a total of 16 child interfaces, can be assigned to a single redundant Ethernet interface when a redundant Ethernet interface LAG is being configured.

---



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

---



## Configuration

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

```
{primary:node0}[edit]
set interfaces ge-1/0/1 gigether-options redundant-parent reth1
set interfaces ge-1/0/2 gigether-options redundant-parent reth1
set interfaces ge-1/0/3 gigether-options redundant-parent reth1
set interfaces ge-12/0/1 gigether-options redundant-parent reth1
set interfaces ge-12/0/2 gigether-options redundant-parent reth1
set interfaces ge-12/0/3 gigether-options redundant-parent reth1
```

**Step-by-Step Procedure** To configure a redundant Ethernet interface link aggregation group:

- Assign Ethernet interfaces to reth1.

```
{primary:node0}[edit]
user@host# set interfaces ge-1/0/1 gigether-options redundant-parent reth1
user@host# set interfaces ge-1/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-1/0/3 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/1 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/2 gigether-options redundant-parent reth1
user@host# set interfaces ge-12/0/3 gigether-options redundant-parent reth1
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces reth1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

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

```
user@host# show interfaces reth1
...
ge-1/0/1 {
  gigether-options {
    redundant-parent reth1;
  }
}
ge-1/0/2 {
  gigether-options {
    redundant-parent reth1;
  }
}
ge-1/0/3 {
  gigether-options {
    redundant-parent reth1;
  }
}
```

```
ge-12/0/1 {  
  gigether-options {  
    redundant-parent reth1;  
  }  
}  
ge-12/0/2 {  
  gigether-options {  
    redundant-parent reth1;  
  }  
}  
ge-12/0/3 {  
  gigether-options {  
    redundant-parent reth1;  
  }  
}  
...
```

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

---

## Verification

### *Verifying the Redundant Ethernet Interface LAG Configuration*

**Purpose** Verify the redundant Ethernet interface LAG configuration.

**Action** From operational mode, enter the **show interfaces terse | match reth** command.

```
{primary:node0}  
user@host> show interfaces terse | match reth  
ge-1/0/1.0          up    down aenet  --> reth1.0  
ge-1/0/2.0          up    down aenet  --> reth1.0  
ge-1/0/3.0          up    down aenet  --> reth1.0  
ge-12/0/1.0         up    down aenet  --> reth1.0  
ge-12/0/2.0         up    down aenet  --> reth1.0  
ge-12/0/3.0         up    down aenet  --> reth1.0  
reth0               up    down  
reth0.0             up    down inet   10.10.37.214/24  
reth1               up    down  
reth1.0             up    down inet
```

**See Also** • [Example: Configuring Aggregated Ethernet Device with LAG and LACP \(CLI Procedure\)](#)

## Understanding Link Aggregation Group Failover in a Chassis Cluster

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.
- When the physical link is Up and LACP is Down, a failover of the redundant ethernet interfaces link aggregation group (LAG) is triggered.

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 267](#)
- [Scenario 2: Monitored Interface Weight Is 75 on page 268](#)
- [Scenario 3: Monitored Interface Weight Is 100 on page 268](#)

### 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.

## Understanding LACP on Chassis Clusters

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 269](#)
- [Sub-LAGs on page 270](#)
- [Supporting Hitless Failover on page 270](#)
- [Managing Link Aggregation Control PDUs on page 271](#)

### Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

A redundant Ethernet interface has active and standby links located on two nodes in a chassis cluster. All active links are located on one node, and all standby links are located on the other node. You can configure up to eight active links and eight standby links per node.

When at least two physical child interface links from each node are included in a redundant Ethernet interface configuration, the interfaces are combined within the redundant Ethernet interface to form a redundant Ethernet interface LAG.

Having multiple active redundant Ethernet interface links reduces the possibility of failover. For example, when an active link is out of service, all traffic on this link is distributed to other active redundant Ethernet interface links, instead of triggering a redundant Ethernet active/standby failover.

Aggregated Ethernet interfaces, known as local LAGs, are also supported on either node of a chassis cluster but cannot be added to redundant Ethernet interfaces. Likewise, any child interface of an existing local LAG cannot be added to a redundant Ethernet interface, and vice versa. The total maximum number of combined individual node LAG interfaces (ae) and redundant Ethernet (reth) interfaces per cluster is 128.

However, aggregated Ethernet interfaces and redundant Ethernet interfaces can coexist, because the functionality of a redundant Ethernet interface relies on the Junos OS aggregated Ethernet framework.

For more information, see [“Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups”](#) on page 261.

### **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 274.

### **Sub-LAGs**

---

LACP maintains a point-to-point LAG. Any port connected to the third point is denied. However, a redundant Ethernet interface does connect to two different systems or two remote aggregated Ethernet interfaces by design.

To support LACP on redundant Ethernet interface active and standby links, a redundant Ethernet interface is created automatically to consist of two distinct sub-LAGs, where all active links form an active sub-LAG and all standby links form a standby sub-LAG.

In this model, LACP selection logic is applied and limited to one sub-LAG at a time. In this way, two redundant Ethernet interface sub-LAGs are maintained simultaneously while all the LACP advantages are preserved for each sub-LAG.

It is necessary for the switches used to connect the nodes in the cluster to have a LAG link configured and 802.3ad enabled for each LAG on both nodes so that the aggregate links are recognized as such and correctly pass traffic.



**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 271](#).

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.

**See Also** • [Example: Configuring LACP on Chassis Clusters on page 271](#)

### Example: Configuring LACP on Chassis Clusters

This example shows how to configure LACP on chassis clusters.

- [Requirements on page 271](#)
- [Overview on page 272](#)
- [Configuration on page 272](#)
- [Verification on page 273](#)

---

#### Requirements

---

Before you begin:

Complete the tasks such as enabling the chassis cluster, configuring interfaces and redundancy groups. See [“SRX Series Chassis Cluster Configuration Overview” on page 81](#) and [“Example: Configuring Chassis Cluster Redundant Ethernet Interfaces” on page 145](#) for more details.

## Overview

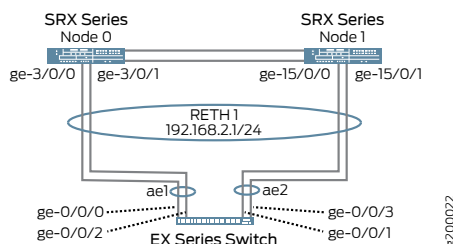
You can combine multiple physical Ethernet ports to form a logical point-to-point link, known as a link aggregation group (LAG) or bundle. You configure LACP on a redundant Ethernet interface of SRX series device in chassis cluster.

In this example, you set the LACP mode for the reth1 interface to active and set the link aggregation control PDU transmit interval to slow, which is every 30 seconds.

When you enable LACP, the local and remote sides of the aggregated Ethernet links exchange protocol data units (PDUs), which contain information about the state of the link. You can configure Ethernet links to actively transmit PDUs, or you can configure the links to passively transmit them (sending out LACP PDUs only when they receive them from another link). One side of the link must be configured as active for the link to be up.

Figure 51 on page 272 shows the topology used in this example.

**Figure 51: Topology for LAGs Connecting SRX Series Devices in Chassis Cluster to an EX Series Switch**



In the Figure 51 on page 272, the ge-3/0/0 interface on SRX Series device is connected to ge-0/0/0 interface on EX Series switch and the ge-15/0/0 interface is connected to ge-0/0/1 on EX Series switch. For more information on EX Series switch configuration, see [Configuring Aggregated Ethernet LACP \(CLI Procedure\)](#).

## 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. Bind redundant child physical interfaces to reth1.

```
[edit interfaces]
```

```
user@host# set interfaces ge-3/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-3/0/1 gigether-options redundant-parent reth1
user@host# set interfaces ge-15/0/0 gigether-options redundant-parent reth1
user@host# set interfaces ge-15/0/1 gigether-options redundant-parent reth1
```

2. Add reth1 to redundancy group 1.

```
[edit interfaces]
```

```
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
```



- Set the LACP on reth1.

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

- Assign an IP address to reth1.

```
[edit interfaces]
user@host# set interfaces reth1 unit 0 family inet address 192.168.2.1/24
```

- 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 reth1** command.

```
user@host> show lacp interfaces reth1
Aggregated interface: reth1
LACP state:      Role  Exp  Def  Dist  Col  Syn  Aggr  Timeout  Activity
ge-15/0/0        Actor  No   No   Yes  Yes  Yes  Yes    Fast    Active
ge-15/0/0        Partner No   No   Yes  Yes  Yes  Yes    Fast    Active
ge-15/0/1        Actor  No   No   Yes  Yes  Yes  Yes    Fast    Active
ge-15/0/1        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
LACP protocol:   Receive State  Transmit State  Mux State
ge-15/0/0        Current      Fast periodic Collecting distributing
ge-15/0/1        Current      Fast periodic Collecting distributing
ge-3/0/0         Current      Fast periodic Collecting distributing
ge-3/0/1         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.

- See Also**
- [Understanding LACP on Chassis Clusters on page 269](#)
  - [Verifying LACP on Redundant Ethernet Interfaces](#)

## Example: Configuring Chassis Cluster Minimum Links

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 274](#)
- [Overview on page 274](#)
- [Configuration on page 274](#)
- [Verification on page 275](#)

---

### Requirements

Before you begin:

- Configure redundant Ethernet interfaces. See [“Example: Configuring Chassis Cluster Redundant Ethernet Interfaces” on page 145](#).
- Understand redundant Ethernet interface link aggregation groups. See [“Example: Configuring Link Aggregation Groups in a Chassis Cluster” on page 263](#).

---

### Overview

When a redundant Ethernet interface has more than two child links, you can set a minimum number of physical links assigned to the interface on the primary node that must be working for the interface to be up. When the number of physical links on the primary node falls below the minimum-links value, the interface will be down even if some links are still working.

In this example, you specify that three child links on the primary node and bound to reth1 (minimum-links value) be working to prevent the interface from going down. For example, in a redundant Ethernet interface LAG configuration in which six interfaces are assigned to reth1, setting the minimum-links value to 3 means that all reth1 child links on the primary node must be working to prevent the interface's status from changing to down.



**NOTE:** Although it is possible to set a minimum-links value for a redundant Ethernet interface with only two child interfaces (one on each node), we do not recommend it.

---

---

### Configuration

#### Step-by-Step Procedure

To specify the minimum number of links:

1. Specify the minimum number of links for the redundant Ethernet interface.

```
{primary:node0}[edit]
user@host# set interfaces reth1 redundant-ether-options minimum-links 3
```

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

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

## Verification

### Verifying the Chassis Cluster Minimum Links Configuration

**Purpose** To verify the configuration is working properly, enter the **show interface reth1** command.

**Action** From operational mode, enter the show **show interfaces reth1** command.

```
{primary:node0}[edit]
user@host> show interfaces reth1
Physical interface: reth1, Enabled, Physical link is Down
  Interface index: 129, SNMP ifIndex: 548
  Link-level type: Ethernet, MTU: 1514, Speed: Unspecified, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Disabled, Minimum links needed: 3, Minimum bandwidth needed: 0
  Device flags   : Present Running
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Current address: 00:10:db:ff:10:01, Hardware address: 00:10:db:ff:10:01
  Last flapped   : 2010-09-15 15:54:53 UTC (1w0d 22:07 ago)
  Input rate      : 0 bps (0 pps)
  Output rate     : 0 bps (0 pps)

Logical interface reth1.0 (Index 68) (SNMP ifIndex 550)
  Flags: Hardware-Down Device-Down SNMP-Traps 0x0 Encapsulation: ENET2
  Statistics          Packets      pps      Bytes      bps
  Bundle:
    Input :              0          0          0          0
    Output:              0          0          0          0
  Security: Zone: untrust
  Allowed host-inbound traffic : bootp bfd bgp dns dvmrp igmp ldp msdp nhrp
  ospf pgm pim rip router-discovery rsvp sap vrrp dhcp finger ftp tftp
  ident-reset http https ike netconf ping reverse-telnet reverse-ssh rlogin
  rpm rsh snmp snmp-trap ssh telnet traceroute xnm-clear-text xnm-ssl lsping
  ntp sip
  Protocol inet, MTU: 1500
  Flags: Sendbroadcast-pkt-to-re
```

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

Support for Ethernet link aggregation groups (LAGs) based on IEEE 802.3ad makes it possible to aggregate physical interfaces on a standalone device. LAGs on standalone devices provide increased interface bandwidth and link availability. Aggregation of links

in a chassis cluster allows a redundant Ethernet interface to add more than two physical child interfaces, thereby creating a redundant Ethernet interface LAG.

- [Requirements on page 276](#)
- [Overview on page 276](#)
- [Configuration on page 276](#)
- [Verification on page 278](#)

---

## Requirements

This example uses the following software and hardware components:

- Junos OS Release 15.1X49-D40 or later for SRX Series devices.
- SRX5800 with IOC2 or IOC3 with Express Path enabled on IOC2 and IOC3. For details, see *Example: Configuring SRX5K-MPC3-100G10G (IOC3) and SRX5K-MPC3-40G10G (IOC3) on an SRX5000 Line Device to Support Express Path*.

---

## Overview

This example shows how to configure a redundant Ethernet interface link aggregation group and configure LACP on chassis clusters on an SRX Series device using the ports from either IOC2 or IOC3 in Express Path mode. Note that configuring child interfaces by mixing links from both IOC2 and IOC3 is not supported.

The following member links are used in this example:

- xe-1/0/0
- xe-3/0/0
- xe-14/0/0
- xe-16/0/0

---

## Configuration

### CLI Quick Configuration

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

```
set chassis cluster reth-count 5
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 redundant-ether-options lacp active
set interfaces reth0 redundant-ether-options lacp periodic fast
set interfaces reth0 redundant-ether-options minimum-links 1
set interfaces reth0 unit 0 family inet address 192.0.2.1/24
set interfaces xe-1/0/0 gigether-options redundant-parent reth0
set interfaces xe-3/0/0 gigether-options redundant-parent reth0
set interfaces xe-14/0/0 gigether-options redundant-parent reth0
set interfaces xe-16/0/0 gigether-options redundant-parent reth0
```

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

To configure LAG Interfaces:

1. Specify the number of aggregated Ethernet interfaces to be created.

```
[edit chassis]
user@host# set chassis cluster reth-count 5
```

2. Bind redundant child physical interfaces to reth0.

```
[edit interfaces]
user@host# set xe-1/0/0 gigether-options redundant-parent reth0
user@host# set xe-3/0/0 gigether-options redundant-parent reth0
user@host# set xe-14/0/0 gigether-options redundant-parent reth0
user@host# set xe-16/0/0 gigether-options redundant-parent reth0
```

3. Add reth0 to redundancy group 1.

```
user@host# set reth0 redundant-ether-options redundancy-group 1
```

4. Assign an IP address to reth0.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.1/24
```

5. Set the LACP on reth0.

```
[edit interfaces]
user@host# set reth0 redundant-ether-options lacp active
user@host# set reth0 redundant-ether-options lacp periodic fast
user@host# set reth0 redundant-ether-options minimum-links 1
```

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

```
[edit]
user@host# show interfaces
xe-1/0/0 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-3/0/0 {
  gigether-options {
    redundant-parent reth0;
  }
}
xe-14/0/0 {
  gigether-options {
```

```
        redundant-parent reth0;
    }
}
xe-16/0/0 {
    gigether-options {
        redundant-parent reth0;
    }
}
reth0 {
    redundant-ether-options {
        lacp {
            active;
            periodic fast;
        }
        minimum-links 1;
    }
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
ae1 {
    aggregated-ether-options {
        lacp {
            active;
        }
    }
    unit 0 {
        family inet {
            address 192.0.2.2/24;
        }
    }
}
[edit]
user@host# show chassis
chassis cluster {
    reth-count 5;
}
```

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

---

## Verification

### *Verifying LACP on Redundant Ethernet Interfaces*

**Purpose** Display LACP status information for redundant Ethernet interfaces.

**Action** From operational mode, enter the **show lacp interfaces** command to check that LACP has been enabled as active on one end.

```
user@host> show lacp interfaces
```

```
Aggregated interface: reth0
```

LACP state:	Role	Exp	Def	Dist	Co1	Syn	Aggr	Timeout	Activity
xe-16/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-16/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-14/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-14/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-1/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-1/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-3/0/0	Actor	No	No	Yes	Yes	Yes	Yes	Fast	Active
xe-3/0/0	Partner	No	No	Yes	Yes	Yes	Yes	Fast	Active
LACP protocol:	Receive State		Transmit State				Mux State		
xe-16/0/0	Current		Fast periodic				Collecting distributing		
xe-14/0/0	Current		Fast periodic				Collecting distributing		
xe-1/0/0	Current		Slow periodic				Collecting distributing		
xe-3/0/0	Current		Slow periodic				Collecting distributing		

The output indicates that LACP has been set up correctly and is active at one end.

## Understanding VRRP on SRX Series Devices

SRX Series devices support the Virtual Router Redundancy Protocol (VRRP) and VRRP for IPv6. This topic covers:

- [Overview of VRRP on SRX Series Devices on page 279](#)
- [Benefits of VRRP on page 280](#)
- [Sample VRRP Topology on page 280](#)
- [SRX Series Devices Support for VRRPv3 on page 281](#)
- [Limitations of VRRPv3 Features on page 281](#)

### Overview of VRRP on SRX Series Devices

Configuring end hosts on your network with static default routes minimizes configuration effort and complexity and reduces processing overhead on the end hosts. When hosts are configured with static routes, the failure of the default gateway normally results in a catastrophic event, isolating all hosts that are unable to detect available alternate paths to their gateway. Using Virtual Router Redundancy Protocol (VRRP) enables you to dynamically provide alternative gateways for end hosts if the primary gateway fails.

You can configure the Virtual Router Redundancy Protocol (VRRP) or VRRP for IPv6 on Gigabit Ethernet interfaces, 10-Gigabit Ethernet interfaces, and logical interfaces on SRX Series devices. VRRP enables hosts on a LAN to make use of redundant devices on that LAN without requiring more than the static configuration of a single default route on the hosts. Devices configured with VRRP share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP configured devices is the master (active) and the others are backups. If the master device fails, then one of the backup devices becomes the new master, providing a virtual default device and enabling traffic on the LAN to be routed without relying on a single device. Using VRRP, a backup SRX Series device can take over a failed default device within a few seconds. This is done with minimum loss of VRRP traffic and without any interaction with the hosts. Virtual Router Redundancy Protocol is not supported on management interfaces.

VRRP for IPv6 provides a much faster switchover to an alternate default device than IPv6 Neighbor Discovery (ND) procedures. VRRP for IPv6 does not support the **authentication-type** or **authentication-key** statements.

Devices running VRRP dynamically elect master and backup devices. You can also force assignment of master and backup devices using priorities from 1 through 255, with 255 being the highest priority. In VRRP operation, the default master device sends advertisements to the backup device at a regular intervals. The default interval is 1 second. If the backup device do not receive an advertisement for a set period, then the backup device with the highest priority takes over as master and begins forwarding packets.

The backup devices do not attempt to preempt the master device unless it has higher priority. This eliminates service disruption unless a more preferred path becomes available. It is possible to administratively prohibit all preemption attempts, with the exception of a VRRP device becoming master device of any device associated with addresses it owns.



**NOTE:** Priority 255 cannot be set for routed VLAN interfaces (RVIs).

VRRP is defined in RFC 3768, *Virtual Router Redundancy Protocol*.

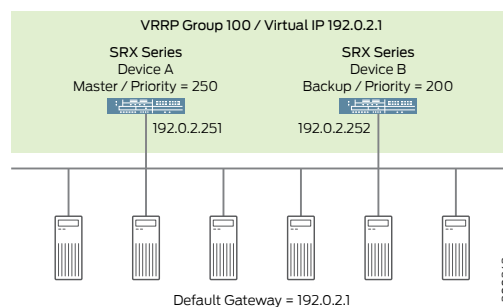
### Benefits of VRRP

- VRRP provides dynamic failover of IP addresses from one device to another in the event of failure.
- You can implement VRRP to provide a highly available default path to a gateway without needing to configure dynamic routing or router discovery protocols on end hosts.

### Sample VRRP Topology

Figure 52 on page 280 illustrates a basic VRRP topology with SRX Series devices. In this example, Devices A and B are running VRRP and share the virtual IP address 192.0.2.1. The default gateway for each of the clients is 192.0.2.1.

**Figure 52: Basic VRRP on SRX Series Switches**





The following illustrates basic VRRP behavior using [Figure 52 on page 280](#) for reference:

1. When any of the servers wants to send traffic out of the LAN, it sends the traffic to the default gateway address of 192.0.2.1. This is a virtual IP address (VIP) owned by VRRP group 100. Because Device A is the master of the group, the VIP is associated with the “real” address 192.0.2.251 on Device A, and traffic from the servers is actually sent to this address. (Device A is the master because it has been configured with a higher priority value.)
2. If there is a failure on Device A that prevents it from forwarding traffic to or from the servers—for example, if the interface connected to the LAN fails—Device B becomes the master and assumes ownership of the VIP. The servers continue to send traffic to the VIP, but because the VIP is now associated with the “real” address 192.0.2.252 on Device B (because of change of master), the traffic is sent to Device B instead of Device A.
3. If the problem that caused the failure on Device A is corrected, Device A becomes the master again and reasserts ownership of the VIP. In this case, the servers resume sending traffic to Device A.

Notice that no configuration changes are required on the servers for them to switch between sending traffic to Device A and Device B. When the VIP moves between 192.0.2.251 and 192.0.2.252, the change is detected by normal TCP-IP behavior and no configuration or intervention is required on the servers.

---

### SRX Series Devices Support for VRRPv3

The advantage of using VRRPv3 is that VRRPv3 supports both IPv4 and IPv6 address families, whereas VRRP supports only IPv4 addresses.

Enable VRRPv3 in your network only if VRRPv3 can be enabled on all the devices configured with VRRP in your network because VRRPv3 (IPv4) does not interoperate with the previous versions of VRRP. For example, if VRRP IPv4 advertisement packets are received by a device on which VRRPv3 is enabled, then the device transitions itself to the backup state to avoid creating multiple masters in the network.

You can enable VRRPv3 by configuring the version-3 statement at the **[edit protocols vrrp]** hierarchy level (for IPv4 or IPv6 networks). Configure the same protocol version on all VRRP devices on the LAN.

---

### Limitations of VRRPv3 Features

Below are some VRRPv3 features limitations.

#### **VRRPv3 Authentication**

When VRRPv3 (for IPv4) is enabled, it does not allow authentication.

- The **authentication-type** and **authentication-key** statements cannot be configured for any VRRP groups.
- You must use non-VRRP authentication.

### **VRRPv3 Advertisement Intervals**

VRRPv3 (for IPv4 and IPv6) advertisement intervals must be set with the `fast-interval` statement at the `[edit interfaces interface-name unit 0 family inet address ip-address vrrp-group group-name]` hierarchy level.

- Do not use the `advertise-interval` statement (for IPv4).
- Do not use the `inet6-advertise-interval` statement (for IPv6).

- See Also**
- For more information on VRRP or VRRP for IPv6, see the [Junos OS High Availability Configuration Guide](#).
  - [show vrrp on page 613](#)

## **Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces**

When Virtual Router Redundancy Protocol (VRRP) is configured, the VRRP groups multiple devices into a virtual device. At any time, one of the devices configured with VRRP is the master (active) and the other devices are backups. If the master fails, one of the backup devices becomes the new master device.

This example describes how to configure VRRP on redundant interface:

- [Requirements on page 282](#)
- [Overview on page 282](#)
- [Configuration VRRP on page 283](#)
- [Verification on page 289](#)

---

### **Requirements**

This example uses the following hardware and software components:

- Junos OS Release 18.1 R1 or later for SRX Series Services Gateways.
- Two SRX Series devices connected in a chassis cluster.
- One SRX Series device connected as standalone device.

---

### **Overview**

You configure VRRP by configuring VRRP groups on redundant interfaces on a chassis cluster devices and on Gigabit Ethernet interface on standalone device. A redundant interface of chassis cluster devices and Gigabit Ethernet interface of standalone device can be a member of one or more VRRP groups. Within a VRRP group, the master redundant interface of chassis cluster devices and the backup Gigabit Ethernet interface of standalone device must be configured.

To configure VRRP group, you must configure group identifier, and virtual IP address to the redundant interfaces and Gigabit Ethernet interfaces that are members of VRRP group. The virtual IP address must be the same for all the interfaces in the VRRP group.

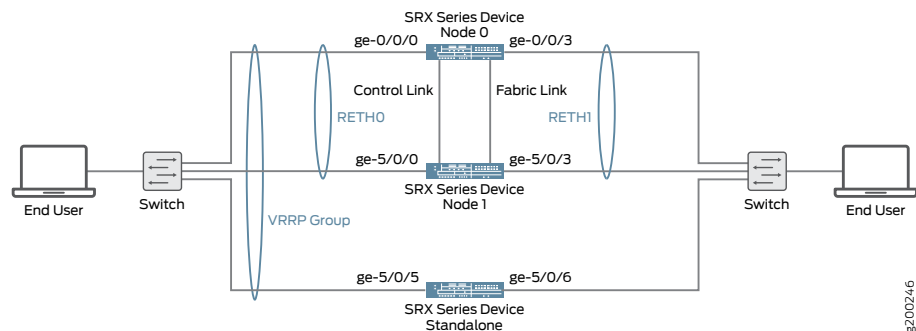
Then you configure the priority to the redundant interfaces and Gigabit Ethernet interfaces to become the master interface.

You can force assignment of master and backup redundant interfaces and Gigabit Ethernet interfaces using priorities from 1 through 255, where 255 is the highest priority.

### Topology

Figure 53 on page 283 shows the topology used in this example.

Figure 53: VRRP on Redundant interface



### Configuration VRRP

- [Configuring VRRPv3, VRRP Groups, and Priority on Chassis Cluster Redundant Ethernet Interfaces on page 283](#)
- [Configuring VRRP Groups on Standalone Device on page 287](#)

#### Configuring VRRPv3, VRRP Groups, and Priority on Chassis Cluster Redundant Ethernet Interfaces

##### CLI Quick Configuration

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

```
set protocols vrrp traceoptions file vrrp.log
set protocols vrrp traceoptions file size 10000000
set protocols vrrp traceoptions flag all
set protocols vrrp version-3
set protocols vrrp ignore-nonstop-routing
set interfaces ge-0/0/0 gigether-options redundant-parent reth0
set interfaces ge-0/0/3 gigether-options redundant-parent reth1
set interfaces ge-5/0/0 gigether-options redundant-parent reth0
set interfaces ge-5/0/3 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 virtual-address 192.0.2.3
set interfaces reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 priority 255
set interfaces reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 accept-data
set interfaces reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 virtual-inet6-address 2001:db8::3
```

```
set interfaces reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2 priority
255
set interfaces reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group 2
accept-data
set interfaces reth1 redundant-ether-options redundancy-group 2
set interfaces reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 virtual-address
192.168.120.3
set interfaces reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 priority 150
set interfaces reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 accept-data
set interfaces reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3
virtual-inet6-address 2001:db8::4
set interfaces reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3 priority
150
set interfaces reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group 3
accept-data
```

**Step-by-Step  
Procedure**

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 VRRPv3, VRRP Groups, and priority on chassis cluster devices:

1. Configure a filename to the traceoptions to trace VRRP protocol traffic.  

```
[edit protocols vrrp]
user@host# set traceoptions file vrrp.log
```
2. Specify the maximum trace file size.  

```
[edit protocols vrrp]
user@host# set traceoptions file size 10000000
```
3. Enable vrrp traceoptions.  

```
[edit protocols vrrp]
user@host# set traceoptions flag all
```
4. Set vrrp version to 3.  

```
[edit protocols vrrp]
user@host# set version-3
```
5. Set up the redundant Ethernet (reth) interfaces and assign the redundant interface to a zone.  

```
[edit interfaces]
user@host# set ge-0/0/0 gigether-options redundant-parent reth0
user@host# set ge-0/0/3 gigether-options redundant-parent reth1
user@host# set ge-5/0/0 gigether-options redundant-parent reth0
user@host# set ge-5/0/3 gigether-options redundant-parent reth1
user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth1 redundant-ether-options redundancy-group 2
```

6. Configure the family inet address and virtual address for the redundant interface 0 unit 0.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0
virtual-address 192.168.110.3
user@host# set reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group
2 virtual-inet6-address 2001:db8::3
```

7. Configure the family inet address and virtual address for the redundant interface 1 unit 0.

```
[edit interfaces]
user@host# set reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1
virtual-address 192.168.120.3
user@host# set reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group
3 virtual-inet6-address 2001:db8::4
```

8. Set the priority of the redundant interface 0 unit 0 to 255.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0 priority
255
user@host# set reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group
2 priority 255
```

9. Set the priority of the redundant interface 1 unit 0 to 150.

```
[edit interfaces]
user@host# set reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 priority
150
user@host# set reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group
3 priority 150
```

10. Configure the redundant interface 0 unit 0 to accept all packets sent to the virtual IP address.

```
[edit interfaces]
user@host# set reth0 unit 0 family inet address 192.0.2.2/24 vrrp-group 0
accept-data
user@host# set reth0 unit 0 family inet6 address 2001:db8::2/32 vrrp-inet6-group
2 accept-data
```

11. Configure the redundant interface 1 unit 0 to accept all packets sent to the virtual IP address.

```
[edit interfaces]
user@host# set reth1 unit 0 family inet address 192.0.2.4/24 vrrp-group 1 accept-data
user@host# set reth1 unit 0 family inet6 address 2001:db8::3/32 vrrp-inet6-group
3 accept-data
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces reth0** and **show interfaces reth1** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces reth0
redundant-ether-options {
    redundancy-group 1;
}
unit 0 {
    family inet {
        address 192.0.2.2/24 {
            vrrp-group 0 {
                virtual-address 192.0.2.3;
                priority 255;
                accept-data;
            }
        }
    }
    family inet6 {
        address 2001:db8::2/32 {
            vrrp-inet6-group 2 {
                virtual-inet6-address 2001:db8::3;
                priority 255;
                accept-data;
            }
        }
    }
}

[edit]
user@host# show interfaces reth1
redundant-ether-options {
    redundancy-group 2;
}
unit 0 {
    family inet {
        address 192.0.2.4/24 {
            vrrp-group 1 {
                virtual-address 192.0.2.5;
                priority 150;
                accept-data;
            }
        }
    }
    family inet6 {
        address 2001:db8::3/32 {
            vrrp-inet6-group 3 {
                virtual-inet6-address 2001:db8::4;
                priority 150;
                accept-data;
            }
        }
    }
}
```

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

### *Configuring VRRP Groups on Standalone Device*

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

```
set protocols vrrp version-3
set interfaces xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 virtual-address 192.0.2.3
set interfaces xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 priority 50
set interfaces xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 accept-data
set interfaces xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 virtual-inet6-address 2001:db8::3
set interfaces xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 priority 50
set interfaces xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 accept-data
set interfaces xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 virtual-address 192.0.2.5
set interfaces xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 priority 50
set interfaces xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 accept-data
set interfaces xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 virtual-inet6-address 2001:db8::4
set interfaces xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 priority 50
set interfaces xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 accept-data
```

**Step-by-Step Procedure** 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 VRRP groups on standalone device:

1. Set vrrp version to 3.

```
[edit protocols vrrp]
user@host# set version-3
```

2. Configure the family inet address and virtual address for the Gigabit Ethernet interface unit 0.

```
[edit interfaces]
user@host# set xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 virtual-address 192.0.2.3
user@host# set xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group 2 virtual-inet6-address 2001:db8::3
user@host# set xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 virtual-address 192.0.2.5
user@host# set xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32 vrrp-inet6-group 3 virtual-inet6-address 2001:db8::4
```

3. Set the priority of the Gigabit Ethernet interface unit 0 to 50.

```
[edit interfaces]
user@host# set xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0 priority
50
user@host# set xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group
2 priority 50
user@host# set xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1 priority
50
user@host# set xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32
vrrp-inet6-group 3 priority 50
```

4. Configure the Gigabit Ethernet interface unit 0 to accept all packets sent to the virtual IP address.

```
[edit interfaces]
user@host# set xe-5/0/5 unit 0 family inet address 192.0.2.1/24 vrrp-group 0
accept-data
user@host# set xe-5/0/5 unit 0 family inet6 address 2001:db8::1/32 vrrp-inet6-group
2 accept-data
user@host# set xe-5/0/6 unit 0 family inet address 192.0.2.1/24 vrrp-group 1
accept-data
user@host# set xe-5/0/6 unit 0 family inet6 address 2001:db8::5/32
vrrp-inet6-group 3 accept-data
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces xe-5/0/5** and **show interfaces xe-5/0/6** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces xe-5/0/5
unit 0 {
  family inet {
    address 192.0.2.1/24 {
      vrrp-group 0 {
        virtual-address 192.0.2.3;
        priority 50;
        accept-data;
      }
    }
  }
  family inet6 {
    address 2001:db8::1/32 {
      vrrp-inet6-group 2 {
        virtual-inet6-address 2001:db8::3;
        priority 50;
        accept-data;
      }
    }
  }
}
[edit]
```



```

user@host# show interfaces xe-5/0/6
unit 0 {
  family inet {
    address 192.0.2.1/24 {
      vrrp-group 1 {
        virtual-address 192.0.2.5;
        priority 50;
        accept-data;
      }
    }
  }
  family inet6 {
    address 2001:db8::5/32 {
      vrrp-inet6-group 3 {
        virtual-inet6-address 2001:db8::4;
        priority 50;
        accept-data;
      }
    }
  }
}

```

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

### Verification

Confirm that the configuration is working properly.

- [Verifying the VRRP on Chassis Cluster Devices on page 289](#)
- [Verifying the VRRP on standalone device on page 290](#)

#### *Verifying the VRRP on Chassis Cluster Devices*

**Purpose** Verify that VRRP on chassis cluster devices has been configured properly.

**Action** From operational mode, enter the **show vrrp brief** command to display the status of VRRP on chassis cluster devices.

```

user@host> show vrrp brief

```

Interface	State	Group	VR state	VR Mode	Timer	Type	Address
reth0.0	up	0	master	Active	A 0.149	1c1	192.0.2.3
						vip	192.0.2.3
reth0.0	up	2	master	Active	A 0.155	1c1	2001:db8::2
						vip	2001:db8:5eff:fe00:202
reth1.0	up	1	master	Active	A 0.445	1c1	2001:db8::2
						vip	192.0.2.4
reth1.0	up	3	master	Active	A 0.414	1c1	2001:db8::4
						vip	192.0.2.4
						vip	2001:db8:5eff:fe00:203
						vip	2001:db8::4

**Meaning** The sample output shows that the four VRRP groups are active and that the redundant interfaces has assumed the correct master roles. The lcl address is the physical address of the interface and the vip address is the virtual address shared by redundant interfaces. The Timer value (A 0.149, A 0.155, A 0.445, and A 0.414) indicates the remaining time (in seconds) in which the redundant interfaces expects to receive a VRRP advertisement from the Gigabit Ethernet interfaces. If an advertisement for group 0, 1, 2, and 3 does not arrive before the timer expires, Chassis cluster devices asserts itself as the master.

### *Verifying the VRRP on standalone device*

**Purpose** Verify that VRRP has been configured properly on a standalone device.

**Action** From operational mode, enter the **show vrrp brief** command to display the status of VRRP on standalone device.

```
user@host> show vrrp brief
Interface      State      Group  VR state VR Mode  Timer  Type  Address
xe-5/0/5.0     up         0      backup  Active  D 3.093 1cl  192.0.2.2.1
                                     vip  192.0.2.2
                                     mas  192.0.2.2.2
xe-5/0/5.0     up         2      backup  Active  D 3.502 1cl  2001:db8::2:1
                                     vip  2001:db8:200:5eff:fe00:202
                                     vip  2001:db8::2
                                     mas
2001:db8:210:dbff:feff:1000
xe-5/0/6.0     up         1      backup  Active  D 3.499 1cl  192.0.2.5.1
                                     vip  192.0.2.5
                                     mas  192.0.2.5.2
xe-5/0/6.0     up         3      backup  Active  D 3.282 1cl  2001:db8::5
                                     vip
2001:db8:200:5eff:fe00:203
                                     vip  2001:db8::4
                                     mas
2001:db8:210:dbff:feff:1001
```

**Meaning** The sample output shows that the four VRRP groups are active and that the Gigabit Ethernet interfaces has assumed the correct backup roles. The lcl address is the physical address of the interface and the vip address is the virtual address shared by Gigabit Ethernet interfaces. The Timer value (D 3.093, D 3.502, D 3.499, and D 3.282) indicates the remaining time (in seconds) in which the Gigabit Ethernet interfaces expects to receive a VRRP advertisement from the redundant interfaces. If an advertisement for

group 0, 1, 2, and 3 does not arrive before the timer expires, then the standalone device continues to be a backup device.

- See Also**
- [authentication-type on page 420](#)
  - [authentication-key on page 419](#)
  - [show vrrp on page 613](#)

- Related Documentation**
- [Chassis Cluster Redundant Ethernet Interfaces on page 140](#)
  - [Configuring Chassis Clustering on an SRX Series Devices on page 150](#)

## NTP Time Synchronization on Chassis Cluster

**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. For more information, see the following topics:

- [NTP Time Synchronization on SRX Series Devices on page 291](#)
- [Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 292](#)

## NTP Time Synchronization on SRX Series Devices

In both standalone and chassis cluster modes, the primary Routing Engine runs the NTP process to get the time from the external NTP server. Although the secondary Routing Engine runs the NTP process in an attempt to get the time from the external NTP server, this attempt fails because of network issues. For this reason, the secondary Routing Engine uses NTP to get the time from the primary Routing Engine.

Use NTP to:

- Send the time from the primary Routing Engine to the secondary Routing Engine through the chassis cluster control link.
- Get the time from an external NTP server to the primary or a standalone Routing Engine.
- Get the time from the Routing Engine NTP process to the Packet Forwarding Engine.



**NOTE:** On SRX Series devices, use the command `set system processes ntpd-service` to configure NTP.

Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, configuring the NTP time adjustment threshold is supported on SRX300, SRX320, SRX340, SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances. This feature allows you to configure and enforce the NTP adjustment threshold

for the NTP service and helps in improve the security and flexibility of the NTP service protocol.

- See Also**
- [NTP Overview](#)
  - [ntp threshold on page 457](#)
  - [show system ntp threshold on page 598](#)
  - [set date ntp on page 582](#)

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

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:

- Real-time objects (RTO)
- Licenses
- Software updates
- Node failovers
- Analyzing system logs (syslogs)
- [Requirements on page 292](#)
- [Overview on page 292](#)
- [Configuration on page 293](#)
- [Verification on page 294](#)

#### 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

---

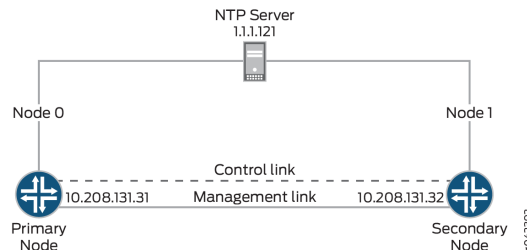
When SRX Series devices are operating in chassis cluster mode, the secondary node cannot access the external NTP server through the revenue port. Junos OS Release 12.1X47

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 54 on page 293 shows the time synchronization from the peer node using the control link.

Figure 54: 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 293](#)
- [Results on page 294](#)

#### 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 294](#)
- [Verifying the NTP Configuration on the Secondary Node on page 296](#)

### 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.

Release History Table

Release	Description
15.1X49-D70	Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, configuring the NTP time adjustment threshold is supported on SRX300, SRX320, SRX340, SRX345, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, and SRX5800 devices and vSRX instances. This feature allows you to configure and enforce the NTP adjustment threshold for the NTP service and helps in improve the security and flexibility of the NTP service protocol.

**Related  
Documentation**

- [Time Management Routing Guide for Administration Devices](#)
- [Verifying Chassis Cluster Configuration Synchronization Status on page 167](#)

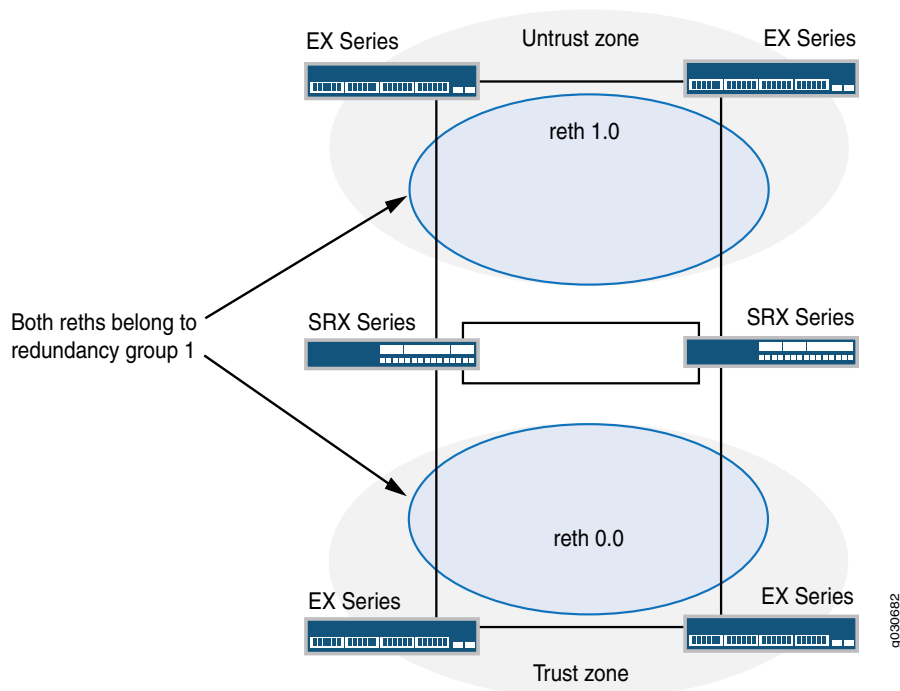
## Active/Passive Chassis Cluster Deployments

- [Understanding Active/Passive Chassis Cluster Deployment on page 298](#)
- [Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Devices on page 299](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\) on page 316](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair \(J-Web\) on page 328](#)
- [Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 329](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel on page 330](#)
- [Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel \(J-Web\) on page 346](#)

### Understanding Active/Passive Chassis Cluster Deployment

In this case, a single device in the cluster is used to route all traffic while the other device is used only in the event of a failure (see [Figure 55 on page 298](#)). When a failure occurs, the backup device becomes master and controls all forwarding.

*Figure 55: Active/Passive Chassis Cluster Scenario*



An active/passive chassis cluster can be achieved by using redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. If any of the interfaces in an active group in a node fails, the group is declared inactive and all the interfaces in the group fail over to the other node.

This configuration minimizes the traffic over the fabric link because only one node in the cluster forwards traffic at any given time.

**See Also** • [Chassis Cluster Overview on page 25](#)

## Example: Configuring an Active/Passive Chassis Cluster on SRX5800 Devices

This example shows how to set up basic active/passive chassis clustering on an SRX5800 devices. In this example, you also enable secure login and to prevent attackers from gaining privileged access through this control port by configuring the internal IP security (IPsec) security association (SA).

- [Requirements on page 299](#)
- [Overview on page 299](#)
- [Configuration on page 302](#)
- [Verification on page 312](#)

### Requirements

Before you begin:

- You need two SRX5800 Services Gateways with identical hardware configurations, one MX240 edge router, and one EX8208 Ethernet Switch.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models.
- Before the cluster is formed, you must configure control ports for each device, as well as assign a cluster ID and node ID to each device, and then reboot. When the system boots, both the nodes come up as a cluster.



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

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

### Overview

This example shows how to set up basic active/passive chassis clustering on an SRX Series device. The basic active/passive example is the most common type of chassis cluster.

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

- One device actively provides routing, firewall, NAT, VPN, and security services, along with maintaining control of the chassis cluster.

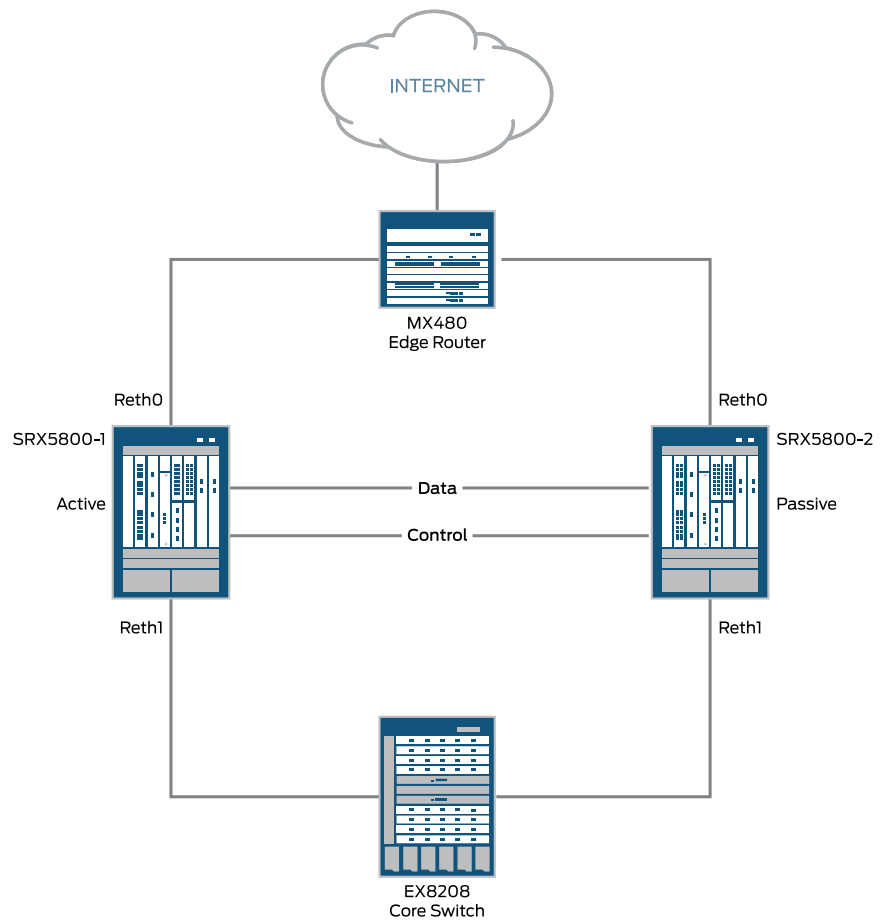
- The other device passively maintains its state for cluster failover capabilities in case the active device becomes inactive.



**NOTE:** This active/passive mode example for the SRX5800 Services Gateway does not describe in detail miscellaneous configurations such as how to configure NAT, security policies, or VPNs. They are essentially the same as they would be for standalone configurations. See *Introduction to NAT*, *Security Policies Overview*, and *IPsec VPN Overview*. However, if you are performing proxy ARP in chassis cluster configurations, you must apply the proxy ARP configurations to the reth interfaces rather than the member interfaces because the RETH interfaces hold the logical configurations. See *Configuring Proxy ARP for NAT (CLI Procedure)*. You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800 Services Gateway. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

Figure 56 on page 301 shows the topology used in this example.

*Figure 56: Basic Active/Passive Chassis Clustering on an SRX Series Device Topology Example*



8031173

## Configuration

---

### *Configuring Internal IPsec Security Association (SA)*

**Step-by-Step Procedure** On SRX5400, SRX5600, and SRX5800 devices have a chassis cluster control port that is used to connect two SRX Series devices to form a chassis cluster. To ensure secure login and to prevent attackers from gaining privileged access through this control port, an internal IPsec SA is installed. Besides using internal IPsec to secure RSH and RCP between the primary and backup Routing Engines, the internal IPsec SA is installed on all the Services Processing Units (SPUs). An attacker cannot access any of the RSH services without knowing the internal IPsec key.

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

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



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

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

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

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

1. Enable the **iked**-encryption:  

```
user@host# set security ipsec internal security-association manual encryption  
ike-ha-link-encryption enable
```
2. Enable the 3des-cbc encryption algorithm:

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

3. Configure the encryption key:

```
user@host# prompt security ipsec internal security-association manual encryption
key ascii-text
```



**NOTE:** The password must be of 24 characters long.

4. Activate internal IPsec:

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

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

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

Control interfaces:

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

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

Select FPC 1/13, because the central point is always on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the central point (for this example, use the SPC in slot 1). You must enter the operational mode commands on both devices. For example:

- On node 0:

```
user@host# set chassis cluster control-ports fpc 1 port 0
user@host# set chassis cluster control-ports fpc 13 port 0
user@host# commit
```

- On node 1:

```
user@host# set chassis cluster control-ports fpc 1 port 0
user@host# set chassis cluster control-ports fpc 13 port 0
user@host# commit
```

7. Set the two devices to cluster mode. A reboot is required to enter into cluster mode after the cluster ID and node ID are set. You can cause the system to boot automatically by including the **reboot** parameter in the CLI command line. You must enter the operational mode commands on both devices. For example:

- On node 0:

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

- On node 1:

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

The cluster ID is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 1 through 255. Setting a cluster ID to 0 is equivalent to disabling a cluster. Cluster ID greater than 15 can only be set when the fabric and control link interfaces are connected back-to-back.

### *Configuring a Chassis Cluster in Active/Passive Mode*

#### **CLI Quick Configuration**

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

On {primary:node0}

```
[edit]
set interfaces fab0 fabric-options member-interfaces ge-11/3/0
set interfaces fab1 fabric-options member-interfaces ge-23/3/0
set groups node0 system host-name SRX5800-1
set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24
set groups node0 system backup-router 10.3.5.254 destination 10.0.0.0/16
set groups node1 system host-name SRX5800-2
set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24
set groups node1 system backup-router 10.3.5.254 destination 10.0.0.0/16
set apply-groups "${node}"
set chassis cluster reth-count 2
set chassis cluster redundancy-group 0 node 0 priority 129
set chassis cluster redundancy-group 0 node 1 priority 128
set chassis cluster redundancy-group 1 node 0 priority 129
set chassis cluster redundancy-group 1 node 1 priority 128
set interfaces xe-6/0/0 gigether-options redundant-parent reth0
set interfaces xe-6/1/0 gigether-options redundant-parent reth1
set interfaces xe-18/0/0 gigether-options redundant-parent reth0
set interfaces xe-18/1/0 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 1.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 2.2.2.1/24
set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
set chassis cluster control-link-recovery
set security zones security-zone untrust interfaces reth0.0
set security zones security-zone trust interfaces reth1.0
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
set routing-options static route 2.0.0.0/8 next-hop 2.2.2.254
```

To quickly configure an EX8208 Core Switch, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your



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

On {primary:node0}

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

To quickly configure an MX240 edge router, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

On {primary:node0}

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

#### Step-by-Step Procedure

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

To configure a chassis cluster on an SRX Series device:



**NOTE:** In cluster mode, the cluster is synchronized between the nodes when you execute a **commit** command. All commands are applied to both nodes regardless of from which device the command is configured.

1. Configure the fabric (data) ports of the cluster that are used to pass RTOs in active/passive mode. For this example, use one of the 1-Gigabit Ethernet ports because running out of bandwidth using active/passive mode is not an issue. Define two fabric interfaces, one on each chassis, to connect together.

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

2. Because the SRX5800 Services Gateway chassis cluster configuration is contained within a single common configuration, to assign some elements of the configuration to a specific member only, you must use the Junos OS node-specific configuration method called groups. The **set apply-groups \${node}** command uses the node variable to define how the groups are applied to the nodes; each node recognizes its number and accepts the configuration accordingly. You must also configure out-of-band management on the fxp0 interface of the SRX5800 Services Gateway using separate IP addresses for the individual control planes of the cluster.



**NOTE:** Configuring the backup router destination address as x.x.x.0/0 is not allowed.

```
{primary:node0}[edit]
user@host# set groups node0 system host-name SRX5800-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24
user@host# set groups node0 system backup-router 10.3.5.254 destination
0.0.0.0/16
user@host# set groups node1 system host-name SRX5800-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24
user@host# set groups node1 system backup-router 10.3.5.254 destination
0.0.0.0/16
user@host# set apply-groups "${node}"
```

3. Configure redundancy groups for chassis clustering. Each node has interfaces in a redundancy group where interfaces are active in active redundancy groups (multiple active interfaces can exist in one redundancy group). Redundancy group 0 controls the control plane and redundancy group 1+ controls the data plane and includes the data plane ports. For this active/passive mode example, only one chassis cluster member is active at a time so you need to define redundancy groups 0 and 1 only. Besides redundancy groups, you must also define:
  - Redundant Ethernet groups—Configure how many redundant Ethernet interfaces (member links) will be active on the device so that the system can allocate the appropriate resources for it.
  - Priority for control plane and data plane—Define which device has priority (for chassis cluster, high priority is preferred) for the control plane, and which device is preferred to be active for the data plane.

**NOTE:**

- In active/passive or active/active mode, the control plane (redundancy group 0) can be active on a chassis different from the data plane (redundancy group 1+ and groups) chassis. However, for this example we recommend having both the control and data plane active on the same chassis member. When traffic passes through the fabric link to go to another member node, latency is introduced (z line mode traffic).
- On SRX Series devices (SRX5000 line), the IPsec VPN is not supported in active/active chassis cluster configuration (that is, when there are multiple RG1+ redundancy groups).

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
user@host# set chassis cluster redundancy-group 0 node 0 priority 129
user@host# set chassis cluster redundancy-group 0 node 1 priority 128
user@host# set chassis cluster redundancy-group 1 node 0 priority 129
user@host# set chassis cluster redundancy-group 1 node 1 priority 128
```

4. Configure the data interfaces on the platform so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly. Seamless transition to a new active node will occur with data plane failover. In case of control plane failover, all the daemons are restarted on the new node thus enabling a graceful restart to avoid losing neighborship with peers (ospf, bgp). This promotes a seamless transition to the new node without any packet loss.

You must define the following items:

- Define the membership information of the member interfaces to the reth interface.
- Define which redundancy group the reth interface is a member of. For this active/passive example, it is always 1.
- Define reth interface information such as the IP address of the interface.

```
{primary:node0}[edit]
user@host# set interfaces xe-6/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-6/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-18/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-18/1/0 gigether-options redundant-parent reth1
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 1.1.1/24
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 2.2.2/24
```

5. Configure the chassis cluster behavior in case of a failure. For the SRX5800 Services Gateway, the failover threshold is set at 255. You can alter the weights to determine the impact on the chassis failover. You must also configure control link recovery.

The recovery automatically causes the secondary node to reboot should the control link fail, and then come back online. Enter these commands on node 0.

```
{primary:node0}[edit]
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0
weight 255
user@host# set chassis cluster control-link-recovery
```

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

6. Configure and connect the reth interfaces to the appropriate zones and virtual routers. For this example, leave the reth0 and reth1 interfaces in the default virtual router inet.0, which does not require any additional configuration.

```
{primary:node0}[edit]
user@host# set security zones security-zone untrust interfaces reth0.0
user@host# set security zones security-zone trust interfaces reth1.0
```

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

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

8. For the EX8208 Ethernet Switch, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably the VLANs, routing, and interface configuration.

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

9. For the MX240 edge router, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably you must use an IRB interface within a virtual switch instance on the switch.

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

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

```
> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX58001;
      backup-router 10.3.5.254 destination 0.0.0.0/16;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 10.3.5.1/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name SRX58002;
      backup-router 10.3.5.254 destination 0.0.0.0/16;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 10.3.5.2/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
system {
  root-authentication {
    encrypted-password "$ABC1234EFGH5678IJKL9101";
```

```
    }
    name-server {
        4.2.2.2;
    }
    services {
        ssh {
            root-login allow;
        }
        netconf {
            ssh;
        }
        web-management {
            http {
                interface fxp0.0;
            }
        }
    }
}
chassis {
    cluster {
        control-link-recovery;
        reth-count 2;
        control-ports {
            fpc 1 port 0;
            fpc 13 port 0;
        }
        redundancy-group 0 {
            node 0 priority 129;
            node 1 priority 128;
        }
        redundancy-group 1 {
            node 0 priority 129;
            node 1 priority 128;
            interface-monitor {
                xe-6/0/0 weight 255;
                xe-6/1/0 weight 255;
                xe-18/0/0 weight 255;
                xe-18/1/0 weight 255;
            }
        }
    }
}
interfaces {
    xe-6/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-6/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    xe-18/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-18/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
}
```

```

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

```

```

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

```

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

### Verification

Confirm that the configuration is working properly.

- [Verifying Chassis Cluster Status on page 312](#)
- [Verifying Chassis Cluster Interfaces on page 313](#)
- [Verifying Chassis Cluster Statistics on page 313](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 314](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 314](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 315](#)
- [Troubleshooting with Logs on page 315](#)

#### *Verifying Chassis Cluster Status*

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

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

```

{primary:node0}
show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

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

Redundancy group: 1 , Failover count: 1

```



node0	129	primary	no	no
node1	128	secondary	no	no

### Verifying Chassis Cluster Interfaces

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

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

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

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

Interface Monitoring:
  Interface      Weight      Status      Redundancy-group
  xe-6/0/0       255        Up          1
  xe-6/1/0       255        Up          1
  xe-18/0/0      255        Up          1
  xe-18/1/0      255        Up          1
```

### Verifying Chassis Cluster Statistics

**Purpose** Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

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

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

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
Services Synchronized:
  Service name      RTOs sent      RTOs received
  Translation context 0                0
  Incoming NAT       0                0
  Resource manager   6                0
  Session create     161             0
  Session close      148             0
  Session change     0                0
  Gate create        0                0
```

Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### *Verifying Chassis Cluster Control Plane Statistics*

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

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

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

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

Services Synchronized:
  Service name          RTOs sent  RTOs received
  Translation context    0           0
  Incoming NAT           0           0
  Resource manager       6           0
  Session create        161         0
  Session close         148         0
  Session change         0           0
  Gate create            0           0
```

Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### Verifying Chassis Cluster Redundancy Group Status

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

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

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

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

### Troubleshooting with Logs

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

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

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

**See Also**

- [Preparing Your Equipment for Chassis Cluster Formation on page 99](#)
- [Connecting SRX Series Devices to Create a Chassis Cluster on page 102](#)

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

This example shows how to configure active/passive chassis clustering for SRX1500 device.

- [Requirements on page 316](#)
- [Overview on page 316](#)
- [Configuration on page 319](#)
- [Verification on page 324](#)

---

### 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.

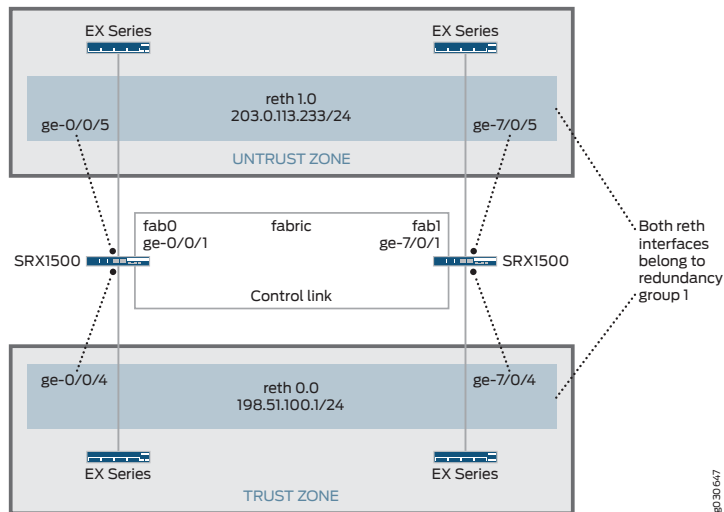
```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

---

### 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 57 on page 317](#).) When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 57: 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 25 on page 317](#) through [Table 28 on page 319](#).

Table 25: 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 26: 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

Table 26: Chassis Cluster Configuration Parameters (continued)

Feature	Name	Configuration Parameters
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: reth0
	ge-7/0/4	Redundant parent: reth0
	ge-0/0/5	Redundant parent: reth1
	ge-7/0/5	Redundant parent: reth1
	reth0	Redundancy group: 1 <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 27: 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 28: 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 {  
      gigether-options {  
        redundant-parent reth0;  
      }  
    }  
    ge-7/0/4 {  
      gigether-options {  
        redundant-parent reth0;  
      }  
    }  
    ge-0/0/5 {  
      gigether-options {  
        redundant-parent reth1;  
      }  
    }  
    ge-7/0/5 {  
      gigether-options {  
        redundant-parent reth1;  
      }  
    }  
    fab0 {  
      fabric-options {  
        member-interfaces {  
          ge-0/0/1;  
        }  
      }  
    }  
    fab1 {  
      fabric-options {  
        member-interfaces {  
          ge-7/0/1;  
        }  
      }  
    }  
  }  
}
```

```

    }
  }
}
reth0 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 198.51.100.1/24;
    }
  }
}
reth1 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family inet {
      address 203.0.113.233/24;
    }
  }
}
}
...
security {
  zones {
    security-zone untrust {
      interfaces {
        reth1.0;
      }
    }
    security-zone trust {
      interfaces {
        reth0.0;
      }
    }
  }
  policies {
    from-zone trust to-zone untrust {
      policy ANY {
        match {
          source-address any;
          destination-address any;
          application any;
        }
        then {
          permit;
        }
      }
    }
  }
}
}

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying Chassis Cluster Status on page 324](#)
- [Verifying Chassis Cluster Interfaces on page 324](#)
- [Verifying Chassis Cluster Statistics on page 325](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 326](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 326](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 327](#)
- [Troubleshooting with Logs on page 327](#)

### Verifying Chassis Cluster Status

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

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

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

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

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

### Verifying Chassis Cluster Interfaces

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

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

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

Control link status: Up

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

Fabric link status: Up

Fabric interfaces:
  Name    Child-interface  Status  Security
```

fab0	ge-0/0/1	Up	Disabled
fab0			
fab1	ge-7/0/1	Up	Disabled
fab1			

#### Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	1

#### Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	1

#### Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-0/0/4	255	Up	1
ge-7/0/4	255	Up	1
ge-0/0/5	255	Up	1
ge-7/0/5	255	Up	1

### Verifying Chassis Cluster Statistics

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

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

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

#### Control link statistics:

```
Control link 0:
  Heartbeat packets sent: 2276
  Heartbeat packets received: 2280
  Heartbeat packets errors: 0
```

#### Fabric link statistics:

```
Child link 0
  Probes sent: 2272
  Probes received: 597
```

#### Services Synchronized:

Service name	RTOs sent	RTOs received
Translation context	0	0
Incoming NAT	0	0
Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0

SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### *Verifying Chassis Cluster Control Plane Statistics*

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

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

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

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

Services Synchronized:
Service name           RTOs sent  RTOs received
Translation context    0           0
Incoming NAT           0           0
Resource manager       6           0
Session create        161         0
Session close         148         0
Session change         0           0
Gate create            0           0
Session ageout refresh requests 0           0
Session ageout refresh replies 0           0
IPSec VPN              0           0
Firewall user authentication 0           0
MGCP ALG               0           0
H323 ALG               0           0
SIP ALG               0           0
```

SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### *Verifying Chassis Cluster Redundancy Group Status*

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

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

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

Redundancy-Group: 1, Failover count: 1
  node0         100      primary no        no
  node1          1      secondary no        no
```

### *Troubleshooting with Logs*

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

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

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

- See Also**
- [Example: Setting the Node ID and Cluster ID for SRX Series Devices in a Chassis Cluster on page 110](#)
  - [Chassis Cluster Management Interfaces on page 113](#)
  - [Chassis Cluster Fabric Interfaces on page 119](#)
  - [Chassis Cluster Control Plane Interfaces on page 129](#)
  - [Chassis Cluster Redundancy Groups on page 134](#)
  - [Chassis Cluster Redundant Ethernet Interfaces on page 140](#)

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

1. Enable clustering. See Step 1 in “[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)” on page 316.
2. Configure the management interface. See Step 2 in “[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)” on page 316.
3. Configure the fabric interface. See Step 3 in “[Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)](#)” on page 316.
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.



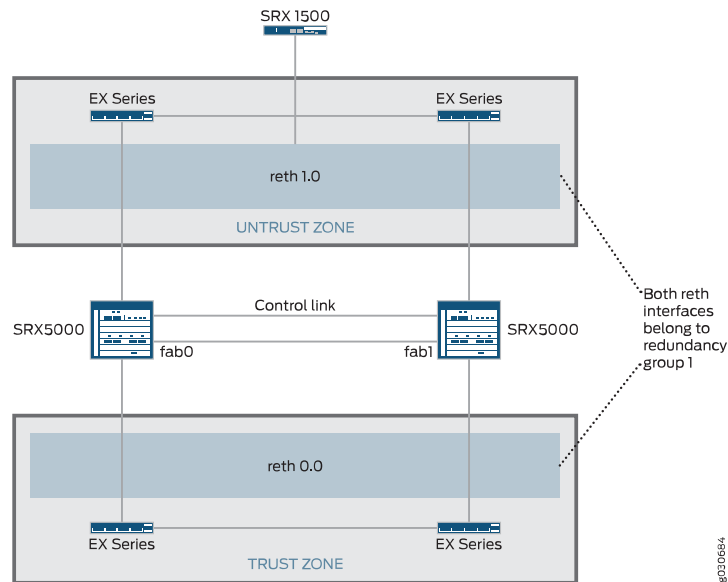
- Click **Apply**.
  - Select **ge-0/0/5**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - Select **ge-7/0/5**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - See Step 5 in [“Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)” on page 316](#) 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 316](#).
  7. Configure the security policies. See Step 7 in [“Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\)” on page 316](#).
  8. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

- See Also**
- [Understanding Active/Passive Chassis Cluster Deployment on page 298](#)
  - [Example: Configuring an Active/Passive Chassis Cluster Pair \(CLI\) on page 316](#)

## Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel

In this case, a single device in the cluster terminates in an IPsec tunnel and is used to process all traffic while the other device is used only in the event of a failure (see [Figure 58 on page 330](#)). When a failure occurs, the backup device becomes master and controls all forwarding.

**Figure 58: 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.

**See Also** • [IPsec VPN Overview](#)

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

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

- [Requirements on page 331](#)
- [Overview on page 331](#)

- [Configuration on page 335](#)
- [Verification on page 342](#)

## Requirements

---

Before you begin:

- Get two SRX5000 models with identical hardware configurations, one SRX1500 device, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:

- On node 0:

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

- On node 1:

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

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

Cluster ID greater than 15 can only be set when the fabric and control link interfaces are connected back-to-back.

- Get two SRX5000 models with identical hardware configurations, one SRX1500 edge router, and four EX Series Ethernet switches.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line.

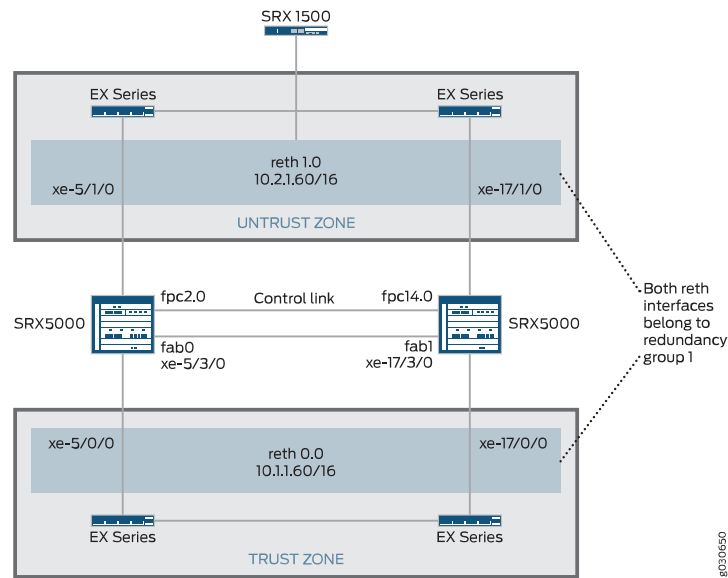
From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device. Member-specific configurations (such as the IP address of the management port of each member) are entered using configuration groups.

## Overview

---

In this example, a single device in the cluster terminates in an IPsec tunnel and is used to process all traffic, and the other device is used only in the event of a failure. (See [Figure 59 on page 332](#).) When a failure occurs, the backup device becomes master and controls all forwarding.

**Figure 59: 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 29 on page 332](#) through [Table 35 on page 335](#).

**Table 29: 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 30: Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: xe-5/3/0
	fab1	Interface: xe-17/3/0
Number of redundant Ethernet interfaces	—	2

Table 30: Chassis Cluster Configuration Parameters (continued)

Feature	Name	Configuration Parameters
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
	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 31: 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> In SRX chassis clustering, only reth and lo0 interfaces are supported for the IKE external interface configuration. Other interface types can be configured, but IPsec VPN might not work. If a lo0 logical interface is used as an IKE gateway external interface, it cannot be configured with RGO.</p>

**Table 32: IPsec Configuration Parameters**

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

**Table 33: 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 34: 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>

Table 34: Security Zone Configuration Parameters (continued)

Name	Configuration Parameters
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 35: Security Policy Configuration Parameters

Purpose	Name	Configuration Parameters
This security policy permits traffic from the trust zone to the untrust zone.	ANY	<ul style="list-style-type: none"> <li>Match criteria: <ul style="list-style-type: none"> <li>source-address any</li> <li>destination-address any</li> <li>application any</li> </ul> </li> <li>Action: permit</li> </ul>
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 gigether-options redundant-parent reth1
set interfaces xe-17/1/0 gigether-options redundant-parent reth1
set interfaces xe-5/0/0 gigether-options redundant-parent reth0
set interfaces xe-17/0/0 gigether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.60/16
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 10.2.1.60/16
set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
set security ike policy preShared mode main
set security ike policy preShared proposal-set standard
set security ike policy preShared pre-shared-key ascii-text "$ABC123"## Encrypted
password
set security ike gateway SRX1500-1 ike-policy preShared
set security ike gateway SRX1500-1 address 10.1.1.90
set security ike gateway SRX1500-1 external-interface reth0.0
set security ipsec policy std proposal-set standard
set security ipsec vpn SRX1500-1 bind-interface st0.0
set security ipsec vpn SRX1500-1 vpn-monitor optimized
set security ipsec vpn SRX1500-1 ike gateway SRX1500-1
set security ipsec vpn SRX1500-1 ike ipsec-policy std
set security ipsec vpn SRX1500-1 establish-tunnels immediately
set routing-options static route 0.0.0.0/0 next-hop 10.2.1.1
set routing-options static route 10.3.0.0/16 next-hop 10.10.1.2
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces reth0.0
set security zones security-zone vpn host-inbound-traffic system-services all 144
set security zones security-zone vpn host-inbound-traffic protocols all
set security zones security-zone vpn interfaces st0.0
set security policies from-zone trust to-zone untrust policy ANY match source-address
any
set security policies from-zone trust to-zone untrust policy ANY match destination-address
any
set security policies from-zone trust to-zone untrust policy ANY match application any
set security policies from-zone trust to-zone vpn policy vpn-any then permit

```

**Step-by-Step Procedure** To configure an active/passive chassis cluster pair with an IPsec tunnel:

1. Configure control ports.

```

{primary:node0}[edit]
user@host# set chassis cluster control-ports fpc 2 port 0
user@host# set chassis cluster control-ports fpc 14 port 0

```



2. Configure the management interface.

```
{primary:node0}[edit]
user@host# set groups node0 system host-name SRX5800-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address
172.19.100.50/24
user@host# set groups node1 system host-name SRX5800-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
172.19.100.51/24
user@host# set apply-groups "${node}"
```

3. Configure the fabric interface.

```
{primary:node0}[edit]
user@host# set interfaces fab0 fabric-options member-interfaces xe-5/3/0
user@host# set interfaces fab1 fabric-options member-interfaces xe-17/3/0
```

4. Configure redundancy groups.

```
{primary:node0}[edit]
user@host# set chassis cluster reth-count 2
user@host# set chassis cluster heartbeat-interval 1000
user@host# set chassis cluster heartbeat-threshold 3
user@host# set chassis cluster node 0
user@host# set chassis cluster node 1
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 254
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 preempt
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/1/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/1/0
weight 255
```

5. Configure redundant Ethernet interfaces.

```
{primary:node0}[edit]
user@host# set interfaces xe-5/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-17/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-5/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-17/0/0 gigether-options redundant-parent reth0
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 10.1.1.60/16
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 10.2.1.60/16
```

6. Configure IPsec parameters.

```
{primary:node0}[edit]
user@host# set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
```

```

user@host# set security ike policy preShared mode main
user@host# set security ike policy preShared proposal-set standard
user@host# set security ike policy preShared pre-shared-key ascii-text "$ABC123"##
    Encrypted password
user@host# set security ike gateway SRX1500-1 ike-policy preShared
user@host# set security ike gateway SRX1500-1 address 10.1.1.90
user@host# set security ike gateway SRX1500-1 external-interface reth0.0
user@host# set security ipsec policy std proposal-set standard
user@host# set security ipsec vpn SRX1500-1 bind-interface st0.0
user@host# set security ipsec vpn SRX1500-1 vpn-monitor optimized
user@host# set security ipsec vpn SRX1500-1 ike gateway SRX1500-1
user@host# set security ipsec vpn SRX1500-1 ike ipsec-policy std
user@host# set security ipsec vpn SRX1500-1 establish-tunnels immediately

```

7. Configure static routes.

```

{primary:node0}[edit]
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.2.1.1
user@host# set routing-options static route 10.3.0.0/16 next-hop 10.10.1.2

```

8. Configure security zones.

```

{primary:node0}[edit]
user@host# set security zones security-zone untrust host-inbound-traffic
    system-services all
user@host# set security zones security-zone untrust host-inbound-traffic protocols
    all
user@host# set security zones security-zone untrust interfaces reth1.0
user@host# set security zones security-zone trust host-inbound-traffic
    system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols
    all
user@host# set security zones security-zone trust interfaces reth0.0
user@host# set security zones security-zone vpn host-inbound-traffic
    system-services all
user@host# set security zones security-zone vpn host-inbound-traffic protocols all
user@host# set security zones security-zone vpn interfaces st0.0

```

9. Configure security policies.

```

{primary:node0}[edit]
user@host# set security policies from-zone trust to-zone untrust policy ANY match
    source-address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match
    destination-address any
user@host# set security policies from-zone trust to-zone untrust policy ANY match
    application any
user@host# set security policies from-zone trust to-zone vpn policy vpn-any then
    permit

```

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

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

```

user@host> show configuration
version x.xx.x;
groups {
  node0 {
    system {
      host-name SRX58001;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 172.19.100.50/24;
          }
        }
      }
    }
  }
  node1 {
    system {
      host-name SRX58002;
    }
    interfaces {
      fxp0 {
        unit 0 {
          family inet {
            address 172.19.100.51/24;
          }
        }
      }
    }
  }
}
apply-groups "${node}";
system {
  root-authentication {
    encrypted-password "$ABC123";
  }
}
chassis {
  cluster {
    reth-count 2;
    heartbeat-interval 1000;
    heartbeat-threshold 3;
    control-ports {
      fpc 2 port 0;
      fpc 14 port 0;
    }
    redundancy-group 0 {
      node 0 priority 254;
      node 1 priority 1;
    }
    redundancy-group 1 {
      node 0 priority 254;
      node 1 priority 1;
      preempt;
      interface-monitor {

```

```
        xe-6/0/0 weight 255;
        xe-6/1/0 weight 255;
        xe-18/0/0 weight 255;
        xe-18/1/0 weight 255;
    }
}
}
interfaces {
    xe-5/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-5/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    xe-17/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-17/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    fab0 {
        fabric-options {
            member-interfaces {
                xe-5/3/0;
            }
        }
    }
    fab1 {
        fabric-options {
            member-interfaces {
                xe-17/3/0;
            }
        }
    }
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.1.1.60/16;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 10.2.1.60/16;
            }
        }
    }
}
```

```

    }
  }
  st0 {
    unit 0 {
      multipoint;
      family inet {
        address 5.4.3.2/32;
      }
    }
  }
}
routing-options {
  static {
    route 0.0.0.0/0 {
      next-hop 10.2.1.1;
    }
    route 10.3.0.0/16 {
      next-hop 10.10.1.2;
    }
  }
}
security {
  zones {
    security-zone trust {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
      interfaces {
        reth0.0;
      }
    }
    security-zone untrust {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
      protocols {
        all;
      }
      interfaces {
        reth1.0;
      }
    }
    security-zone vpn {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
      protocols {
        all;
      }
      interfaces {
        st0.0;
      }
    }
  }
}

```

```

}
policies {
  from-zone trust to-zone untrust {
    policy ANY {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
  }
  from-zone trust to-zone vpn {
    policy vpn {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
  }
}
}

```

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

## Verification

Confirm that the configuration is working properly.

- [Verifying Chassis Cluster Status on page 342](#)
- [Verifying Chassis Cluster Interfaces on page 343](#)
- [Verifying Chassis Cluster Statistics on page 343](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 344](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 344](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 345](#)
- [Troubleshooting with Logs on page 345](#)

### Verifying Chassis Cluster Status

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

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

```

{primary:node0}
show chassis cluster status
Cluster ID: 1
Node                Priority    Status    Preempt  Manual failover

```

```

Redundancy group: 0 , Failover count: 1
  node0          1      primary no      no
  node1          254    secondary no      no

Redundancy group: 1 , Failover count: 1
  node0          1      primary yes     no
  node1          254    secondary yes    no

```

### Verifying Chassis Cluster Interfaces

**Purpose** Verify the chassis cluster interfaces.

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

```

{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

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

Interface Monitoring:
  Interface      Weight      Status      Redundancy-group
  xe-5/0/0       255        Up          1
  xe-5/1/0       255        Up          1
  xe-17/0/0      255        Up          1
  xe-17/1/0      255        Up          1

```

### Verifying Chassis Cluster Statistics

**Purpose** Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

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

```

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

Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
Services Synchronized:
  Service name      RTOs sent      RTOs received
  Translation context 0              0
  Incoming NAT       0              0

```

Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### *Verifying Chassis Cluster Control Plane Statistics*

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

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

```
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

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

```
Services Synchronized:
  Service name          RTOs sent  RTOs received
  Translation context    0           0
  Incoming NAT           0           0
```



Resource manager	6	0
Session create	161	0
Session close	148	0
Session change	0	0
Gate create	0	0
Session ageout refresh requests	0	0
Session ageout refresh replies	0	0
IPSec VPN	0	0
Firewall user authentication	0	0
MGCP ALG	0	0
H323 ALG	0	0
SIP ALG	0	0
SCCP ALG	0	0
PPTP ALG	0	0
RPC ALG	0	0
RTSP ALG	0	0
RAS ALG	0	0
MAC address learning	0	0
GPRS GTP	0	0

### *Verifying Chassis Cluster Redundancy Group Status*

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

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

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

Redundancy-Group: 1, Failover count: 1
  node0          0          primary   yes      no
  node1          254        secondary yes      no
```

### *Troubleshooting with Logs*

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

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

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

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

1. Enable clusters. See Step 1 in [“Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel”](#) on page 330.
2. Configure the management interface. See Step 2 in [“Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel”](#) on page 330.
3. Configure the fabric interface. See Step 3 in [“Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel”](#) on page 330.
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 330.

6. Configure the IPsec configuration. See Step 6 in [“Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel”](#) on page 330.

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 330.
9. Configure the security policies. See Step 9 in [“Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel”](#) on page 330.
10. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

**Related Documentation**

- [SRX Series Chassis Cluster Configuration Overview on page 81](#)
- [Configuring Chassis Clustering on an SRX Series Devices on page 150](#)

---

## Multicast Routing and Asymmetric Routing on Chassis Cluster

---

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

Multicast routing support in a chassis cluster allows different multicast protocols to send traffic across interfaces to multiple recipients. Asymmetric routing is the situation where packets from source host to destination host but follow a different path than packets from destination host to source host. For more information, see the following topics:

- [Understanding Multicast Routing on a Chassis Cluster on page 348](#)
- [Understanding Asymmetric Routing on a Chassis Cluster on page 349](#)
- [Example: Configuring an Asymmetric Chassis Cluster Pair on page 351](#)

### Understanding Multicast Routing on a Chassis Cluster

Multicast routing support across nodes in a chassis cluster allows multicast protocols, such as Protocol Independent Multicast (PIM) versions 1 and 2, Internet Group Management Protocol (IGMP), Session Announcement Protocol (SAP), and Distance Vector Multicast Routing Protocol (DVMRP), to send traffic across interfaces in the cluster. Note, however, that the multicast protocols should not be enabled on the chassis management interface (**fxp0**) or on the fabric interfaces (**fab0** and **fab1**). Multicast sessions are synched across the cluster and maintained during redundant group failovers. During failover, as with other types of traffic, there might be some multicast packet loss.

Multicast data forwarding in a chassis cluster uses the incoming interface to determine whether or not the session remains active. Packets are forwarded to the peer node if a leaf session's outgoing interface is on the peer instead of on the incoming interface's node. Multicast routing on a chassis cluster supports tunnels for both incoming and outgoing interfaces.

Multicast traffic has an upstream (toward source) and downstream (toward subscribers) direction in traffic flows. The devices replicate (fanout) a single multicast packet to multiple networks that contain subscribers. In the chassis cluster environment, multicast packet fanouts can be active on either nodes.

If the incoming interface is active on the current node and backup on the peer node, then the session is active on the current node and backup on the peer node.

Multicast configuration on a chassis cluster is the same as multicast configuration on a standalone device. See the [Multicast Protocols Feature Guide](#) for more information.

### Understanding PIM Data Forwarding

Protocol Independent Multicast (PIM) is used between devices to track the multicast packets to be forwarded to each other.

A PIM session encapsulates multicast data into a PIM unicast packet. A PIM session creates the following sessions:

- Control session
- Data session

The data session saves the control session ID. The control session and the data session are closed independently. The incoming interface is used to determine whether the PIM session is active or not. If the outgoing interface is active on the peer node, packets are transferred to the peer node for transmission.

### Understanding Multicast and PIM Session Synchronization

Synchronizing multicast and PIM sessions helps to prevent packet loss due to failover because the sessions do not need to be set up again when there is a failover.

In PIM sessions, the control session is synchronized to the backup node, and then the data session is synchronized.

In multicast sessions, the template session is synchronized to the peer node, then all the leaf sessions are synchronized, and finally the template session is synchronized again.

**See Also** • [Chassis Cluster Overview on page 25](#)

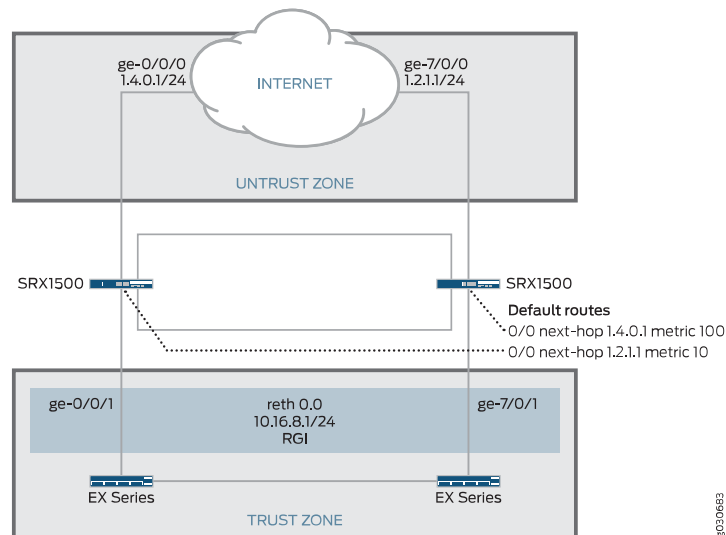
## Understanding Asymmetric Routing on a Chassis Cluster

You can use SRX Series devices in chassis clusters asymmetric routing scenarios (see [Figure 60 on page 350](#)). 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 60: 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 350](#)
- [Understanding Failures in the Untrust Zone Interfaces on page 350](#)

### 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 reth 0.0, to the Internet. Because the primary Internet connection is on node 0, sessions are created in node 0 and synced to node 1. However, sessions are only active on node 0.

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

### Understanding Failures in the Untrust Zone Interfaces

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

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

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

**See Also** • [Chassis Cluster Overview on page 25](#)

### Example: Configuring an Asymmetric Chassis Cluster Pair

This example shows how to configure a chassis cluster to allow asymmetric routing. Configuring asymmetric routing for a chassis cluster allows traffic received on either device to be processed seamlessly.

- [Requirements on page 351](#)
- [Overview on page 352](#)
- [Configuration on page 354](#)
- [Verification on page 359](#)

#### 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.
 

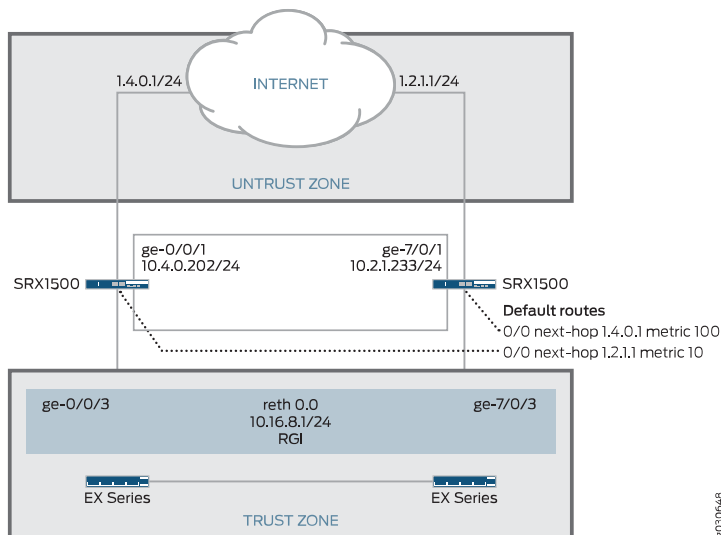
```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```
3. Connect to the other device using the console port.
  - a. Set the cluster ID and node number.
 

```
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 61 on page 352](#), 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 61: 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 36 on page 352](#) through [Table 39 on page 354](#).

**Table 36: 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 37: Chassis Cluster Configuration Parameters**

Feature	Name	Configuration Parameters
Fabric links	fab0	Interface: ge-0/0/7



Table 37: Chassis Cluster Configuration Parameters (continued)

Feature	Name	Configuration Parameters
	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 38: 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 39: 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 {
    gigether-options {
      redundant-parent reth0;
    }
  }
  ge-7/0/3 {
    gigether-options {
      redundant-parent reth0;
    }
  }
  ge-0/0/1 {
    unit 0 {
      family inet {
        address 10.4.0.202/24;
      }
    }
  }
  ge-7/0/1 {
    unit 0 {
      family inet {
        address 10.2.1.233/24;
      }
    }
  }
}
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/7;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-7/0/7;
    }
  }
}

```

```
    }
    reth0 {
      gether-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 359](#)
- [Verifying Chassis Cluster Interfaces on page 359](#)
- [Verifying Chassis Cluster Statistics on page 360](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 360](#)
- [Verifying Chassis Cluster Data Plane Statistics on page 361](#)
- [Verifying Chassis Cluster Redundancy Group Status on page 361](#)
- [Troubleshooting with Logs on page 362](#)

#### 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
  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: 258689
    Heartbeat packets received: 258684
    Heartbeat packets errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 258681
    Probes received: 258681
```

### *Verifying Chassis Cluster Data Plane Statistics*

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

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

Services Synchronized:
  Service name           RTOs sent  RTOs received
  Translation context    0          0
  Incoming NAT           0          0
  Resource manager       6          0
  Session create         160        0
  Session close          147        0
  Session change         0          0
  Gate create            0          0
  Session ageout refresh requests 0          0
  Session ageout refresh replies 0          0
  IPSec VPN              0          0
  Firewall user authentication 0          0
  MGCP ALG               0          0
  H323 ALG               0          0
  SIP ALG                0          0
  SCCP ALG               0          0
  PPTP ALG              0          0
  RPC ALG               0          0
  RTSP ALG              0          0
  RAS ALG               0          0
  MAC address learning   0          0
  GPRS GTP              0          0
```

### *Verifying Chassis Cluster Redundancy Group Status*

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

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

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

```
Cluster ID: 1
  Node          Priority  Status  Preempt  Manual failover

Redundancy-Group: 1, Failover count: 1
  node0         100      primary no        no
  node1          1      secondary no        no
```

### *Troubleshooting with Logs*

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

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

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

- Related Documentation**
- [Chassis Cluster Overview on page 25](#)
  - [Configuring Chassis Clustering on an SRX Series Devices on page 150](#)

---

## Ethernet Switching on Chassis Cluster

You can configure a chassis cluster to act as a Layer 2 Ethernet switch. For more information, see the following topics:.

- [Layer 2 Ethernet Switching Capability in a Chassis Cluster Mode on page 362](#)
- [Example: Configuring Switch Fabric Interfaces to Enable Switching in Chassis Cluster Mode on a Security Device on page 364](#)
- [Example: Configuring IRB and VLAN with Members Across Two Nodes on a Security Device on page 366](#)

### Layer 2 Ethernet Switching Capability in a Chassis Cluster Mode

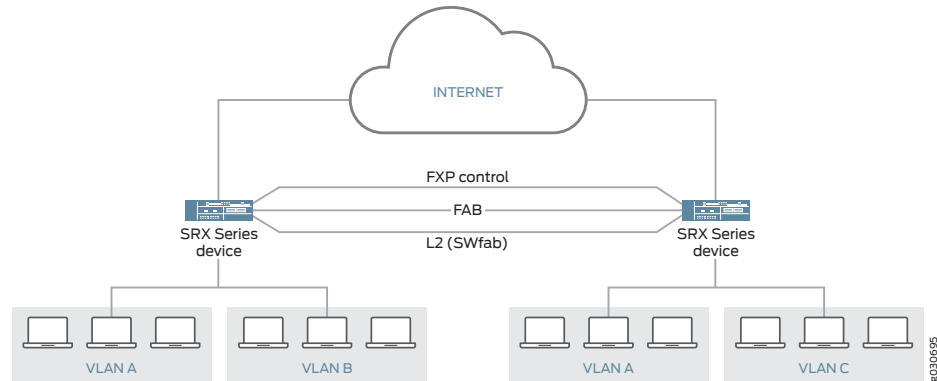
- [Understanding Layer 2 Ethernet Switching Capability in a Chassis Cluster on SRX Series Devices on page 362](#)
- [Understanding Chassis Cluster Failover and New Primary Election on page 363](#)
- [Benefits of Ethernet Switching on Chassis Cluster on page 363](#)

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

Ethernet ports support various Layer 2 features such as spanning-tree protocols (STPs), IEEE 802.1x, Link Layer Discovery Protocol (LLDP), and Multiple VLAN Registration Protocol (MVRP). With the extension of Layer 2 switching capability to devices in a chassis cluster, you can use Ethernet switching features on both nodes of a chassis cluster.

Figure 62 on page 363 shows the Layer 2 switching across chassis cluster nodes.

Figure 62: 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*. Its purpose is to carry Layer 2 traffic between nodes.



**NOTE:** Configuring a LAG with members and reth interface with family ethernet switching across nodes is not supported.



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

### Understanding Chassis Cluster Failover and New Primary Election

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



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

### Benefits of Ethernet Switching on Chassis Cluster

- Enables Ethernet switching functionality on both nodes of a chassis cluster and provides the option to configure the ethernet ports on either node for family Ethernet switching.
- Enables configuring a Layer 2 VLAN domain with member ports from both nodes and the Layer 2 switching protocols on both devices.

**See Also** • [Ethernet Switching and Layer 2 Transparent Mode Overview](#)

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

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

- [Requirements on page 364](#)
- [Overview on page 364](#)
- [Configuration on page 364](#)

---

### Requirements

- The physical link used as the switch fabric member must be directly connected to the device. (this sentence is confusing)
- Switching fabric interfaces must be configured on ports that support switching features. See *Ethernet Ports Switching Overview for Security Devices* for information about the ports on which switching features are supported.

The physical link used as the switch fabric member must be directly connected to the device. Switching supported ports must be used for switching fabric interfaces. See *Ethernet Ports Switching Overview for Security Devices* for switching supported ports.

Before you begin, See “[Example: Configuring the Chassis Cluster Fabric Interfaces](#)” on [page 125](#).

---

### Overview

In this example, pseudointerfaces swfab0 and swfab1 are created for Layer 2 fabric functionality. You also configure dedicated Ethernet ports on each node to be associated with the swfab interfaces.

---

### Configuration

#### CLI Quick Configuration

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

```
set interfaces swfab0 fabric-options member-interfaces ge-0/0/9
set interfaces swfab0 fabric-options member-interfaces ge-0/0/10
set interfaces swfab1 fabric-options member-interfaces ge-7/0/9
set interfaces swfab1 fabric-options member-interfaces ge-7/0/10
```

#### Step-by-Step Procedure

To configure swfab interfaces:

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

```
{primary:node0} [edit]
user@host# set interfaces swfab0 fabric-options member-interfaces ge-0/0/9
user@host# set interfaces swfab0 fabric-options member-interfaces ge-0/0/10
user@host# set interfaces swfab1 fabric-options member-interfaces ge-7/0/9
user@host# set interfaces swfab1 fabric-options member-interfaces ge-7/0/10
```

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

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

**Results** From configuration mode, confirm your configuration by entering the **show interfaces swfab0** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces swfab0
fabric-options{
  member-interfaces {
    ge-0/0/9;
    ge-0/0/10;
  }
}
```

### Verification

To confirm that the configuration is working properly, perform these tasks:

- [Verifying Switching Fabric Ports on page 365](#)

### Verifying Switching Fabric Ports

**Purpose** Verify that you are able to configure multiple ports as members of switching fabric ports.

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

```
user@host# show interfaces swfab0
fabric-options{
  member-interfaces {
    ge-0/0/9;
    ge-0/0/10;
  }
}
```

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

```
user@host# run show chassis cluster ethernet-switching interfaces
swfab0:
  Name              Status
  ge-0/0/9          up
  ge-0/0/10         up
```

```

swfab1:
  Name      Status
  ge-7/0/9  up
  ge-7/0/10 up

```

**See Also** • [SRX Series Chassis Cluster Configuration Overview on page 81](#)

## Example: Configuring IRB and VLAN with Members Across Two Nodes on a Security Device

- [Requirements on page 366](#)
- [Overview on page 366](#)
- [Configuration on page 366](#)
- [Verification on page 368](#)

### Requirements

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

### Overview

This example shows the configuration of integrated routing and bridging (IRB) and configuration of a VLAN with members across node 0 and node 1.

### Configuration

#### CLI Quick Configuration

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

```

set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-7/0/5 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members vlan100
set interfaces irb unit 100 family inet address 192.0.2.100/24
set vlans vlan100 vlan-id 100
set vlans vlan100 l3-interface irb.100

```

#### Step-by-Step Procedure

To configure IRB and a VLAN:

1. Configure Ethernet switching on the node0 interface.

```

{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode
access
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode
access

```

2. Configure Ethernet switching on the node1 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching interface-mode
trunk
```

3. Create VLAN vlan100 with vlan-id 100.

```
{primary:node0} [edit]
user@host# set vlans vlan100 vlan-id 100
```

4. Add interfaces from both nodes to the VLAN.

```
{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members
vlan100
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members
vlan100
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members
vlan100
```

5. Create an IRB logical interface.

```
user@host# set interfaces irb unit 100 family inet address 192.0.2.100/24
```

6. Associate an IRB interface with the VLAN.

```
user@host# set vlans vlan100 l3-interface irb.100
```

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

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

**Results** From configuration mode, confirm your configuration by entering the **show vlans** and **show interfaces** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct the configuration.

```
[edit]
user@host# show vlans
vlan100 {
  vlan-id 100;
  l3-interface irb.100;
}
[edit]
user@host# show interfaces
ge-0/0/3 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members vlan100;
      }
    }
  }
}
```

```

    }
  }
}
ge-0/0/4 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members vlan100;
      }
    }
  }
}
ge-7/0/5 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members vlan100;
      }
    }
  }
}
}
}
irb {
  unit 100 {
    family inet {
      address 192.0.2.100/24;
    }
  }
}
}

```

## Verification

### Verifying VLAN and IRB

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

**Action** From operational mode, enter the **show interfaces terse ge-0/0/3** command to view the node 0 interface.

```

user@host> show interfaces terse ge-0/0/3
Interface      Admin Link Proto  Local          Remote
ge-0/0/3       up    up
ge-0/0/3.0     up    up  eth-switch

```

From operational mode, enter the **show interfaces terse ge-0/0/4** command to view the node 0 interface.

```

user@host> show interfaces terse ge-0/0/4
Interface      Admin Link Proto  Local          Remote
ge-0/0/4       up    up
ge-0/0/4.0     up    up  eth-switch

```



From operational mode, enter the **show interfaces terse ge-7/0/5** command to view the node1 interface.

```
user@host> show interfaces terse ge-7/0/5
Interface           Admin Link Proto  Local           Remote
ge-7/0/5             up    up
ge-7/0/5.0           up    up  eth-switch
```

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

```
user@host> show vlans
Routing instance    VLAN name    Tag    Interfaces
default-switch      default      1
default-switch      vlan100      100    ge-0/0/3.0*
                                   ge-0/0/4.0*
                                   ge-7/0/5.0*
```

From operational mode, enter the **show ethernet-switching interface** command to view the information about Ethernet switching interfaces.

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                        LH - MAC limit hit, DN - interface down,
                        MMAS - Mac-move action shutdown, AS - Autostate-exclude
                        enabled,
                        SCTL - shutdown by Storm-control )

Logical      Vlan      TAG    MAC    STP      Logical
Tagging      members
interface
ge-0/0/3.0    untagged              limit state interface flags
16383        DN
              vlan100    100    1024    Discarding
              untagged              16383        DN
              vlan100    100    1024    Discarding
              untagged              16383        DN
ge-0/0/4.0    untagged              16383        DN
              tagged              16383        DN
ge-7/0/5.0    tagged              16383        DN
              tagged              16383        DN
              vlan100    100    1024    Discarding
              tagged
```

**Meaning** The output shows the VLAN and IRB are configured and working fine.

**See Also** • [Example: Configuring an IRB Interface](#)

**Related Documentation** • [Configuring Chassis Clustering on an SRX Series Devices on page 150](#)

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

---

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

Media Access Control Security (MACsec) is an industry-standard security technology that provides secure communication for all traffic on Ethernet links. For more information, see the following topics:

- [Understanding Media Access Control Security \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)

### Understanding Media Access Control Security (MACsec)

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

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

Starting in Junos OS Release 15.1X49-D60, Media Access Control Security (MACsec) is supported on control and fabric ports of SRX340 and SRX345 devices in chassis cluster mode.

Starting in Junos OS Release 17.4R1, MACsec is supported on HA control and fabric ports of SRX4600 devices in chassis cluster mode.

This topic contains the following sections:

- [How MACsec Works on page 370](#)
- [Understanding Connectivity Associations and Secure Channels on page 371](#)
- [Understanding Static Connectivity Association Key Security Mode on page 371](#)
- [MACsec Considerations on page 372](#)

#### How MACsec Works

---

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

MACsec provides industry-standard security through the use of secured point-to-point Ethernet links. The point-to-point links are secured after matching security keys. When you enable MACsec using static connectivity association key (CAK) security mode,

user-configured pre-shared keys are exchanged and verified between the interfaces at each end of the point-to-point Ethernet link.

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

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

MACsec can also be used to encrypt all traffic on the Ethernet link. The encryption used by MACsec ensures that the data in the Ethernet frame cannot be viewed by anybody monitoring traffic on the link.

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

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

---

### Understanding Connectivity Associations and Secure Channels

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

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

---

### Understanding Static Connectivity Association Key Security Mode

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

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

Once matching pre-shared keys are successfully exchanged, the MACsec Key Agreement (MKA) protocol is enabled. The MKA protocol is responsible for maintaining MACsec on the link, and decides which switch on the point-to-point link becomes the key server. The

key server then creates an SAK that is shared with the switch at the other end of the point-to-point link only, and that SAK is used to secure all data traffic traversing the link. The key server will continue to periodically create and share a randomly-created SAK over the point-to-point link for as long as MACsec is enabled.

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

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



**NOTE:** Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, the 802.1x protocol process (daemon) does not support restart on SRX340 and SRX345 devices.

---

### 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:** For MACsec configurations, identical configurations must exist on both the ends. That is, each node should contain the same configuration as the other node. If the other node is not configured or improperly configured with MACsec on the other side, the port is disabled and stops forwarding the traffic.



**WARNING:** Prior to 15.1X49-D100, SRX340 and SRX345 devices did not support MACsec for host-to-host or switch-to-host connections.



**NOTE:** SRX4600 devices currently do not support MACsec for host-to-host connections.



**NOTE:** On SRX340 and SRX345 devices, fabric interfaces must be configured such that the Media Access Control Security (MACsec) configurations are local to the nodes. Otherwise, the fabric link will not be reachable

## Configuring Media Access Control Security (MACsec)

Starting in Junos OS Release 15.1X49-D60, Media Access Control Security (MACsec) is supported on control and fabric ports of SRX340 and SRX345 devices in chassis cluster mode.

Starting in Junos OS Release 17.4R1, MACsec is supported on control and fabric ports of SRX4600 devices in chassis cluster mode.

This topic shows how to configure MACsec on control and fabric ports of supported SRX Series device in chassis cluster to secure point-to-point Ethernet links between the peer devices in a cluster. Each point-to-point Ethernet link that you want to secure using MACsec must be configured independently. You can enable MACsec encryption on device-to-device links using static connectivity association key (CAK) security mode.

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

- [Configuration Considerations When Configuring MACsec on Chassis Cluster Setup on page 373](#)
- [Configuring MACsec Using Static Connectivity Association Key Security Mode on page 375](#)
- [Configuring Static CAK on the Chassis Cluster Control Port on page 379](#)
- [Configuring Static CAK on the Chassis Cluster Fabric Port on page 380](#)
- [Configuring Static CAK on the Control Port of SRX4600 Device in Chassis Cluster on page 380](#)
- [Verifying MACSEC Configuration on page 381](#)

### Configuration Considerations When Configuring MACsec on Chassis Cluster Setup

Before you begin, follow these steps to configure MACsec on control ports:

1. If the chassis cluster is already up, disable it by using the **set chassis cluster disable** command and reboot both nodes.
2. Configure MACsec on the control port with its attributes as described in the following sections “[Configuring Static CAK on the Chassis Cluster Control Port](#)” on page 379. Both nodes must be configured independently with identical configurations.
3. Enable the chassis cluster by using **set chassis cluster cluster-id *id*** on both of the nodes. Reboot both nodes.

Control port states affect the integrity of a chassis cluster. Consider the following when configuring MACsec on control ports:

- Any new MACsec chassis cluster port configurations or modifications to existing MACsec chassis cluster port configurations will require the chassis cluster to be disabled. Once disabled, you can apply the preceding configurations and reenabling 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 300 seconds when MACsec on the chassis cluster control port is enabled on SRX340 and SRX345 devices.



**NOTE:** If both control link fail, Junos OS changes the operating state of the secondary node to ineligible for a 180 seconds. When MACsec is enabled on the control port, the ineligibility duration is 200 seconds for SRX4600 devices.



**NOTE:** Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, the initial hold timer is extended from 30 seconds to 120 seconds in chassis clusters on SRX340 and SRX345 devices.



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

Consider the following when configuring MACsec on fabric ports:

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

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



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

### Configuring MACsec Using Static Connectivity Association Key Security Mode

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



**NOTE:** For SRX340 and SRX345 devices, ge-0/0/0 is a fabric port and ge-0/0/1 is a control port for the chassis cluster and assigned as cluster-control-port 0.



**NOTE:** For SRX4600 devices, dedicated control and fabric ports are available. You can configure MACsec on control ports (control port 0 [em0] and port 1 [em1]) and fabric ports 0 [fab 0] and [fab 1] on SRX4600 devices.

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

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

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

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

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

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

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

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

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

3. Create the preshared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

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

A preshared key is exchanged between directly-connected links to establish a MACsec-secure link. The pre-shared-key includes the CKN and the CAK. The CKN is

a 64-digit hexadecimal number and the CAK is a 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.

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

To configure a CKN of

**11c1c1c11xx012xx5xx8ef284aa23ff6729xx2e4xxx66e91fe34ba2cd9fe311** and CAK of **228xx255aa23xx6729xx664xxx66e91f** on connectivity association **ca1**:

```
[edit security macsec]
user@host# set connectivity-association ca1 pre-shared-key ckn
11c1c1c11xx012xx5xx8ef284aa23ff6729xx2e4xxx66e91fe34ba2cd9fe311
user@host# set connectivity-association ca1 pre-shared-key cak
228xx255aa23xx6729xx664xxx66e91f
```



**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
```

5. (Optional) Set the MKA transmit interval.

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

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

The default *interval* is 2000 ms. 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.



**NOTE:** Starting from Junos OS Release 17.4, for SRX340, SRX345, and SRX4600, the default MKA transmit interval is 10000 ms on HA links.

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

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

6. (Optional) Disable MACsec encryption.

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

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

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

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

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

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

```
[edit security macsec connectivity-association ca1]
```

```
user@host# set offset 30
```

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

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

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

8. (Optional) Enable replay protection.

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

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

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

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

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

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

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

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

9. (Optional) Exclude a protocol from MACsec.

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

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

```
[edit security macsec connectivity-association connectivity-association-name]
```

```
user@host# set exclude-protocol lldp
```

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

10. Assign the connectivity association to a chassis cluster control interface.

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

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

For instance, to assign connectivity association ca1 to interface ge-0/0/1 (For SRX340/SRX345):

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

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

```
[edit security macsec]
user@host# set cluster-data-port port-number connectivity-association
connectivity-association-name
[edit security macsec]
user@host# set cluster-data-port interfaces ge-5/0/2 connectivity-association ca1
```

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

### Configuring Static CAK on the Chassis Cluster Control Port

To establish a CA over a chassis cluster control link on two SRX345 devices.

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

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

2. Create the preshared key by configuring the connectivity association key name (CKN).

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

The CKN must contain 32 hexadecimal characters.

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

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

The CAK must contain 64 hexadecimal characters.

4. Specify chassis cluster control ports for the connectivity association.

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

### Configuring Static CAK on the Chassis Cluster Fabric Port

---

To establish a connectivity association over a chassis cluster fabric link on two SRX345 devices:

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

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

2. Create the preshared key by configuring the connectivity association key name (CKN).

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

The CKN must contain 32 hexadecimal characters.

3. Create the preshared key by configuring the connectivity association key (CAK).

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

The CAK must contain 64 hexadecimal characters.

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

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

### Configuring Static CAK on the Control Port of SRX4600 Device in Chassis Cluster

---

Use this procedure to establish a CA over a chassis cluster control link on two SRX4600 devices.

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

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

2. Create the preshared key by configuring the connectivity association key name (CKN).

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

The CKN must contain 32 hexadecimal characters.

3. Create the preshared key by configuring the connectivity association key (CAK).

```
[edit]
user@host# set security macsec connectivity-association ca1 pre-shared-key cak
""$9$XX.XXXrXX8XX69X0X1yrevXXX-Xb24oXhXrvX8dXwXgoaXji.
Xfz7-XYg4XjHqmf5Xn6Xpu1XXjqmX3n/Xtu0IXhreX8XX"
```

The CAK must contain 64 hexadecimal characters.

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

```
[edit]
user@host# set security macsec cluster-control-port 0 connectivity-association ca1
user@host# set security macsec cluster-control-port 1 connectivity-association ca1
```

### Verifying MACSEC Configuration

To confirm that the configuration provided in “Configuring Static CAK on the Control Port of SRX4600 Device in Chassis Cluster” on page 380 is working properly, perform these tasks:

- [Display the Status of Active MACsec Connections on the Device on page 381](#)
- [Display MACsec Key Agreement \(MKA\) Session Information on page 382](#)
- [Verifying That MACsec-Secured Traffic Is Traversing Through the Interface on page 382](#)
- [Verifying Chassis Cluster Ports Are Secured with MACsec Configuration on page 383](#)

#### *Display the Status of Active MACsec Connections on the Device*

**Purpose** Verify that MACsec is operational on the chassis cluster setup.

**Action** From the operational mode, enter the **show security macsec connections interface-name** command on one or both of the nodes of chassis cluster setup.

```
{primary:node0}[edit]
user@host# show security macsec connections

Interface name: em0
CA name: ca1
Cipher suite: GCM-AES-128   Encryption: on
Key server offset: 0        Include SCI: no
Replay protect: off        Replay window: 0
Outbound secure channels
  SC Id: 02:00:00:01:01:04/1
  Outgoing packet number: 1
  Secure associations
    AN: 3 Status: inuse Create time: 00:01:43
Inbound secure channels
  SC Id: 02:00:00:02:01:04/1
  Secure associations
    AN: 3 Status: inuse Create time: 00:01:43
```

**Meaning** The **Interface name** and **CA name** outputs show that the MACsec connectivity association is operational on the interface em0. The output does not appear when the connectivity association is not operational on the interface.

#### *Display MACsec Key Agreement (MKA) Session Information*

**Purpose** Display MACsec Key Agreement (MKA) session information for all interfaces.

**Action** From the operational mode, enter the **show security mka sessions** command.

```
user@host> show security mka sessions
Interface name: em0
  Member identifier: B51CXXX2678A7F5F6C12345
  CAK name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

  Transmit interval: 10000(ms)
  Outbound SCI: 02:00:00:01:01:04/1
  Message number: 270           Key number: 8
  Key server: yes               Key server priority: 16
  Latest SAK AN: 3              Latest SAK KI: B51C8XXX2678A7A5B6C54321/8
  Previous SAK AN: 0            Previous SAK KI: 0000000000000000000000/0
  Peer list
  1. Member identifier: 0413427B38817XXXXF054321 (live)
     Message number: 8 Hold time: 59000 (ms)
     SCI: 02:00:00:02:01:04/1
     Lowest acceptable PN: 0
```

**Meaning** The outputs show the status of MKA sessions.

#### *Verifying That MACsec-Secured Traffic Is Traversing Through the Interface*

**Purpose** Verify that traffic traversing through the interface is MACsec-secured.

**Action** From the operational mode, enter the **show security macsec statistics** command.

```
user@host> show security macsec statistics interface em0 detail
```

```
Interface name: em0
  Secure Channel transmitted
    Encrypted packets: 2397305
    Encrypted bytes: 129922480
    Protected packets: 0
    Protected bytes: 0
  Secure Association transmitted
    Encrypted packets: 2397305
    Protected packets: 0
  Secure Channel received
    Accepted packets: 2395850
    Validated bytes: 0
    Decrypted bytes: 131715088
  Secure Association received
    Accepted packets: 2395850
```

```
Validated bytes: 0
Decrypted bytes: 0
```

**Meaning** The **Encrypted packets** line under the **Secure Channel transmitted** field are the values incremented each time a packet is sent from the interface that is secured and encrypted by MACsec.

The **Accepted packets** line under the **Secure Association received** field are the values incremented each time a packet that has passed the MACsec integrity check is received on the interface. The **Decrypted bytes** line under the **Secure Association received** output is incremented each time an encrypted packet is received and decrypted.

### *Verifying Chassis Cluster Ports Are Secured with MACsec Configuration*

**Purpose** Verify that MACsec is configured on chassis cluster ports.

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

```
user@host> show chassis cluster interfaces
```

```
Control link status: Up
```

```
Control interfaces:
```

Index	Interface	Monitored-Status	Internal-SA	Security
0	em0	Up	Disabled	Secured

```
Fabric link status: Down
```

```
Fabric interfaces:
```

Name	Child-interface	Status (Physical/Monitored)	Security
fab0	xe-1/1/6	Up / Down	Disabled
fab0			
fab1	xe-8/1/6	Up / Down	Disabled
fab1			

```
Redundant-ethernet Information:
```

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	2
reth2	Down	Not configured
reth3	Down	Not configured
reth4	Down	Not configured
reth5	Down	Not configured
reth6	Down	Not configured
reth7	Down	Not configured

```
Redundant-pseudo-interface Information:
```

Name	Status	Redundancy-group
lo0	Up	0

**Meaning** The **Security** line under the **Control interfaces** output for em0 interface shown as **Secured** means that the traffic sent from the em0 interface is secured and encrypted by MACsec.

You can also use the **show chassis cluster status** command to display the current status of the chassis cluster.

- See Also**
- [Understanding Media Access Control Security \(MACsec\) on page 370](#)
  - [macsec on page 450](#)

Release History Table

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, MACsec is supported on HA control and fabric ports of SRX4600 devices in chassis cluster mode
15.1X49-D60	Starting in Junos OS Release 15.1X49-D60, Media Access Control Security(MACsec) is supported on control and fabric ports of SRX340 and SRX345 devices in chassis cluster mode.

- Related Documentation**
- [SRX Series Chassis Cluster Configuration Overview on page 81](#)
  - [Understanding SRX Series Chassis Cluster Slot Numbering and Physical Port and Logical Interface Naming on page 86](#)
  - [macsec on page 450](#)

## Understanding SCTP Behavior in Chassis Cluster

**Supported Platforms** [SRX1500](#), [SRX4100](#), [SRX4200](#), [SRX5400](#), [SRX5600](#), [SRX5800](#), [vSRX](#)

In a chassis cluster configuration mode, the SCTP configuration and the established SCTP association is synced with the peer device. The SCTP module supports both active-active and active-passive modes.

The established SCTP association sends a creation or deletion message to the peer whenever an association is created or deleted on the active device. The secondary device adds or deletes an association respectively upon receiving the message from the established SCTP association. SCTP module then registers the corresponding callback function to receive and handle this message. There is no continuous timer sync between the two associations.

SCTP module will register a cold start sync function when a secondary device joins the cluster or reboots. The SCTP cold start function is called to sync all SCTP associations with the peer devices at the same time.

After the switchover, the established SCTP associations will remain functioning, but the associations in the progress of establishment will be lost and the establishment procedure needs to be re-initiated. It is also possible that the associations in the progress of teardown miss the ack message and leaves unestablished SCTP associations in the firewall. These



associations will be cleaned up when the timer expires (5 hours by default) due to no activity in the association.



NOTE:

- You should configure all policies for your required SCTP sessions. For example, suppose you have endpoints A and B. Endpoint A has one SCTP association with x number of IPs (IP\_a1, IP\_a2, IP\_a3...IP\_ax). Endpoint B has one SCTP association with y number of IPs (IP\_b1, IP\_b2, IP\_b3...IP\_by.) The policy on the security device should permit all possible x\*y paths in both directions.
- When an SCTP association is removed, the related SCTP sessions still exist and time out by themselves.

**Related  
Documentation**

- *Understanding Stream Control Transmission Protocol*
- *SCTP Configuration Overview*
- *SCTP Packet Structure Overview*
- *SCTP Features Overview*



## CHAPTER 5

# Upgrading or Disabling a Chassis Cluster

- [Upgrading Individual Devices in a Chassis Cluster Separately on page 387](#)
- [Upgrading Devices in a Chassis Cluster Using ICU on page 388](#)
- [Upgrading a Chassis Cluster Using In-Service Software Upgrade on page 392](#)
- [Disabling a Chassis Cluster on page 408](#)

## 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 Devices in a Chassis Cluster Using ICU on page 388](#)
- [Upgrading a Chassis Cluster Using In-Service Software Upgrade on page 392](#)

## Upgrading Devices in a Chassis Cluster Using ICU

---

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

The chassis cluster ICU feature allows both devices in a cluster to be upgraded from supported Junos OS versions using a single command. For more information, see the following topics:

- [Upgrading Both Devices in a Chassis Cluster Using ICU on page 388](#)
- [Upgrading ICU Using a Build Available Locally on a Primary Node in a Chassis Cluster on page 389](#)
- [Upgrading ICU Using a Build Available on an FTP Server on page 390](#)
- [Aborting an Upgrade in a Chassis Cluster During an ICU on page 390](#)

### Upgrading Both Devices in a Chassis Cluster Using ICU

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

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.

You must use the in-band cluster upgrade (ICU) commands on SRX1500 device to upgrade following Junos OS Releases:

- Junos OS Release 15.1X49-D50 to Junos OS Release 15.1X49-D60
- Junos OS Release 15.1X49-D60 to Junos OS Release 15.1X49-D70
- Junos OS Release 15.1X49-D50 to Junos OS Release 15.1X49-D70

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.

- 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 389](#) and [“Upgrading ICU Using a Build Available on an FTP Server” on page 390](#).
- 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, RG0, changes to the secondary node after an ICU upgrade.
- During ICU, the chassis cluster redundancy groups are failed over to the primary node to change the cluster to active/passive mode.
- ICU states can be checked from the syslog or with the console/terminal logs.
- ICU requires that both nodes be running a dual-root partitioning scheme with one exception being the SRX1500. ICU will not continue if it fails to detect dual-root partitioning on either of the nodes. Requirement of the dual-root partitioning is applicable only for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Dual-root partitioning is not supported on SRX1500 devices. SRX1500 uses solid-state drive (SSD) as secondary storage.

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



**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 prior to Junos OS Release 15.1X49-D70)
```

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



**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 prior to Junos OS Release 15.1X49-D70) Start ICU by entering the following command:

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

```
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

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 40 on page 391 lists the options and their descriptions for the **request system software in-service-upgrade** command.

*Table 40: 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**

- [Upgrading Individual Devices in a Chassis Cluster Separately on page 387](#)
- [Upgrading a Chassis Cluster Using In-Service Software Upgrade on page 392](#)
- [Disabling a Chassis Cluster on page 408](#)

---

## Upgrading a Chassis Cluster Using In-Service Software Upgrade

**Supported Platforms**    [SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, SRX5800](#)

In-service software upgrade (ISSU) enables a software upgrade from one Junos OS version to a later Junos OS version with minimal downtime. For more information, see the following topics:

- [Understanding ISSU for a Chassis Cluster on page 392](#)
- [ISSU System Requirements on page 396](#)
- [Upgrading Both Devices in a Chassis Cluster Using ISSU on page 397](#)
- [Rolling Back Devices in a Chassis Cluster After an ISSU on page 399](#)
- [Enabling an Automatic Chassis Cluster Node Failback After an ISSU on page 400](#)
- [Understanding Log Error Messages for Troubleshooting ISSU-Related Problems on page 400](#)
- [Managing Chassis Cluster ISSU-Related Problems on page 405](#)

### Understanding ISSU for a Chassis Cluster

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

The chassis cluster ISSU feature enables both devices in a cluster to be upgraded from supported Junos OS versions with a minimal disruption in traffic and without a disruption in service.

Starting with Junos OS Release 15.1X49-D80, SRX4100 and SRX4200 devices support ISSU.

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



- On SRX1500, SRX4100, and SRX4200 devices, ISSU is not supported for upgrading to 17.4 releases from previous Junos OS releases. ISSU is supported for upgrading from Junos OS 17.4 to successive 17.4 releases.
- On SRX5400, SRX5600 and SRX5800 devices, ISSU is not supported for upgrading to 17.3 and higher releases from earlier Junos OS releases. ISSU is supported for upgrading from Junos OS 17.3 to Junos 17.4 releases.
- SRX300 Series devices, SRX550M devices and vSRX do not support ISSU.

**NOTE:**

You can use the in-band cluster upgrade (ICU) commands on SRX4100 and SRX4200 devices to upgrade following Junos OS Releases:

- Junos OS Release 15.1X49-D65 to Junos OS Release 15.1X49-D70
  - Junos OS Release 15.1X49-D70 to Junos OS Release 15.1X49-D80.
- 

You must use the in-band cluster upgrade (ICU) commands on SRX1500 device to upgrade following Junos OS Releases:

- Junos OS Release 15.1X49-D50 to Junos OS Release 15.1X49-D60
- Junos OS Release 15.1X49-D60 to Junos OS Release 15.1X49-D70
- Junos OS Release 15.1X49-D50 to Junos OS Release 15.1X49-D70

ISSU provides the following benefits:

- Eliminates network downtime during software image upgrades
- Reduces operating costs, while delivering higher service levels
- Allows fast implementation of new features

**NOTE:**

ISSU has the following limitations:

- ISSU is available only for Junos OS Release 10.4R4 or later.
- ISSU does not support software downgrades.
- If you upgrade from a Junos OS version that supports only IPv4 to a version that supports both IPv4 and IPv6, the IPv4 traffic continue to work during the upgrade process. If you upgrade from a Junos OS version that supports both IPv4 and IPv6 to a version that supports both IPv4 and IPv6, both the IPv4 and IPv6 traffic continue to work during the upgrade process. Junos OS Release 10.2 and later releases support flow-based processing for IPv6 traffic.
- During an ISSU, you cannot bring any PICs online. You cannot perform operations such as commit, restart, or halt.
- During an ISSU, operations like fabric monitoring, control link recovery, and RGX preempt are suspended.
- During an ISSU, you cannot commit any configurations.



**NOTE:** For details about ISSU support status, see knowledge base article [KB17946](#).

The following process occurs during an ISSU for devices in a chassis cluster. The sequences given below are applicable when RG-0 is node 0 (primary node). Note that you must initiate an ISSU from RG-0 primary. If you initiate the upgrade on node 1 (RG-0 secondary), an error message is displayed.

1. At the beginning of a chassis cluster ISSU, the system automatically fails over all RG-1+ redundancy groups that are not primary on the node from which the ISSU is initiated. This action ensures that all the redundancy groups are active on only the RG-0 primary node.



**NOTE:** The automatic failover of all RG-1+ redundancy groups is available from Junos OS release 12.1 or later. If you are using Junos OS release 11.4 or earlier, before starting the ISSU, ensure that all the redundancy groups are all active on only the RG-0 primary node.

After the system fails over all RG-1+ redundancy groups, it sets the manual failover bit and changes all RG-1+ primary node priorities to 255, regardless of whether the redundancy group failed over to the RG-0 primary node.

2. The primary node (node 0) validates the device configuration to ensure that it can be committed using the new software version. Checks are made for disk space availability

for the `/var` file system on both nodes, unsupported configurations, and unsupported Physical Interface Cards (PICs).

If the disk space available on either of the Routing Engines is insufficient, the ISSU process fails and returns an error message. However, unsupported PICs do not prevent the ISSU. The software issues a warning to indicate that these PICs will restart during the upgrade. Similarly, an unsupported protocol configuration does not prevent the ISSU. However, the software issues a warning that packet loss might occur for the protocol during the upgrade.

3. When the validation succeeds, the kernel state synchronization daemon (`ksyncd`) synchronizes the kernel on the secondary node (node 1) with the node 0.
4. Node 1 is upgraded with the new software image. Before being upgraded, the node 1 gets the configuration file from node 0 and validates the configuration to ensure that it can be committed using the new software version. After being upgraded, it is resynchronized with node 0.
5. The chassis cluster process (`chassisd`) on the node 0 prepares other software processes for the ISSU. When all the processes are ready, `chassisd` sends a message to the PICs installed in the device.
6. The Packet Forwarding Engine on each Flexible PIC Concentrator (FPC) saves its state and downloads the new software image from node 1. Next, each Packet Forwarding Engine sends a message (`unified-ISSU ready`) to the `chassisd`.
7. After receiving the message (`unified-ISSU ready`) from a Packet Forwarding Engine, the `chassisd` sends a reboot message to the FPC on which the Packet Forwarding Engine resides. The FPC reboots with the new software image. After the FPC is rebooted, the Packet Forwarding Engine restores the FPC state and a high-speed internal link is established with node 1 running the new software. The `chassisd` is also reestablished with node 0.
8. After all Packet Forwarding Engines have sent a *ready* message using the `chassisd` on node 0, other software processes are prepared for a node switchover. The system is ready for a switchover at this point.
9. Node switchover occurs and node 1 becomes the new primary node (hitherto secondary node 1).
10. The new secondary node (hitherto primary node 0) is now upgraded to the new software image.

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



**NOTE:** When upgrading a version cluster that does not support encryption to a version that supports encryption, upgrade the first node to the new version. Without the encryption configured and enabled, two nodes with different versions can still communicate with each other and service is not broken. After upgrading the first node, upgrade the second node to the new version. Users can decide whether to turn on the encryption feature after completing the upgrade. Encryption must be deactivated before downgrading to a version that does not support encryption. This ensures that communication between an encryption-enabled version node and a downgraded node does not break, because both are no longer encrypted.

## ISSU System Requirements

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

To perform an ISSU, your device must be running a Junos OS release that supports ISSU for the specific platform. See [Table 41 on page 396](#) for platform support.

*Table 41: ISSU Platform Support*

Device	Junos OS Release
SRX5800	10.4R4 or later
SRX5600	10.4R4 or later
SRX5400	12.1X46-D20 or later
SRX1500	15.1X49-D70 or later
SRX4100	15.1X49-D80 or later
SRX4200	15.1X49-D80 or later
SRX4600	17.4R1 or later



**NOTE:** For additional details on ISSU support and limitations, see [ISSU/ICU Upgrade Limitations on SRX Series Devices](#).

Note the following limitations related to an ISSU:

- The ISSU process is aborted if the Junos OS version specified for installation is a version earlier than the one currently running on the device.
- The ISSU process is aborted if the specified upgrade conflicts with the current configuration, the components supported, and so forth.

- ISSU does not support the extension application packages developed using the Junos OS SDK.
- ISSU does not support version downgrading on all supported SRX Series devices.
- ISSU occasionally fails under heavy CPU load.



**NOTE:** To downgrade from an ISSU-capable release to an earlier release (ISSU-capable or not), use the `request system software add` command. Unlike an upgrade using the ISSU process, a downgrade using the `request system software add` command might cause network disruptions and loss of data.

We strongly recommend that you perform ISSU under the following conditions:

- When the devices are operating in chassis cluster mode
- When both the primary and secondary nodes are healthy
- During system maintenance period
- During the lowest possible traffic period
- When the Routing Engine CPU usage is less than 40 percent

In cases where ISSU is not supported or recommended, while still downtime during the system upgrade must be minimized, the minimal downtime procedure can be used, see knowledge base article [KB17947](#).

## Upgrading Both Devices in a Chassis Cluster Using ISSU

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

Before you begin the ISSU for upgrading both the devices, note the following guidelines:

- Back up the software using the `request system snapshot` command on each Routing Engine to back up the system software to the device's hard disk.
- If you are using Junos OS Release 11.4 or earlier, before starting the ISSU, set the failover for all redundancy groups so that they are all active on only one node (primary). See [“Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 255](#).

If you are using Junos OS Release 12.1 or later, Junos OS automatically fails over all RGs to the RGO primary.

- We recommend that you enable graceful restart for routing protocols before you start an ISSU.



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

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

Starting with Junos OS Release 15.1X49-D80, SRX4100 and SRX4200 devices support ISSU.

Starting with Junos OS Release 17.4R1, SRX4600 devices support ISSU.

To perform an ISSU from the CLI:

1. Download the software package from the Juniper Networks Support website:  
<http://www.juniper.net/support/downloads/>
2. Copy the package on primary node of the cluster. We recommend that you copy the package to the **/var/tmp** directory, which is a large file system on the hard disk. Note that the node from where you initiate the ISSU must have the software image.  
  
**user@host> file copy ftp://username:prompt@ftp.hostname.net/filename /var/tmp/filename**
3. Verify the current software version running on both nodes by issuing the **show version** command on the primary node.
4. Start the ISSU from the node that is primary for all the redundancy groups by entering the following command:

```
user@host> request system software in-service-upgrade image-name-with-full-path reboot
```



.....  
**NOTE:** For SRX1500, SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, and SRX5800 devices, you must include **reboot** in the command. If **reboot** is not included, the command fails.  
.....



.....  
**NOTE:** For SRX1500, SRX4100, and SRX4200 devices, you can optionally remove the original image file by including **unlink** in the command.  
.....

```
user@host> request system software in-service-upgrade  
image-name-with-full-path reboot unlink
```

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

5. Wait a few minutes, and then log in to the device again. Verify by using the **show version** command that both devices in the cluster are running the new Junos OS release.

6. Verify that all policies, zones, redundancy groups, and other real-time objects (RTOs) return to their correct states.
7. Make node 0 the primary node again by issuing the **request chassis cluster failover node *node-number* redundancy-group *group-number*** command.



**NOTE:** If you want redundancy groups to automatically return to node 0 as the primary after an in-service software upgrade (ISSU), you must set the redundancy group priority such that node 0 is primary and enable the **preempt** option. Note that this method works for all redundancy groups except redundancy group 0. You must manually set the failover for redundancy group 0.

To set the redundancy group priority and enable the **preempt** option, see [“Example: Configuring Chassis Cluster Redundancy Groups” on page 138](#).

To manually set the failover for a redundancy group, see [“Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 255](#).



**NOTE:** During the upgrade, both devices might experience redundancy group failovers, but traffic is not disrupted. Each device validates the package and checks version compatibility before beginning the upgrade. If the system finds that the new package version is not compatible with the currently installed version, the device refuses the upgrade or prompts you to take corrective action. Sometimes a single feature is not compatible, in which case, the upgrade software prompts you to either abort the upgrade or turn off the feature before beginning the upgrade.



**NOTE:** If you want to operate the SRX Series device back as a standalone device or to remove a node from a chassis cluster, ensure that you have aborted the ISSU procedure on both the nodes (in case ISSU procedure is initiated)

**See Also** • [In-Service Hardware Upgrade for SRX5K-RE-1800X4 and SRX5K-SCBE in a Chassis Cluster](#)

## Rolling Back Devices in a Chassis Cluster After an ISSU

If an ISSU fails to complete and only one device in the cluster is upgraded, you can roll back to the previous configuration on the upgraded device alone by issuing one of the following commands on the upgraded device:

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

- [request system reboot](#)

## Enabling an Automatic Chassis Cluster Node Failback After an ISSU

If you want redundancy groups to automatically return to node 0 as the primary after the an in-service software upgrade (ISSU), you must set the redundancy group priority such that node 0 is primary and enable the **preempt** option. Note that this method works for all redundancy groups except redundancy group 0. You must manually set the failover for a redundancy group 0. To set the redundancy group priority and enable the **preempt** option, see [“Example: Configuring Chassis Cluster Redundancy Groups” on page 138](#). To manually set the failover for a redundancy group, see [“Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 255](#).



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

---

## Understanding Log Error Messages for Troubleshooting ISSU-Related Problems

The following problems might occur during an ISSU upgrade. You can identify the errors by using the details in the logs. For detailed information about specific system log messages, see [System Log Explorer](#).

- [Chassisd Process Errors on page 400](#)
- [Understanding Common Error Handling for ISSU on page 400](#)
- [ISSU Support-Related Errors on page 403](#)
- [Initial Validation Checks Failure on page 403](#)
- [Installation-Related Errors on page 404](#)
- [Redundancy Group Failover Errors on page 405](#)
- [Kernel State Synchronization Errors on page 405](#)

---

### Chassisd Process Errors

**Problem**    **Description:** Errors related to chassisd.

**Solution**    Use the error messages to understand the issues related to chassisd.

When ISSU starts, a request is sent to chassisd to check whether there are any problems related to the ISSU from a chassis perspective. If there is a problem, a log message is created.

---

### Understanding Common Error Handling for ISSU

**Problem**    **Description:** You might encounter some problems in the course of an ISSU. This section provides details on how to handle them.



**Solution** Any errors encountered during an ISSU result in the creation of log messages, and ISSU continues to function without impact to traffic. If reverting to previous versions is required, the event is either logged or the ISSU is halted, so as not to create any mismatched versions on both nodes of the chassis cluster. [Table 42 on page 401](#) provides some of the common error conditions and the workarounds for them. The sample messages used in the [Table 42 on page 401](#) are from the SRX1500 device and are also applicable to all supported SRX Series devices.

**Table 42: ISSU-Related Errors and Solutions**

Error Conditions	Solutions
Attempt to initiate an ISSU when previous instance of an ISSU is already in progress	<p>The following message is displayed:</p> <p><b>warning: ISSU in progress</b></p> <p>You can abort the current ISSU process, and initiate the ISSU again using the <b>request chassis cluster in-service-upgrade abort</b> command.</p>
Reboot failure on the secondary node	<p>No service downtime occurs, because the primary node continues to provide required services. Detailed console messages are displayed requesting that you manually clear existing ISSU states and restore the chassis cluster.</p> <pre>error: [Oct 6 12:30:16]: Reboot secondary node failed (error-code: 4.1)  error: [Oct 6 12:30:16]: ISSU Aborted! Backup node maybe in inconsistent state, Please restore backup node [Oct 6 12:30:16]: ISSU aborted. But, both nodes are in ISSU window. Please do the following: 1. Rollback the node with the newer image using rollback command Note: use the 'node' option in the rollback command  otherwise, images on both nodes will be rolled back  2. Make sure that both nodes (will) have the same image  3. Ensure the node with older image is primary for all RGs 4. Abort ISSU on both nodes 5. Reboot the rolled back node</pre> <p>Starting with Junos OS Release 17.4R1, the hold timer for the initial reboot of the secondary node during the ISSU process is extended from 15 minutes (900 seconds) to 45 minutes (2700 seconds) in chassis clusters on SRX1500, SRX4100, SRX4200, and SRX4600 devices.</p>

Table 42: ISSU-Related Errors and Solutions (continued)

Error Conditions	Solutions
Secondary node failed to complete the cold synchronization	<p>The primary node times out if the secondary node fails to complete the cold synchronization. Detailed console messages are displayed that you manually clear existing ISSU states and restore the chassis cluster. No service downtime occurs in this scenario.</p> <pre>[Oct 3 14:00:46]: timeout waiting for secondary node node1 to sync(error-code: 6.1) Chassis control process started, pid 36707  error: [Oct 3 14:00:46]: ISSU Aborted! Backup node has been upgraded, Please restore backup node [Oct 3 14:00:46]: ISSU aborted. But, both nodes are in ISSU window. Please do the following: 1. Rollback the node with the newer image using rollback command Note: use the 'node' option in the rollback command otherwise, images on both nodes will be rolled back  2. Make sure that both nodes (will) have the same image  3. Ensure the node with older image is primary for all RGs 4. Abort ISSU on both nodes 5. Reboot the rolled back node</pre>
Failover of newly upgraded secondary failed	<p>No service downtime occurs, because the primary node continues to provide required services. Detailed console messages are displayed requesting that you manually clear existing ISSU states and restore the chassis cluster.</p> <pre>[Aug 27 15:28:17]: Secondary node0 ready for failover. [Aug 27 15:28:17]: Failing over all redundancy-groups to node0 ISSU: Preparing for Switchover error: remote rg1 priority zero, abort failover. [Aug 27 15:28:17]: failover all RGs to node node0 failed (error-code: 7.1) error: [Aug 27 15:28:17]: ISSU Aborted! [Aug 27 15:28:17]: ISSU aborted. But, both nodes are in ISSU window. Please do the following: 1. Rollback the node with the newer image using rollback command Note: use the 'node' option in the rollback command otherwise, images on both nodes will be rolled back 2. Make sure that both nodes (will) have the same image 3. Ensure the node with older image is primary for all RGs 4. Abort ISSU on both nodes 5. Reboot the rolled back node {primary:node1}</pre>
Upgrade failure on primary	<p>No service downtime occurs, because the secondary node fails over as primary and continues to provide required services.</p>

Table 42: ISSU-Related Errors and Solutions (continued)

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

### ISSU Support-Related Errors

**Problem**    **Description:** Installation failure occurs because of unsupported software and unsupported feature configuration.

**Solution**    Use the following error messages to understand the compatibility-related problems:

```
WARNING: Current configuration not compatible with
/var/tmp/junos-srx5000-11.4X3.2-domestic.tgz
Exiting in-service-upgrade window
Exiting in-service-upgrade window
```

### Initial Validation Checks Failure

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

**Solution**    The validation checks fail if the image is not present or if the image file is corrupt. The following error messages are displayed when initial validation checks fail when the image is not present and the ISSU is aborted:

#### When Image Is Not Present

```
user@host> ...0120914_srx_12q1_major2.2-539764-domestic.tgz reboot
Chassis ISSU Started
Chassis ISSU Started
ISSU: Validating Image
Initiating in-service-upgrade
Initiating in-service-upgrade
Fetching package...
error: File does not exist:
/var/tmp/junos-srx1k3k-12.1I20120914_srx_12q1_major2.2-539764-domestic.tgz
error: Couldn't retrieve package
/var/tmp/junos-srx1k3k-12.1I20120914_srx_12q1_major2.2-539764-domestic.tgz
Exiting in-service-upgrade window
Exiting in-service-upgrade window
Chassis ISSU Aborted
Chassis ISSU Aborted
Chassis ISSU Aborted
```

```
ISSU: IDLE
ISSU aborted; exiting ISSU window.
```

### When Image File Is Corrupted

If the image file is corrupted, the following output displays:

```
user@host> ...junos-srx1k3k-11.4X9-domestic.tgz_1 reboot
Chassis ISSU Started
node1:
-----
Chassis ISSU Started
ISSU: Validating Image
Initiating in-service-upgrade

node1:
-----
Initiating in-service-upgrade
ERROR: Cannot use /var/tmp/junos-srx1k3k-11.4X9-domestic.tgz_1:
gzip: stdin: invalid compressed data--format violated
tar: Child returned status 1
tar: Error exit delayed from previous errors
ERROR: It may have been corrupted during download.
ERROR: Please try again, making sure to use a binary transfer.
Exiting in-service-upgrade window

node1:
-----
Exiting in-service-upgrade window
Chassis ISSU Aborted
Chassis ISSU Aborted

node1:
-----
Chassis ISSU Aborted
ISSU: IDLE
ISSU aborted; exiting ISSU window.

{primary:node0}
```

The primary node validates the device configuration to ensure that it can be committed using the new software version. If anything goes wrong, the ISSU aborts and error messages are displayed.

---

### Installation-Related Errors

**Problem**    **Description:** The install image file does not exist or the remote site is inaccessible.

**Solution**    Use the following error messages to understand the installation-related problems:

```
error: File does not exist: /var/tmp/junos-srx5000-11.4X3.2-domest
error: Couldn't retrieve package /var/tmp/junos-srx5000-11.4X3.2-domest
```

ISSU downloads the install image as specified in the ISSU command as an argument. The image file can be a local file or located at a remote site. If the file does not exist or the remote site is inaccessible, an error is reported.

---

### Redundancy Group Failover Errors

---

**Problem**    **Description:** Problem with automatic redundancy group (RG) failure.

**Solution**    Use the following error messages to understand the problem:

```
failover all RG 1+ groups to node 0
error: Command failed.  None of the redundancy-groups has been failed over.
      Some redundancy-groups on node1 are already in manual failover mode.
      Please execute 'failover reset all' first..
```

---

### Kernel State Synchronization Errors

---

**Problem**    **Description:** Errors related to ksyncd.

**Solution**    Use the following error messages to understand the issues related to ksyncd:

```
Failed to get kernel-replication error information from Standby Routing Engine.
mgd_slave_peer_has_errors() returns error at line 4414 in mgd_package_issu.
```

ISSU checks whether there are any ksyncd errors on the secondary node (node 1) and displays the error message if there are any problems and aborts the upgrade.

## Managing Chassis Cluster ISSU-Related Problems

This topic includes the following sections:

- [Viewing ISSU Progress on page 405](#)
- [Stopping ISSU Process if it Halts During an Upgrade on page 406](#)
- [Recovering the Node in Case of a Failed ISSU on page 406](#)

---

### Viewing ISSU Progress

---

**Problem**    **Description:** Rather than wait for an ISSU failure, you can display the progress of the ISSU as it occurs, noting any message indicating that the ISSU was unsuccessful. Providing such messages to JTAC can help with resolving the issue.

**Solution** After starting an ISSU, issue the **show chassis cluster information issu** command. Output similar to the following is displayed indicating the progress of the ISSU for all Services Processing Units (SPUs).

```
Note: Any management session to secondary node will be disconnected.
Shutdown NOW!
[pid 2480]
ISSU: Backup RE Prepare Done
Waiting for node1 to reboot.
Current time: Tue Apr 22 14:37:32 2014
Max. time to complete: 15min 0sec.
Note: For real time ISSU status, open a new management session and run
<show chassis cluster information issu> for detail information
node1 booted up.
Waiting for node1 to become secondary
Current time: Tue Apr 22 14:40:32 2014
Max. time to complete: 60min 0sec.
Note: For real time ISSU status, open a new management session and run
<show chassis cluster information issu> for detail information
node1 became secondary.
Waiting for node1 to be ready for failover
ISSU: Preparing Daemons
Current time: Tue Apr 22 14:41:27 2014
Max. time to complete: 60min 0sec.
Note: For real time ISSU status, open a new management session and run
<show chassis cluster information issu> for detail information
Secondary node1 ready for failover.
Installing package
'/var/tmp/junos-srx5000-12.1I20140421_srx_12q1_x47.0-643920-domestic.tgz' ...
Verified SHA1 checksum of issu-indb.tgz
Verified junos-boot-srx5000-12.1I20140421_srx_12q1_x47.0-643920.tgz signed by
PackageDevelopment_12_1_0
Verified junos-srx5000-12.1I20140421_srx_12q1_x47.0-643920-domestic signed by
PackageDevelopment_12_1_0
```

---

### Stopping ISSU Process if it Halts During an Upgrade

---

**Problem** **Description:** The ISSU process halts in the middle of an upgrade.

**Solution** If the ISSU fails to complete and only one device in the cluster is upgraded, you can roll back to the previous configuration on the upgraded device alone by issuing one of the following commands on the upgraded device:

- **request chassis cluster in-service-upgrade abort** to abort the ISSU on both nodes.
- **request system software rollback *node node-id* reboot** to roll back the image.
- **request system reboot** to reboot the rolled back node.

---

### Recovering the Node in Case of a Failed ISSU

---

**Problem** **Description:** The ISSU procedure stops progressing.

**Solution** Open a new session on the primary device and issue the **request chassis cluster in-service-upgrade abort** command.

This step aborts an in-progress ISSU. This command must be issued from a session other than the one on which you issued the **request system in-service-upgrade** command that launched the ISSU. If the node is being upgraded, this command cancels the upgrade. The command is also helpful in recovering the node in case of a failed ISSU.

When an ISSU encounters an unexpected situation that necessitates an abort, the system message provides you with detailed information about when and why the upgrade stopped along with recommendations for the next steps to take.

For example, the following message is issued when a node fails to become RG-0 secondary when it boots up:

```
Rebooting Secondary Node
Shutdown NOW!
[pid 2120]
ISSU: Backup RE Prepare Done
Waiting for node1 to reboot.
node1 booted up.
Waiting for node1 to become secondary
error: wait for node1 to become secondary failed (error-code: 5.1)
ISSU aborted. But, both nodes are in ISSU window.
Please do the following:
1. Log on to the upgraded node.
2. Rollback the image using rollback command with node option
Note: Not using the 'node' option might cause
the images on both nodes to be rolled back
3. Make sure that both nodes (will) have the same image
4. Ensure the node with older image is primary for all RGs
5. Abort ISSU on both nodes
6. Reboot the rolled back node
{primary:node0}
```



**NOTE:** If you attempt to upgrade a device pair running a Junos OS release earlier than Release 9.6, ISSU fails without changing anything on either device in the cluster. Devices running Junos OS releases earlier than Release 9.6 must be upgraded separately using individual device upgrade procedures.

If the secondary device experiences a power-off condition before it boots up using the new image specified when the ISSU was initiated, the newly upgraded device will still be waiting to end the ISSU after power is restored. To end the ISSU, issue the **request chassis cluster in-service-upgrade abort** command.

Release History Table

Release	Description
15.1X49-D70	Starting with Junos OS Release 15.1X49-D70, SRX1500 devices support ISSU.
15.1X49-D70	Starting with Junos OS Release 15.1X49-D70, SRX1500 devices support ISSU.
15.1X49-D80	Starting with Junos OS Release 15.1X49-D80, SRX4100 and SRX4200 devices support ISSU.
15.1X49-D80	Starting with Junos OS Release 15.1X49-D80, SRX4100 and SRX4200 devices support ISSU.
17.4R1	Starting with Junos OS Release 17.4R1, SRX4600 devices support ISSU.
Junos OS Release 17.4R1	Starting with Junos OS Release 17.4R1, the hold timer for the initial reboot of the secondary node during the ISSU process is extended from 15 minutes (900 seconds) to 45 minutes (2700 seconds) in chassis clusters on SRX1500, SRX4100, SRX4200, and SRX4600 devices.

#### Related Documentation

- [Upgrading Individual Devices in a Chassis Cluster Separately on page 387](#)
- [Upgrading Devices in a Chassis Cluster Using ICU on page 388](#)
- [Disabling a Chassis Cluster on page 408](#)

## Disabling a Chassis Cluster

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

If you want to operate the SRX Series device back as a standalone device or to remove a node from a chassis cluster, you must disable the chassis cluster.

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 node 0:

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



- To disable cluster on node 1:

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



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

**Related  
Documentation**

- [Upgrading Individual Devices in a Chassis Cluster Separately on page 387](#)
- [Upgrading Devices in a Chassis Cluster Using ICU on page 388](#)



## PART 1

# Configuration Statements and Operational Commands

- Configuration Statements on page 413
- Operational Commands on page 483



## CHAPTER 6

# Configuration Statements

- [aggregated-devices](#) on page 415
- [apply-groups \(Chassis Cluster\)](#) on page 416
- [arp-detect](#) on page 417
- [arp-throttle](#) on page 418
- [authentication-key](#) on page 419
- [authentication-type](#) on page 420
- [cak](#) on page 421
- [ckn](#) on page 422
- [cluster \(Chassis\)](#) on page 423
- [configuration-synchronize \(Chassis Cluster\)](#) on page 425
- [connectivity-association](#) on page 426
- [connectivity-association \(MACsec Interfaces\)](#) on page 427
- [control-link-recovery](#) on page 428
- [control-ports](#) on page 429
- [exclude-protocol](#) on page 430
- [fabric-options](#) on page 431
- [gether-options \(Chassis Cluster\)](#) on page 433
- [global-threshold](#) on page 435
- [global-weight](#) on page 436
- [gratuitous-arp-count](#) on page 437
- [heartbeat-interval](#) on page 438
- [heartbeat-threshold](#) on page 439
- [hold-down-interval](#) on page 440
- [include-sci](#) on page 441
- [interface \(Chassis Cluster\)](#) on page 442
- [interfaces \(MACsec\)](#) on page 443
- [interface-monitor](#) on page 444
- [ip-monitoring](#) on page 445

- [key-server-priority \(MACsec\) on page 447](#)
- [lacp \(Interfaces\) on page 448](#)
- [link-protection \(Chassis Cluster\) on page 449](#)
- [macsec on page 450](#)
- [mka on page 451](#)
- [must-secure on page 452](#)
- [network-management on page 453](#)
- [no-encryption \(MACsec\) on page 454](#)
- [node \(Chassis Cluster Redundancy Group\) on page 455](#)
- [ntp on page 456](#)
- [ntp threshold on page 457](#)
- [offset on page 459](#)
- [preempt \(Chassis Cluster\) on page 461](#)
- [pre-shared-key on page 462](#)
- [priority \(Protocols VRRP\) on page 463](#)
- [redundancy-group \(Chassis Cluster\) on page 464](#)
- [redundant-ether-options on page 466](#)
- [redundant-parent \(Interfaces\) on page 467](#)
- [redundant-pseudo-interface-options on page 468](#)
- [replay-protect on page 469](#)
- [replay-window-size on page 470](#)
- [reth-count \(Chassis Cluster\) on page 471](#)
- [retry-count \(Chassis Cluster\) on page 472](#)
- [retry-interval \(Chassis Cluster\) on page 473](#)
- [route-active-on on page 474](#)
- [security-mode on page 475](#)
- [traceoptions \(Chassis Cluster\) on page 476](#)
- [transmit-interval \(MACsec\) on page 478](#)
- [virtual-address on page 479](#)
- [vrrp-group on page 480](#)
- [weight on page 482](#)

## aggregated-devices

**Supported Platforms** M Series, MX Series, PTX Series, SRX Series, T Series

**Syntax**

```
aggregated-devices {
  ethernet {
    device-count number;
    lacp {
      link-protection {
        non-revertive;
      }
      system-priority;
    }
  }
  sonet {
    device-count number;
  }
  maximum-links maximum-links-limit;
}
```

**Hierarchy Level** [edit chassis]

**Release Information** Statement introduced before Junos OS Release 7.4.  
Support for LACP link protection and system priority introduced in Junos OS Release 9.3.

**Description** Configure properties for aggregated devices on the router.

**Options** The remaining statements are explained separately.

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

**Related Documentation**

- *Configuring Junos OS for Supporting Aggregated Devices*

## apply-groups (Chassis Cluster)

---

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

**Syntax** `apply-groups [$node]`

**Hierarchy Level** `[edit chassis cluster]`

**Release Information** Statement introduced in Junos OS Release 9.0.

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

You can use the **[edit groups]** option to create a unique configuration between the cluster members by defining a unique hostname for each node and assigning a unique IP address for the fxp0 interface on each node.

Each group is named after the node it is applied to (node 0 and node 1) and once you apply the configured groups using the **apply-groups** statement, only the group that matches the node name is applied.

The configuration you specified under group node 0 will be active only on node 0 and the configuration you specified under group node 1 will be active only on node 1.

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

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

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)



## arp-detect

**Supported Platforms** SRX3400, SRX3600, SRX5400, SRX5600, SRX5800

**Syntax** arp-detect *milliseconds*;

**Hierarchy Level** [edit forwarding-options next-hop]

**Release Information** Statement introduced in Junos OS Release 12.3X48-D65 and Junos OS Release 15.1X49-D130.

**Description** Define the length of time (in milliseconds) for an SPU to wait for an acknowledgement from the Routing Engine that an ARP request has been received from the SPU. If the Routing Engine fails to respond within the specific time interval, the SPU considers that the Routing Engine CPU utilization is high at that moment, and initiates the ARP throttling. ARP throttling is initiated on the logical interface, where the incoming packet had triggered the ARP request.

Configuring a shorter ARP detect time interval results in triggering of ARP throttling more frequently. Frequent ARP throttling is useful for lowering Routing Engine CPU utilization caused by excessive ARP requests.

For example, when you configure the **set forwarding-options nexthop arp-detect 300** option, the nexthop resolution request must be acknowledged by the Routing Engine within 300 milliseconds. If the SPU does not get an acknowledgment from the Routing Engine in 300 milliseconds, the logical interface which had received the packet that triggered the nexthop request, changes into ARP throttle state. While the ARP throttle state is active for that interface, traffic entering into that interface does not trigger new nexthop resolution requests.



**CAUTION:** We recommend that only advanced Junos OS users attempt to configure ARP throttle and ARP detect feature. Because, improper configuration might result in high CPU utilization of Routing Engine affecting other processes on your device.

**Options** *milliseconds*—Number of seconds the SPU waits before receiving a response from Routing Engine.

**Range:** 1 through 10000 milliseconds

**Default:** 10000 milliseconds

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

Related Documentation • [arp-throttle on page 418](#)

---

## arp-throttle

---

**Supported Platforms** [SRX5400, SRX5600, SRX5800](#)

**Syntax** `arp-throttle seconds;`

**Hierarchy Level** `[edit forwarding-options next-hop]`

**Release Information** Statement introduced in Junos OS Release 12.3X48-D65, Junos OS Release 15.1X49-D60, and Junos OS Release 15.1X49-D130.

**Description** Define the time duration (in seconds) for Address Resolution Protocol (ARP) request throttling to remain active when it is triggered.

When ARP throttling is triggered, it is active for a given logical interface. For the configured duration of time, the ARP throttling remains active, and the traffic entering into the specific interface does not trigger ARP nexthop resolution requests, which are being sent to the Routing Engine from the specific SPU.

When you configure a longer time duration, the ARP throttling can protect the Routing Engine by preventing too many ARP requests being triggered by incoming traffic.

For example, if there is a large amount of traffic destined to a directly connected, unresolved IP address, chances of getting frequent ARP requests is very high, which eventually results into a high CPU load on the Routing Engine. By setting a longer time interval of the ARP throttle, the Routing Engine is protected from numerous ARP requests.

For example, when you configure the `set forwarding-options nexthop arp-throttle 15` option, and the interface state changes to throttle state, the nexthop requests triggered by incoming traffic into this interface, will not be sent. After 15 seconds, interface changes back from throttle to normal state. Then the nexthop requests triggered by the incoming traffic into this interface are sent to the Routing Engine again.

**Options** *seconds*—Time interval (in seconds) for Address Resolution Protocol (ARP) request throttling to remain active when it is triggered.

**Range:** 10 through 100 seconds

**Default:** 10 seconds

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

Related Documentation • [arp-detect on page 417](#)

## authentication-key

**Supported Platforms** [EX Series](#), [M Series](#), [MX Series](#), [NFX Series](#), [OCX1100](#), [QFX Series](#), [SRX Series](#), [T Series](#)

**Syntax** authentication-key *key*;

**Hierarchy Level** [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],  
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

**Release Information** Statement introduced before Junos OS Release 7.4.  
Statement introduced in Junos OS 11.3 for the QFX Series.  
Statement introduced in Junos OS Release 14.1x53-D20 for the OCX Series.  
Statement introduced in Junos OS Release 18.1R1 for the SRX Series devices.

**Description** Configure a Virtual Router Redundancy Protocol (VRRP) IPv4 authentication key. You also must specify a VRRP authentication scheme by including the **authentication-type** statement.

All devices in the VRRP group must use the same authentication scheme and password.



**NOTE:** When VRRPv3 is enabled, the **authentication-type** and **authentication-key** statements cannot be configured for any VRRP groups.

**Options** *key*—Authentication password. For simple authentication, it can be 1 through 8 characters long. For Message Digest 5 (MD5) authentication, it can be 1 through 16 characters long. If you include spaces, enclose all characters in quotation marks (" ").

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

**Related Documentation**

- [Configuring VRRP Authentication \(IPv4 Only\)](#)
- [authentication-type on page 420](#)
- [version-3](#)
- [Understanding VRRP on SRX Series Devices on page 279](#)
- [Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces on page 282](#)

## authentication-type

**Supported Platforms** EX Series, M Series, MX Series, NFX Series, OCX1100, QFabric System, QFX Series, SRX Series, T Series

**Syntax** authentication-type (md5 | simple);

**Hierarchy Level** [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*],  
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-id*]

**Release Information** Statement introduced before Junos OS Release 7.4.  
Statement introduced in Junos OS 11.3 for the QFX Series.  
Statement introduced in Junos OS Release 14.1x53-D20 for the OCX Series.  
Statement introduced in Junos OS Release 18.1R1 for the SRX Series devices.

**Description** Enable Virtual Router Redundancy Protocol (VRRP) IPv4 authentication and specify the authentication scheme for the VRRP group. If you enable authentication, you must specify a password by including the **authentication-key** statement.

All devices in the VRRP group must use the same authentication scheme and password.



**NOTE:** When VRRPv3 is enabled, the **authentication-type** and **authentication-key** statements cannot be configured for any VRRP groups.

**Options** authentication—Authentication scheme:

- **simple**—Use a simple password. The password is included in the transmitted packet, so this method of authentication is relatively insecure.
- **md5**—Use the MD5 algorithm to create an encoded checksum of the packet. The encoded checksum is included in the transmitted packet. The receiving routing platform uses the authentication key to verify the packet, discarding it if the digest does not match. This algorithm provides a more secure authentication scheme.

**Default:** none (no authentication is performed).

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

**Related Documentation**

- [Configuring VRRP Authentication \(IPv4 Only\)](#)
- [authentication-key on page 419](#)
- [version-3](#)

- [Understanding VRRP on SRX Series Devices on page 279](#)
- [Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces on page 282](#)

## cak

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

**Syntax** `ckn hexadecimal-number;`

**Hierarchy Level** [edit security macsec connectivity-association pre-shared-key]

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

**Description** Specifies the connectivity association key (CAK) for a pre-shared key.

To configure MACsec on the supported ports, you need to create the preshared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

A preshared key is exchanged between directly-connected links to establish a MACsec-secure link. 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.

**Default** No CAK exists, by default.

**Options** *hexadecimal-number* —The key name, in hexadecimal format.

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.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## ckn

---

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

**Syntax** *ckn hexadecimal-number;*

**Hierarchy Level** [edit security macsec connectivity-association pre-shared-key]

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

**Description** Specifies the connectivity association key name (CKN) for a pre-shared key.

To configure MACsec on supported ports, you need to create the preshared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

A preshared key is exchanged between directly-connected links to establish a MACsec-secure link. 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.

**Default** No CKN exists, by default.

**Options** *hexadecimal-number* —The key name, in hexadecimal format.

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.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## cluster (Chassis)

Supported Platforms [SRX Series, vSRX](#)

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

Hierarchy Level [\[edit chassis\]](#)

<b>Release Information</b>	Statement introduced in Junos OS Release 9.0.
<b>Description</b>	Configure a chassis cluster. You can perform the configuration under the [edit chassis cluster] configuration stanza to define chassis cluster configuration, operations, and monitoring. The configuration includes specifying configuration synchronization, control link recovery, heartbeat interval and threshold, network management, redundancy group, and traceoptions.
<b>Options</b>	<p><b>configuration-synchronize</b>—Disable automatic chassis cluster synchronization. See <a href="#">configuration-synchronize (Chassis Cluster)</a>.</p> <p><b>control-link-recovery</b>—Enable automatic control link recovery option.</p> <p><b>control-ports</b>—Enable specific chassis cluster control ports.</p> <p><b>Values:</b></p> <ul style="list-style-type: none"><li>• <b>fpc</b>—FPC slot number</li><li>• <b>port</b>—Port number</li></ul> <p><b>heartbeat-interval</b>—Interval between successive heartbeats (milliseconds)</p> <p><b>Default:</b> 1000</p> <p><b>Range:</b> 1000-2000</p> <p><b>heartbeat-threshold</b>—Number of consecutive missed heartbeats to indicate device failure</p> <p><b>Default:</b> 3</p> <p><b>Range:</b> 3-8</p> <p><b>network-management</b>—Define parameters for network management. See <a href="#">network-management</a>.</p> <p><b>redundancy-group <i>name</i></b>—Define a redundancy group. See <a href="#">redundancy-group (Chassis Cluster)</a>.</p> <p><b>reth-count</b>—Number of redundant ethernet interfaces</p> <p><b>Range:</b> 1-128</p> <p><b>traceoptions</b>—Define chassis cluster redundancy process tracing operations. See <a href="#">traceoptions (Chassis Cluster)</a>.</p> <p>The remaining statements are explained separately. See <a href="#">CLI Explorer</a>.</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="#">ip-monitoring on page 445</a></li></ul>



## configuration-synchronize (Chassis Cluster)

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

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

**Hierarchy Level** [edit chassis cluster]

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

**Description** Disables the automatic chassis cluster synchronization between the primary and secondary nodes.

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. 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.

To reenable automatic chassis cluster synchronization, use the **delete chassis cluster configuration-synchronize no-secondary-bootup-auto** command in configuration mode.

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

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

**Related Documentation**

- [Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 166](#)
- [request chassis cluster configuration-synchronize on page 493](#)
- [show chassis cluster information configuration-synchronization on page 526](#)

## 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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

---

## connectivity-association (MACsec Interfaces)

---

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

**Syntax** `connectivity-association connectivity-association-name;`

**Hierarchy Level** `[edit security macsec cluster-control-port <idx>]  
[edit security macsec cluster-data-port interface]`

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

**Description** Applies a connectivity association to an interface, which enables Media Access Control Security (MACsec) on that interface.

**Default** No connectivity associations are associated with any interfaces.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## control-link-recovery

---

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

**Syntax** control-link-recovery;

**Hierarchy Level** [edit chassis cluster]

**Release Information** Statement introduced in Junos OS Release 9.5.

**Description** Enable control link recovery to be done automatically by the system.

If the control link fails, the secondary node enters a disabled state. To recover the node from the disabled mode, you must reboot the node to resume operations. You can make this reboot automatic by using the **control-link-recovery** configuration option.

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

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

**Related Documentation**

- [interface \(Chassis Cluster\) on page 442](#)

## control-ports

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

**Syntax** `control-ports fpc {  
 port;  
}`

**Hierarchy Level** [edit chassis cluster]

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

**Description** Enable the specific control port to use as a control link for the chassis cluster. By default, all control ports are disabled.

After connecting the control ports, you need to configure the control ports to setup control links as a step to establish the chassis cluster.

You need to configure a minimum of one control port per chassis of the cluster. If you configure port 0 only, the Juniper Services Redundancy Protocol process (jsrpd) does not send control heartbeats on control link 1 and the counters it sends will show zeroes.

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



**NOTE:** FPC slot range depends on platform. The maximum range of 0 through 23 applies to SRX5800 devices; for SRX5600 devices, the only applicable range is 0 through 11; for SRX5400 devices, the applicable slot range is 0 through 5. See [“Chassis Cluster Control Plane Interfaces” on page 129](#) for details.

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

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

**Related Documentation** • [cluster \(Chassis\) on page 423](#)

## exclude-protocol

---

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

**Syntax** `exclude-protocol protocol-name;`

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

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

**Description** 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.

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.

**Default** Disabled.

All packets are secured on a link when MACsec is enabled, with the exception of all types of Spanning Tree Protocol (STP) packets.

**Options** *protocol-name* —Specifies the name of the protocol that should not be MACsec-secured. Options include:

- **cdp** —Cisco Discovery Protocol.
- **lACP** —Link Aggregation Control Protocol.
- **lldp** —Link Level Discovery Protocol.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## 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.

The fabric is the data link between the nodes and is used to forward traffic between the chassis. Each node in the chassis requires a fabric interface configured. To create a fabric link between the two chassis requires the creation of a special interface called the fab interface. Node 0's fabric interface, called fab0, and node 1's fabric interface, called fab1.

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.

```
fab0 {
  fabric-options {
    member-interfaces {
      ge-0/0/4;
    }
  }
}
fab1 {
  fabric-options {
    member-interfaces {
      ge-2/0/4;
    }
  }
}
```



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

<b>Required Privilege</b>	interface—To view this statement in the configuration.
<b>Level</b>	interface-control—To add this statement to the configuration.
<b>Related Documentation</b>	<ul style="list-style-type: none"><li>• <a href="#">Example: Configuring the Chassis Cluster Fabric Interfaces on page 125</a></li></ul>



## gigether-options (Chassis Cluster)

Supported Platforms [SRX Series, vSRX](#)

```
Syntax  gigether-options {
        802.3ad {
            backup | primary | bundle;
            lacp {
                port-priority priority;
            }
        }
        auto-negotiation {
            remote-fault {
                local-interface-offline | local-interface-online;
            }
        }
        no-auto-negotiation;
        ethernet-switch-profile {
            mac-learn-enable;
            tag-protocol-id [tpids];
            ethernet-policer-profile {
                input-priority-map {
                    ieee802.1p {
                        premium [values];
                    }
                }
                output-priority-map {
                    classifier {
                        premium {
                            forwarding-class class-name {
                                loss-priority (high | low);
                            }
                        }
                    }
                }
            }
            policer cos-policer-name {
                aggregate {
                    bandwidth-limit bps;
                    burst-size-limit bytes;
                }
                premium {
                    bandwidth-limit bps;
                    burst-size-limit bytes;
                }
            }
        }
        flow-control | no-flow-control;
        ieee-802-3az-eee;
        ignore-l3-incompletes;
        loopback | no-loopback;
        mpls {
            pop-all-labels {
                required-depth (1 | 2);
            }
        }
    }
```

```
    }  
  }  
  redundant-parent (Interfaces Gigabit Ethernet) interface-name;  
  source-address-filter {  
    mac-address;  
  }  
}
```

**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**

- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 145](#)

## global-threshold

---

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

**Syntax** `global-threshold number;`

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

**Release Information** Statement introduced in Junos OS Release 10.1.

**Description** Define global threshold for IP monitoring. This is the number that needs to be met or exceeded by all of the cumulative weights of the monitored IP addresses to trigger a failover.

When a monitored address is marked as unreachable, the weight value associated with that address is deducted from the the redundancy group IP address monitoring global threshold. If the accumulated monitored address weight values surpass the global-threshold value, that is, when the global threshold reaches 0, the global weight is deducted from the redundancy group threshold. Every redundancy group has a default threshold of 255. If the threshold reaches 0, a failover is triggered

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

**Range:** 0 through 255

**Default:** 0

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

**Related Documentation**

- [ip-monitoring on page 445](#)

## global-weight

---

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

**Syntax** `global-weight number;`

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

**Release Information** Statement introduced in Junos OS Release 10.1.

**Description** Define global weight for IP monitoring. This is the weight that is subtracted from the redundancy group weight for all of the hosts being monitored. This number specifies the relative importance of IP address monitored objects in the operation of the redundancy group.

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

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

**Range:** 0 through 255

**Default:** 255

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

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

**Related Documentation**

- [ip-monitoring on page 445](#)

## gratuitous-arp-count

---

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

**Syntax** `gratuitous-arp-count number;`

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

**Release Information** Statement introduced in Junos OS Release 9.0.

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

You can configure this option to 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.

By default, the SRX series device sends four GARPs per reth on a failover. You can modify the number of GARPs sent per-redundancy-group basis.

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

**Range:** 1 through 16

**Default:** 4

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

**Related Documentation** • [redundancy-group \(Chassis Cluster\) on page 464](#)

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

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

## heartbeat-threshold

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

**Syntax** `heartbeat-threshold number;`

**Hierarchy Level** `[edit chassis cluster]`

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

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

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

For example, a **heartbeat-threshold** of 3 and a **heartbeat-interval** of 1000 milliseconds result in a total wait of 3 seconds before failover is triggered. To increase this wait to 5 seconds, you could increase the **heartbeat-threshold**, the **heartbeat-interval**, or both. A **heartbeat-threshold** of 5 and a **heartbeat-interval** of 1000 milliseconds would yield a wait time of 5 seconds. Setting the **heartbeat-threshold** to 4 and the **heartbeat-interval** to 1250 milliseconds would also yield a wait time of 5 seconds.

**Options** *number*—Number of consecutive missed heartbeats.

**Range:** 3 through 8

**Default:** 3

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

## hold-down-interval

---

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

**Syntax** hold-down-interval *number*;

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

**Release Information** Statement introduced in Junos OS Release 10.0.

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

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

For some configurations, such as ones with a large number of routes or logical interfaces, the default or specified interval for redundancy group 0 might not be sufficient. In such cases, the system automatically extends the dampening time in increments of 60 seconds until the system is ready for failover.

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

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

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

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

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)



## include-sci

---

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

**Syntax** `include-sci;`

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

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

**Description** Specify that the SCI tag be appended to each packet on a link that has enabled MACsec.

You must enable SCI tagging on a switch that is enabling MACsec on an Ethernet link connecting to an SRX device.

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.

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.

**Default** SCI tagging is enabled on an SRX device that have enabled MACsec using static connectivity association key (CAK) security mode, by default.

SCI tagging is disabled on all other interfaces, 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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## interface (Chassis Cluster)

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

**Syntax** *logical-interface-name* secondary-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**

- ***logical-interface-name***—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 423](#)

## interfaces (MACsec)

---

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

**Syntax** *interface-name* {  
    connectivity-association *connectivity-association-name*;  
}

**Hierarchy Level** [edit security macsec cluster-data-port]

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

**Description** Specify chassis cluster fabric interface on which MACsec is enabled. For SRX340, and SRX345 devices, the fabric interface can be any 1 G Ethernet interface. Use this configuration to apply a connectivity association to an interface, which enables Media Access Control Security (MACsec) on that interface.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## 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.

For a redundancy group to automatically failover to another node, its interfaces must be monitored. Interface monitoring monitors the physical status of an interface.

When you configure a redundancy group, you can specify a set of interfaces that the redundancy group is to monitor for status (or "health") of interface 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 assign a weight to the interface. 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, the redundancy group fails over to the other node in the cluster.

**Options** **name**—Name of the interface to monitor

**weight**—Weight assigned to this interface that influences failover

**Range:** 0-255

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

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

## 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** *IPv4 address*—The address to be continually monitored for reachability. You also set up a secondary IP address to allow testing from the secondary node.



**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).

**interface *interface* secondary-ip-address *ip-address*;**—Define source address for monitoring packets on secondary link.

**global-threshold**—Define global threshold for IP monitoring. See [global-threshold](#).

**Default:** 0

**Range:** 0-255

**global-weight**—Define global weight for IP monitoring. See [global-weight](#).

**Default:** 255

**Range:** 0-255

**retry-count**—Number of retries needed to declare reachability failure. See [retry-count \(Chassis Cluster\)](#).

**Default:** 5

**Range:** 5-15

**retry-interval**—Define the time interval in seconds between retries. See [retry-interval \(Chassis Cluster\)](#).

**Default:** 1

**Range:** 1-30

The remaining statements are explained separately. See [CLI Explorer](#).

<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 Redundancy Group IP Address Monitoring on page 235</a></li></ul>
------------------------------	---

## key-server-priority (MACsec)

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

**Syntax** `key-server-priority priority-number;`

**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 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.

The switch with the lower *priority-number* is selected as the key server.

If the *priority-number* 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.

**Default** The default key server priority number is 16.

**Options** *priority-number* —Specifies the MKA server election priority number.

The *priority-number* can be any number between 0 and 255. The lower the number, the higher the priority.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## 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*
  - *periodic (Interfaces)*



## link-protection (Chassis Cluster)

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

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

**Hierarchy Level** [edit chassis aggregated-devices ethernet lacp]

**Release Information** Statement introduced in Junos OS Release 10.2.

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

By default LACP link protection reverts to a higher-priority (lower-numbered) link when the higher-priority link becomes operational or when a higher-priority link is added to the aggregated Ethernet bundle.

You can suppress link calculation by adding the non-revertive statement to the link protection configuration. In nonrevertive mode, when a link is active in sending and receiving LACP packets, adding a higher-priority link to the bundle does not change the status of the currently active link. It remains active.

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

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)
- *Example: Configuring Aggregated Ethernet Device with LAG and LACP (CLI Procedure)*

## macsec

**Supported Platforms** SRX340, SRX345

**Syntax**

```
macsec {
  cluster-control-port <idx> {
    connectivity-association connectivity-association-name;
  }
  cluster-data-port interface-name {
    connectivity-association connectivity-association-name;
  }
  connectivity-association connectivity-association-name {
    exclude-protocol protocol-name;
    include-sci;
    mka {
      key-server-priority priority-number;
      must-secure;
      transmit-interval milliseconds;
    }
    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;
  }
  traceoptions {
    file {
      filename;
      files number;
      match regular-expression;
      (world-readable | no-world-readable);
      size maximum-file-size;
    }
    flag flag;
  }
}
```

**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).

**Options** **cluster-control-port <idx>**—Specify chassis cluster control interface on which MACsec is enabled.  
**Values:** 0.

**cluster-data-port *interface-name***—Specify chassis cluster fabric interface on which MACsec is enabled.

**connectivity-association**—Create or configure a MACsec connectivity association.

**traceoptions**—Define MACsec configuration tracing operations.

The remaining statements are explained separately. See [CLI Explorer](#).

**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 373](#)
- [Understanding Media Access Control Security \(MACsec\) on page 370](#)

## mka

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

**Syntax**

```
mka {
    must-secure;
    key-server-priority priority-number;
    transmit-interval interval;
}
```

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

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

**Description** Specify parameters for the MACsec Key Agreement (MKA) protocol.

**Options** The remaining statements are explained separately.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## must-secure

---

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

**Syntax** `must-secure;`

**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 that all traffic traversing the MACsec-secured link must be forwarded onward.

When the **must-secure** is enabled, all traffic that is not MACsec-secured that is received on the interface is dropped.

When the **must-secure** 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.

The **must-secure** 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 does not support MACsec can continue to send and receive traffic over the network—provided the **must-secure** 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.

**Default** The **must-secure** option is disabled.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## network-management

---

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

**Syntax**

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

**Hierarchy Level** [edit chassis cluster]

**Release Information** Statement introduced in Junos OS Release 11.1.

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

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

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

## 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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## node (Chassis Cluster Redundancy Group)

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

**Syntax** `node ( 0 | 1 ) {  
    priority number;  
}`

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

**Release Information** Statement introduced in Junos OS Release 9.0.

**Description** Identify each cluster node in a redundancy group and set its relative priority for mastership.

A redundancy group is 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. The primacy of a redundancy group is dependent on the priority configured for the node, the node ID (in case of tied priorities), and the order in which the node comes up.

**Options** **node**—Cluster node number, You can set with the `set chassis cluster node node-number` statement.

**Values:**

- 0—Node identifier 0
- 1—Node identifier 1

**priority *number***—Priority value of the node. Each node is given a priority within a redundancy group. The eligible node with the highest priority is elected master. Initiating a failover with the `request chassis cluster failover node` or `request chassis cluster failover redundancy-group` command overrides the priority settings.

**Range:** 1-254

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

**Related Documentation**

- [redundancy-group \(Chassis Cluster\) on page 464](#)

## ntp

---

**Supported Platforms** [ACX Series](#), [EX Series](#), [M Series](#), [MX Series](#), [PTX Series](#), [SRX Series](#), [T Series](#)

**Syntax** ntp {  
    authentication-key *number* type *type* value *password*;  
    boot-server *address*;  
    broadcast <*address*> <key *key-number*> <routing-instance-name *routing-instance-name*>  
        <version *value*> <ttl *value*>;  
    broadcast-client;  
    multicast-client <*address*>;  
    peer *address* <key *key-number*> <version *value*> <prefer>;  
    server *address* <key *key-number*> <version *value*> <prefer>;  
    source-address *source-address* <routing-instance *routing-instance-name*>;  
    trusted-key [ *key-numbers* ];  
}

**Hierarchy Level** [edit system]

**Release Information** Statement introduced before Junos OS Release 7.4.  
Statement introduced in Junos OS Release 9.0 for EX Series switches.

**Description** Configure NTP on the router or switch.  
  
The remaining statements are explained separately.

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

**Related Documentation**

- [Synchronizing and Coordinating Time Distribution Using NTP](#)
- [NTP Time Synchronization on SRX Series Devices on page 291](#)



## ntp threshold

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

**Syntax** threshold action (accept | reject) *value*;

**Hierarchy Level** [edit system ntp]

**Release Information** Statement introduced in Junos OS Release 15.1X49-D70.

**Description** Assign a threshold value for Network Time Protocol (NTP) adjustment that is outside of the acceptable NTP update and specify whether to accept or reject NTP synchronization when the proposed time from the NTP server exceeds the configured threshold value. If **accept** is the specified action, the system synchronizes the device time with the NTP server, but logs the time difference between the configured threshold and the time proposed by the NTP server; if **reject** is the specified action, synchronization with the time proposed by the NTP server is rejected, but the system provides the option of manually synchronizing the device time with the time proposed by the NTP server and logs the time difference between the configured threshold and the time proposed by the NTP server. By logging the time difference and rejecting synchronization when the configured threshold is exceeded, this feature helps improve security on the NTP service.

**Options** *value*—Specify the maximum value in seconds allowed for NTP adjustment.

**Range:** 1 through 600.

**Default:** The default value is 400.

**action**—Specify the actions for NTP abnormal adjustment.

- **accept**—Enable log mode for abnormal NTP adjustment. When the proposed time from the NTP server is outside of the configured threshold value, the device time synchronizes with the NTP server, but the system logs the time difference between the configured threshold and the time proposed by the NTP server.
- **reject**—Enable log and reject mode for abnormal NTP adjustment. When the proposed time from the NTP server is outside of the configured threshold value, the system rejects synchronization, but provides the option for manually synchronizing the time and logs the time difference between the configured threshold and the time proposed by the NTP server.

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

**Related Documentation**

- [ntp on page 456](#)
- [set date ntp on page 582](#)
- [show system ntp threshold on page 598](#)

- [NTP Time Synchronization on SRX Series Devices on page 291](#)

## 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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## preempt (Chassis Cluster)

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

**Syntax**

```
preempt {
    delay seconds;
    limit limit;
    period seconds;
}
```

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

**Release Information** Statement introduced in Junos OS Release 9.0. Support for **delay**, **limit**, and **period** options are added in Junos OS Release 17.4R1.

**Description** Allow preemption of the primary system based on the priority within a redundancy group.

By configuring the preemptive delay timer and failover limit, you can limit the flapping of the redundancy group state between the secondary and the primary in a preemptive failover.

By default, preemption is disabled.

Example: Consider the following scenario where you have configured a preemptive period as 300 seconds and preemptive limit as 50.

When the preemptive limit is configured as 50, the count starts at 0 and increments with a first preemptive failover; this process continues until the count reaches the configured preemptive limit, that is 50, before the preemptive period expires. When the preemptive limit (50) is exceeded, you must manually reset the preempt count to allow preemptive failovers to occur again.

When you have configured the preemptive period as 300 seconds, and if the time difference between the first preemptive failover and the current failover has already exceeded 300 seconds, and the preemptive limit (50) is not yet reached, then the preemptive period will be reset. After resetting, the last failover is considered as the first preemptive failover of the new preemptive period and the process starts all over again.

**Options** **delay**—Time to wait before the node in secondary state transitions to primary state in a preemptive failover.

**Range:** 1 to 21,600 seconds

**Default:** 1

**limit**— Maximum number of preemptive failovers allowed in a configured preemptive period.

**Range:** 1 to 50

**period**—Time period during which the preemptive limit is applied.

**Range:** 1 to 1400 seconds

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

**Related Documentation**

- [redundancy-group \(Chassis Cluster\) on page 464](#)
- [Understanding Chassis Cluster Redundancy Group Failover on page 249](#)

---

## pre-shared-key

---

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

**Syntax**

```
pre-shared-key {  
    cak hexadecimal-number;  
    ckn hexadecimal-number;  
}
```

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

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

**Description** Specifies the pre-shared key used to enable MACsec using static connectivity association key (CAK) security mode.

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.

**Default** No pre-shared keys exist, by default.

**Options** The remaining statements are explained separately.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## priority (Protocols VRRP)

<b>Supported Platforms</b>	EX Series, M Series, NFX Series, OCX1100, QFabric System, QFX Series, SRX Series, T Series
<b>Syntax</b>	<code>priority <i>priority</i>;</code>
<b>Hierarchy Level</b>	<p>[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family inet address <i>address</i> vrrp-group <i>group-id</i>],</p> <p>[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family inet6 address <i>address</i> vrrp-inet6-group <i>group-id</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family inet address <i>address</i> vrrp-group <i>group-id</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family inet6 address <i>address</i> vrrp-inet6-group <i>group-id</i>]</p>
<b>Release Information</b>	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS 11.3 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1x53-D20 for the OCX Series.</p> <p>Statement introduced in Junos OS Release 18.1R1 for the SRX Series devices.</p>
<b>Description</b>	Configure a Virtual Router Redundancy Protocol (VRRP) device's priority for becoming the master default device. The device with the highest priority within the group becomes the master.
<b>Options</b>	<p><b>priority</b>—Device's priority for being elected to be the master device in the VRRP group. A larger value indicates a higher priority for being elected.</p> <p><b>Range:</b> 1 through 255</p> <p><b>Default:</b> 100. If two or more devices have the highest priority in the VRRP group, the device with the VRRP interface that has the highest IP address becomes the master, and the others serve as backups.</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="#">Configuring Basic VRRP Support</a></li> <li>• <a href="#">Understanding VRRP on SRX Series Devices on page 279</a></li> <li>• <a href="#">Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces on page 282</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 {
        delay seconds;
        limit limit;
        period seconds;
    }
}

```

**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.



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

**Range:** 0 through 128

**interface-monitor**—Specify a redundancy group interface to be monitored for failover and the relative weight of the interface.

**ip-monitor**—Specify IP address of interface to be monitored for end-to-end connectivity.

**gratuitous-arp-count**—Number of gratuitous ARPs to send on an active interface after failover

**Range:** 1-16

**hold-down-interval**—RG failover interval. RG0(300-1800) RG1+(0-1800) (seconds)

**Range:** 0-1800

**node**—Identify each cluster node in a redundancy group and set its relative priority for mastership.

**preempt**—Allow preemption of the primary system based on the priority within a redundancy group.

The remaining statements are explained separately. See [CLI Explorer](#).

<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 445</a></li> </ul>
------------------------------	---

## redundant-ether-options

---

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

**Syntax** `redundant-ether-options {  
 (flow-control | no-flow-control);  
 lacp {  
 (active | passive);  
 periodic (fast | slow);  
 }  
 link-speed speed;  
 (loopback | no-loopback);  
 minimum-links number;  
 redundancy-group number;  
 source-address-filter mac-address;  
 (source-filtering | no-source-filtering);  
}`

**Hierarchy Level** `[edit interfaces interface-name]`

**Release Information** Statement introduced in Junos OS Release 9.2.

**Description** Configure Ethernet redundancy options for a chassis cluster.

**Options** The remaining statements are explained separately. See [CLI Explorer](#).

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

**Related Documentation**

- [Example: Enabling Eight-Queue Class of Service on Redundant Ethernet Interfaces on SRX Series Devices in a Chassis Cluster on page 168](#)
- [Example: Configuring Chassis Cluster Redundant Ethernet Interfaces on page 145](#)

---

## redundant-parent (Interfaces)

---

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

**Syntax** `redundant-parent redundant-ethernet-interface-name;`

**Hierarchy Level** `[edit interfaces interface-name gigeether-options]`  
`[edit interfaces interface-name fastether-options]`

**Release Information** Statement introduced in Junos OS Release 10.2.

**Description** Specify redundant Ethernet interfaces (reth) and assign local (child) interfaces to the 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.

**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 on page 145](#)

## redundant-pseudo-interface-options

---

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

**Syntax**

```
redundant-pseudo-interface-options {  
    redundancy-group redundancy-group-number;  
}
```

**Hierarchy Level** [edit interfaces lo0]

**Release Information** Statement introduced in Junos OS Release 12.1X44-D10.

**Description** Configure the loopback pseudointerface in a redundancy group.

Redundancy groups are used to bundle interfaces into a group for failover purpose in a chassis cluster setup. You can configure a loopback interface as an alternative physical interface to reach the peer gateway. Loopback interfaces can be configured on any redundancy group.

For example: An Internet Key Exchange (IKE) gateway operating in chassis cluster, needs an external interface to communicate with a peer device. When an external interface (a reth interface or a standalone interface) is used for communication; the interface might go down when the physical interfaces are down. Instead, use loopback interfaces as an alternative to physical interfaces.

**Options** *redundancy-group-number*— Configure the redundancy group number.

**Range:** 0 through 255

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

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

**Related Documentation**

- *Understanding the Loopback Interface for a High Availability VPN*

---

## replay-protect

---

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

**Syntax**

```
replay-protect {  
    replay-window-size number-of-packets;  
}
```

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

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

**Description** Enable replay protection for MACsec.

A replay window size specified using the **replay-window-size***number-of-packets* statement must be specified to enable replay protection.

**Options** The remaining statements are explained separately.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## replay-window-size

---

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

**Syntax** `replay-window-size number-of-packets;`

**Hierarchy Level** [edit security macsec connectivity-association replay-protect]

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

**Description** Specifies the size of the replay protection window.

This statement has to be configured to enable replay protection.

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.

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.

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.

**Default** Replay protection is disabled.

**Options** *number-of-packets* —Specifies the size of the replay protection window, in packets.

When this variable is set to 0, all packets that arrive out-of-order are dropped. The maximum out-of-order number-of-packets that can be configured is 65535.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)

- [macsec on page 450](#)

## reth-count (Chassis Cluster)

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

**Syntax** `reth-count number;`

**Hierarchy Level** [edit chassis cluster]

**Release Information** Statement introduced in Junos OS Release 9.0.

**Description** Specify the number of redundant Ethernet (**reth**) interfaces allowed in the chassis cluster. You need to specify the total number of interfaces in the chassis cluster before redundant Ethernet interfaces are created. For example, the **set chassis cluster reth-count 2** allow you to create two reth interfaces (example: reth0 and reth1)

Note that the number of **reth** interfaces configured determines the number of redundancy groups that can be configured and each SRX series device has a maximum number of reths that it can support.

**Options** *number* —Number of redundant Ethernet interfaces allowed.  
**Range:** 1 through 128  
**Default:** 0

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

**Related Documentation** • [cluster \(Chassis\) on page 423](#)

## 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 in IP address monitoring.

In the IP address monitoring, 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. In this configuration, the retry interval determines the ping interval for each IP address monitored by the redundancy group and the retry count sets the number of allowed consecutive ping failures for each IP address monitored by the redundancy group.

(See [retry-interval \(Chassis Cluster\)](#) 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.

**Default:** 5

**Range:** 5-15

**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 423](#)



## retry-interval (Chassis Cluster)

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

**Syntax** `retry-interval interval;`

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

**Release Information** Statement introduced in Junos OS Release 10.1.

**Description** Specify the ping packet send frequency (in seconds) for each IP address monitored by the redundancy group.

In the IP address monitoring, 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. In this configuration, the retry interval determines the ping interval for each IP address monitored by the redundancy group and the retry count sets the number of allowed consecutive ping failures for each IP address monitored by the redundancy group.

(See [retry-count \(Chassis Cluster\)](#) for a related IP address monitoring configuration variable.)

**Options** *interval*—Pause time (in seconds) between each ping sent to each IP address monitored by the redundancy group.

**Default:** 1

**Range:** 1-30

**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 445](#)

## route-active-on

---

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

**Syntax** route-active-on (node0 | node1);

**Hierarchy Level** [edit policy-options condition *condition-name*]

**Release Information** Statement introduced in Junos OS Release 9.0.

**Description** For chassis cluster configurations, identify the device (node) on which a route is active on a defined policy condition.

**Options** **node0**—Route active on node 0  
**node1**—Route active on node 1

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

## security-mode

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

**Syntax** `security-mode security-mode;`

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

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

**Description** Configure the MACsec security mode for the connectivity association.

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.

**Options** ***security-mode*** —Specifies the MACsec security mode. Options include:

- **dynamic**—Dynamic mode.

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.

- **static-cak** —Static connectivity association key (CAK) mode.

Static CAK security mode is used to enable MACsec on switch-to-switch Ethernet links. In **static-cak** 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.

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## 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.

A minimum traceoptions configuration must include both a target file and a flag. The target file determines where the trace output is recorded. The flag defines what type of data is collected.

- 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

<b>Required Privilege Level</b>	trace—To view this statement in the configuration. trace-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 423</a></li> </ul>
------------------------------	---

## 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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)

## virtual-address

<b>Supported Platforms</b>	EX Series, M Series, MX Series, NFX Series, OCX1100, QFabric System, QFX Series, SRX Series, T Series
<b>Syntax</b>	virtual-address [ <i>addresses</i> ];
<b>Hierarchy Level</b>	[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family inet address <i>address</i> vrrp-group <i>group-id</i> ], [edit logical-systems <i>logical-system-name</i> interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family inet address <i>address</i> vrrp-group <i>group-id</i> ]
<b>Release Information</b>	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS 11.3 for the QFX Series. Statement introduced in Junos OS Release 14.1x53-D20 for the OCX Series. Statement introduced in Junos OS Release 18.1R1 for the SRX Series devices.
<b>Description</b>	Configure the addresses of the devices in a Virtual Router Redundancy Protocol (VRRP) IPv4 or IPv6 group. You can configure up to eight addresses.
<b>Options</b>	<b>addresses</b> —Addresses of one or more devices. Do not include a prefix length. If the address is the same as the interface's physical address, the interface becomes the master device for the group.
<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="#">Configuring Basic VRRP Support</a></li> <li>• <a href="#">Understanding VRRP on SRX Series Devices on page 279</a></li> <li>• <a href="#">Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces on page 282</a></li> </ul>

## vrrp-group

Supported Platforms [SRX Series](#)

**Syntax** `vrrp-group group-id {  
 (accept-data | no-accept-data);  
 advertise-interval seconds;  
 advertisements-threshold number;  
 authentication-key key;  
 authentication-type (md5 | simple);  
 fast-interval milliseconds;  
 no-accept-data;  
 no-preempt;  
 preempt {  
 hold-time seconds;  
 }  
 preferred;  
 priority number;  
 track {  
 interface name {  
 bandwidth-threshold bits-per-second priority-cost priority {  
 priority-cost priority;  
 }  
 }  
 priority-hold-time seconds;  
 route route_address {  
 routing-instance;  
 }  
 }  
 virtual-address virtual-link-local-address;  
 vrrp-inherit-from {  
 active-group active-group;  
 active-interface active-interface;  
 }  
}`

**Hierarchy Level** [edit interfaces *name* unit *name* family inet address],  
 [edit interfaces *name* unit *name* family inet inet6 address]

**Release Information** Statement introduced before Junos OS Release 18.1R1 for SRX Series devices.

**Description** Virtual Router Redundancy Protocol group

**Options** **group-id**—VRRP group identifier. If you enable MAC source address filtering on the interface, you must include the virtual MAC address in the list of source MAC addresses that you specify in the source-address-filter statement. MAC addresses ranging from 00:00:5e:00:53:00 through 00:00:5e:00:53:ff are reserved for VRRP, as defined in RFC 2338. The VRRP group number must be the decimal equivalent of the last hexadecimal byte of the virtual MAC address.

**Range:** 0 through 255



**accept-data**—Accept packets destined for virtual IP address

**advertise-interval**—Advertisement interval (seconds)

**Range:** 1 through 255

**inet6-advertise-interval**—Inet6 advertisement interval (milliseconds)

**Range:** 100-40000

**advertisements-threshold**—Number of vrrp advertisements missed before declaring master down

**Range:** 1 through 15

**authentication-key**—Authentication key

**authentication-type**—Authentication type

**Values:**

- **md5**—HMAC-MD5-96
- **simple**—Simple password

**fast-interval**—Fast advertisement interval (milliseconds)

**Range:** 10 through 40950

**no-accept-data**—Don't accept packets destined for virtual IP address

**no-preempt**—Don't allow preemption

**preempt**—Allow preemption

**preferred**—Preferred group on subnet

**priority**—Device election priority

**Range:** 0 through 255

**track**—Interfaces to track for VRRP group

**virtual-address**—One or more virtual IPv4 addresses

**virtual-inet6-address**—One or more virtual inet6 addresses

**vrrp-inherit-from**—VRRP group to follow for the vrrp-group or vrrp-inet6-group

The remaining statements are explained separately. See [CLI Explorer](#).

**Required Privilege Level** interface

**Related Documentation**

- [Understanding VRRP on SRX Series Devices on page 279](#)
- [Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces on page 282](#)

## weight

---

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

**Syntax** `weight number;`

**Hierarchy Level** `[edit chassis cluster redundancy-group group-number interface-monitor interface ]`  
`[edit chassis cluster redundancy-group group-number ip-monitoring IP-address]`

**Release Information** Statement modified in Junos OS Release 10.1.

**Description** Specify the relative importance of the object to the operation of the redundancy group. This statement is primarily used with interface monitoring and IP address monitoring objects. The failure of an object—such as an interface—with a greater weight brings the group closer to failover. Every monitored object is assigned a weight.

- interface-monitor objects—If the object fails, its weight is deducted from the threshold of its redundancy group;
- ip-monitoring objects—If a monitored IP address becomes unreachable for any reason, the weight assigned to that monitored IP address is deducted from the redundancy group's global-threshold for IP address monitoring.

Every redundancy group has a default threshold of 255. If the threshold reaches 0, a failover is triggered. Failover is triggered even if the redundancy group is in manual failover mode and preemption is not enabled.

**Options** *number*—Weight assigned to the interface or monitored IP address. A higher weight value indicates a greater importance.

**Range:** 0 through 255

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

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

## CHAPTER 7

# 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 cb
- 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 chassis cluster in-service-upgrade abort (ISSU)
- request security internal-security-association refresh
- request system scripts add
- request system reboot
- request system software in-service-upgrade (Maintenance)
- request system software rollback (SRX Series)
- set chassis cluster cluster-id node node-number 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 information issu
- 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 environment cb`
- `show chassis ethernet-switch`
- `show chassis fabric plane`
- `show chassis fabric plane-location`
- `show chassis fabric summary`
- `show chassis hardware (View)`
- `show chassis routing-engine (View)`
- `show configuration chassis cluster traceoptions`
- `set date ntp`
- `show interfaces (Gigabit Ethernet)`
- `show system ntp threshold`
- `show security macsec connections`
- `show security macsec statistics (SRX Series Devices)`
- `show security mka statistics`
- `show security mka sessions (SRX Series Device)`
- `show security internal-security-association`
- `show system license (View)`
- `show vrrp`

## clear chassis cluster control-plane statistics

---

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

**Syntax** `clear chassis cluster control-plane statistics`

**Release Information** Command introduced in Junos OS Release 9.3.

**Description** Clear the control plane statistics of a chassis cluster.

**Required Privilege Level** clear

**Related Documentation**

- [show chassis cluster control-plane statistics on page 512](#)

**List of Sample Output** [clear chassis cluster control-plane statistics on page 485](#)

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

### Sample Output

`clear chassis cluster control-plane statistics`

```
user@host> clear chassis cluster control-plane statistics
Cleared control-plane statistics
```

## clear chassis cluster data-plane statistics

---

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

**Syntax** `clear chassis cluster data-plane statistics`

**Release Information** Command introduced in Junos OS Release 9.3.

**Description** Clear the data plane statistics of a chassis cluster.

**Required Privilege Level** clear

**Related Documentation**

- [show chassis cluster data-plane statistics on page 515](#)

**List of Sample Output** [clear chassis cluster data-plane statistics on page 486](#)

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

### Sample Output

`clear chassis cluster data-plane statistics`

```
user@host> clear chassis cluster data-plane statistics
Cleared data-plane statistics
```

## clear chassis cluster failover-count

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

**Syntax** `clear chassis cluster failover-count`

**Release Information** Command introduced in Junos OS Release 9.3.

**Description** Clear the failover count of all redundancy groups.

**Required Privilege Level** clear

**Related Documentation**

- [request chassis cluster failover node on page 494](#)
- [request chassis cluster failover reset on page 497](#)
- [show chassis cluster status on page 542](#)

**List of Sample Output**

- [show chassis cluster status on page 487](#)
- [clear chassis cluster failover-count on page 487](#)
- [show chassis cluster status on page 488](#)

**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 ip-address on page 490](#)

**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 487](#)
- [clear chassis cluster ip-monitoring failure-count on page 489](#)

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

### Sample Output

```
user@host> clear chassis cluster ip-monitoring failure-count ip-address 1.1.1.1

node0:
-----
Cleared failure count for IP: 1.1.1.1

node1:
-----
Cleared failure count for IP: 1.1.1.1
```

## clear chassis cluster statistics

---

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

**Syntax** `clear chassis cluster statistics`

**Release Information** Command introduced in Junos OS Release 9.3.

**Description** Clear the control plane and data plane statistics of a chassis cluster.

**Required Privilege Level** clear

**Related Documentation**

- [show chassis cluster statistics on page 538](#)

**List of Sample Output** [clear chassis cluster statistics on page 491](#)

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

### Sample Output

#### clear chassis cluster statistics

```
user@host> clear chassis cluster statistics
Cleared control-plane statistics
Cleared data-plane statistics
```

## request chassis cb

---

**Supported Platforms** [PTX10008](#), [SRX Series](#), [vSRX](#)

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

**Release Information** Command introduced in Junos OS Release 9.2.  
Command introduced in Junos OS Release 17.2 for PTX10008 Routers.  
Starting with Junos OS Release 12.1X47-D15, the SRX5K-SCBE (SCB2) is introduced and starting with Junos OS Release 15.1X49-D10, the SRX5K-SCB3 (SCB3) with enhanced midplane is introduced.

**Description** Control the operation (take the CB offline or bring online) of the Control Board (CB).

**Options** **offline**—Take the Control Board offline.  
**online**—Bring the Control Board online.  
**slot *slot-number***—Control Board slot number.

**Required Privilege Level** maintenance

**Related Documentation**

- [show chassis environment cb on page 549](#)

**List of Sample Output** [request chassis cb \(SRX Series\) on page 492](#)  
[request chassis cb \(PTX10008 Router\) on page 492](#)

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

## Sample Output

### request chassis cb (SRX Series)

```
user@host> request chassis cb offline slot 2 node local
node0:
-----
Offline initiated, use "show chassis environment cb" to verify
```

### request chassis cb (PTX10008 Router)

```
user@host> request chassis cb offline slot 1
Offline initiated, use "show chassis environment cb" to verify
```

## request chassis cluster configuration-synchronize

---

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

**Syntax** request chassis cluster configuration-synchronize

**Release Information** Command introduced in Junos OS Release 12.1X47-D10.

**Description** Synchronize 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 166](#)
- [Verifying Chassis Cluster Configuration Synchronization Status on page 167](#)
- [NTP Time Synchronization on SRX Series Devices on page 291](#)

**List of Sample Output** [request chassis cluster configuration-synchronize on page 493](#)

**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 487](#)
  - [request chassis cluster failover reset on page 497](#)
  - [show chassis cluster status on page 542](#)

**List of Sample Output** [request chassis cluster failover node on page 494](#)

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

## Sample Output

### request chassis cluster failover node

```
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** [SRX Series, vSRX](#)

**Syntax** `request chassis cluster failover node node-number 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 or 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 255](#)
  - [Verifying Chassis Cluster Failover Status on page 259](#)

**List of Sample Output** [request chassis cluster failover redundancy-group on page 495](#)

**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

```
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 487](#)
- [request chassis cluster failover node on page 494](#)
- [show chassis cluster status on page 542](#)

**List of Sample Output** [request chassis cluster failover reset on page 497](#)

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

## Sample Output

[request chassis cluster failover reset](#)

```
user@host> request chassis cluster failover reset redundancy-group 0
```

## request chassis fpc

---

**Supported Platforms** [PTX10008, SRX Series](#)

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

**Release Information** Command modified in Junos OS Release 9.2.  
Command introduced in Junos OS Release 17.2 for PTX10008 Routers.

**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 (SRX Series)

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

### request chassis fpc (PTX10008 Router)

```
user@host> request chassis fpc online slot 1
FPC 0 already online
```

## request chassis cluster in-service-upgrade abort (ISSU)

---

**Supported Platforms** [SRX1500, SRX5400, SRX5600, SRX5800](#)

**Syntax** request chassis cluster in-service-upgrade abort

**Release Information** Command introduced in Junos OS Release 11.2.

**Description** Abort an upgrade in a chassis cluster during an in-service software upgrade (ISSU). Use this command to end the ISSU on any nodes in a chassis cluster followed by **reboot** to abort the ISSU on that device.

**Options** This command has no options.

**Required Privilege Level** view

**Related Documentation**

- [request system software in-service-upgrade \(Maintenance\) on page 505](#)

**List of Sample Output** [request chassis cluster in-service-upgrade abort on page 499](#)

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

### Sample Output

#### request chassis cluster in-service-upgrade abort

```
user@host> request chassis cluster in-service-upgrade abort
Exiting in-service-upgrade window
Chassis ISSU Aborted
```

## request security internal-security-association refresh

---

**Supported Platforms** [SRX5400, SRX5600, SRX5800](#)

**Syntax** request security internal-security-association refresh

**Release Information** Command introduced in Junos OS Release 12.1X45-D10.

**Description** Activate internal IPsec so an attacker cannot gain unauthorized information.

**Required Privilege Level** maintenance

**Related Documentation**

- [show security internal-security-association on page 609](#)
- *internal (Security IPsec)*

**List of Sample Output** [request security internal-security-association refresh on page 500](#)

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

### Sample Output

#### request security internal-security-association refresh

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

## request system scripts add

Supported Platforms	<a href="#">SRX Series</a>
Syntax	request system scripts add <i>package-name</i> <no-copy   unlink> <master> <backup>
Release Information	Command introduced before Junos OS Release 9.0. The options <b>master</b> and <b>backup</b> are introduced in Junos OS Release 15.1X49-D50.
Description	CLI command to install AI-Script install packages on SRX Series devices in a chassis cluster.
Options	<p><b>package-name</b>—Specify AI-Script install package name.</p> <p><b>no-copy</b>—(Optional) Do not save a copy of the AI script package file.</p> <p><b>unlink</b>—(Optional) Remove the AI script package after successful installation.</p> <p><b>master</b>—(Optional) Install AI script packages on the primary node.</p> <p><b>backup</b>—(Optional) Install AI script packages on the secondary node.</p>
Additional Information	This command eliminates the AI script installation on both primary node and secondary node separately.
Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none"> <li><a href="#">Understanding Automatic Chassis Cluster Synchronization Between Primary and Secondary Nodes on page 166</a></li> </ul>
List of Sample Output	<a href="#">request system scripts add package-name on page 501</a> <a href="#">request system scripts add package-name on page 502</a>

## Sample Output

### request system scripts add package-name

```
user@host> request system scripts add jais-5.0R1.0-signed.tgz master
```

```
[ : -a: unexpected operator
grep: /etc/db/pkg/jais/+COMMENT: No such file or directory
Installing package '/var/tmp/jais-5.0R4.0-signed.tgz' ...
Verified jais-5.0R4.0.tgz signed by PackageProductionRSA_2016
Adding jais...
Available space: 798414 require: 1814
Installing AI-Scripts version: 5.0R4
```

```

Saving package file in /var/db/scripts/commit/jais-5.0R4.0-signed.tgz ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
ln: ///etc/rc.d/ais: Read-only file system
Model: srx5600
Model: srx5600
Linking in Junos ES manifest file.
Creating srx5800/srx5600 trend data file.
Creating SRX intelligence attachments file.
Creating SRX events attachments file.
Creating AI-Scripts FIFO
Starting AI-Scripts FIFO handler
77834: old priority 0, new priority 20
77842: old priority 0, new priority 20
RSI parameters are now being set
BIOS validation parameter is now being set
BIOS interval parameter is now being set
JMB cleanup age is now being set
JMB Event file is now being set
JMB User Event file is now being set
PHDC collect parameter is now being set
PHDC duration parameter is now being set
PHDC commands file is now being set
JMB Progress Logging parameter is now being set
iJMB generation parameters are now being set
AI-Scripts platform support flag is now being set
Interval event commands file is now being set
Interval event enabled parameter is now being set
All node log collect parameter is now being set
Disk Warning Threshold is now being set
Disk Full Threshold is now being set
RSI Lite Enabled is now being set
Removing any old files that need to be updated
Copying updated files
Restarting eventd ...
Event processing process started, pid 78147
Installation completed
Saving package file in /var/sw/pkg/jais-5.0R4.0-signed.tgz ...
Saving state for rollback ...

```

#### request system scripts add package-name

```

user@host> request system scripts add jais-5.0R1.0-signed.tgz backup
Pushing bundle to node1
[: -a: unexpected operator
grep: /etc/db/pkg/jais/+COMMENT: No such file or directory
Installing package '/var/tmp/jais-5.0R4.0-signed.tgz' ...
Verified jais-5.0R4.0.tgz signed by PackageProductionRSA_2016
Adding jais...
Available space: 2619677 require: 1814
Installing AI-Scripts version: 5.0R4
chmod: /var/db/scripts/event/cron.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event2.slax: No such file or directory
chmod: /var/db/scripts/op/ais_bit.slax: No such file or directory
Saving package file in /var/db/scripts/commit/jais-5.0R4.0-signed.tgz ...
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
ln: ///etc/rc.d/ais: Read-only file system
Mounted jais package on /dev/md2...
Verified manifest signed by PackageProductionRSA_2016

```

```
Verified jais-5.0R4.0 signed by PackageProductionRSA_2016
Model: srx5600
Model: srx5600
Linking in Junos ES manifest file.
Creating srx5800/srx5600 trend data file.
Creating SRX intelligence attachments file.
Creating SRX events attachments file.
Creating AI-Scripts FIFO
Starting AI-Scripts FIFO handler
99423: old priority 0, new priority 20
99428: old priority 0, new priority 20
99429: old priority 0, new priority 20
99430: old priority 0, new priority 20
RSI parameters are now being set
BIOS validation parameter is now being set
BIOS interval parameter is now being set
JMB cleanup age is now being set
JMB Event file is now being set
JMB User Event file is now being set
PHDC collect parameter is now being set
PHDC duration parameter is now being set
PHDC commands file is now being set
JMB Progress Logging parameter is now being set
iJMB generation parameters are now being set
AI-Scripts platform support flag is now being set
Interval event commands file is now being set
Interval event enabled parameter is now being set
All node log collect parameter is now being set
Disk Warning Threshold is now being set
Disk Full Threshold is now being set
RSI Lite Enabled is now being set
chmod: /var/db/scripts/event/cron.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event.slax: No such file or directory
chmod: /var/db/scripts/event/bit_event2.slax: No such file or directory
chmod: /var/db/scripts/op/ais_bit.slax: No such file or directory
Removing any old files that need to be updated
Copying updated files
Restarting eventd ...
Event processing process started, pid 99730
Installation completed
Saving package file in /var/sw/pkg/jais-5.0R4.0-signed.tgz ...
Saving state for rollback ...
```

## 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* (Optional)— Specify the time at which to reboot the device. You can specify time in one of the following ways:
    - *now*— Reboot the device immediately. This is the default.
    - *+minutes*— Reboot the device in the number of minutes from now that you specify.
    - *yymmddhhmm*— Reboot 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*— Reboot 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* (Optional)— Specify the number of minutes from now to reboot the device. This option is a synonym for the *at +minutes* option
  - *media type* (Optional)— Specify the boot device to boot the device from:
    - *disk/internal*— Reboot from the internal media. This is the default.
    - *usb*— Reboot from the USB storage device.
    - *compact flash*— Reboot from the external CompactFlash card.



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

- *message "text"* (Optional)— Provide 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 510](#)



## request system software in-service-upgrade (Maintenance)

**Supported Platforms** SRX1500, SRX300, SRX320, SRX340, SRX345, SRX4100, SRX4200, 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**—Specify the location and name of the software upgrade package to be installed.
  - **no-copy**—(Optional) Install the software upgrade package but do not save the copies of package files.



**NOTE:** This option is not supported on SRX1500 devices.

- **no-sync**—(Optional) Stop 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) Create 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) Disable 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**—(Optional) Reboot 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) Remove the software package after successful installation.

Required Privilege Level	maintenance
Related Documentation	<ul style="list-style-type: none"> <li>• <a href="#">request system software rollback (SRX Series) on page 510</a></li> </ul>
List of Sample Output	<a href="#">request system software in-service-upgrade (SRX300, SRX320, SRX340, SRX345, and SRX550M Devices) on page 507</a> <a href="#">request system software in-service-upgrade (SRX1400) on page 508</a>
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 (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

```

## Sample Output

### request system software in-service-upgrade (SRX1400)

```

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]
```

## request system software rollback (SRX Series)

---

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

**Syntax**   `request system software rollback  
<node-id>`

**Release Information**   Command introduced in Junos OS Release 10.1.  
Command introduced in Junos OS Release 15.1X49-D50 for SRX1500 devices.  
Command introduced in Junos OS Release 17.4R1 for SRX4100 and SRX4200 devices.

**Description**   Revert to the software that was loaded at the last successful **request system software add** command. The FreeBSD 11 Junos OS image provides an option to save a recovery image in an Operation, Administration, and Maintenance (OAM) partition, but that option will save only the Junos OS image, not the Linux image. If a user saves the Junos OS image and recovers it later, it might not be compatible with the Linux software loaded on the system.

**Options**   *node-id*—Identification number of the chassis cluster node. It can be 0 or 1.

**Required Privilege Level**   maintenance

**Related Documentation**   • [request system reboot on page 504](#)

## set chassis cluster cluster-id node node-number reboot

**Supported Platforms** SRX Series, vSRX

**Syntax** set chassis cluster cluster-id *cluster-id* node *node-number* reboot

**Release Information** Support for extended cluster identifiers (more than 15 identifiers) added in Junos OS Release 12.1X45-D10.

**Description** 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 through 1

**Required Privilege Level** maintenance

**Related Documentation**

- [Example: Setting the Chassis Cluster Node ID and Cluster ID on page 110](#)
- [Understanding the Interconnect Logical System and Logical Tunnel Interfaces](#)
- [Example: Configuring Logical Systems in an Active/Passive Chassis Cluster \(Master Administrators Only\)](#)

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

## show chassis cluster control-plane statistics

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

**Syntax** `show chassis cluster control-plane statistics`

**Release Information** Command introduced in Junos OS Release 9.3. Output changed to support dual control ports in Junos OS Release 10.0.

**Description** Display information about chassis cluster control plane statistics.

**Required Privilege Level** view

**Related Documentation**

- [clear chassis cluster control-plane statistics on page 485](#)

**List of Sample Output** [show chassis cluster control-plane statistics on page 513](#)  
[show chassis cluster control-plane statistics \(SRX5000 Line Devices\) on page 513](#)

**Output Fields** [Table 43 on page 512](#) lists the output fields for the **show chassis cluster control-plane statistics** command. Output fields are listed in the approximate order in which they appear.

*Table 43: show chassis cluster control-plane 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 (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>
<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>
<b>Switch fabric link statistics</b>	<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 rcv 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: 2061982
    Heartbeat packets received: 2060367
    Heartbeat packet errors: 0
  Control link 1:
    Heartbeat packets sent: 2061982
    Heartbeat packets received: 0
    Heartbeat packet errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 3844342
    Probes received: 3843841
  Child link 1
    Probes sent: 0
    Probes received: 0
```

## show chassis cluster data-plane interfaces

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

**Syntax** `show chassis cluster data-plane interfaces`

**Release Information** Command introduced in Junos OS Release 10.2.

**Description** Display the status of the data plane interface (also known as a fabric interface) in a chassis cluster configuration.

**Required Privilege Level** view

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

**List of Sample Output** [show chassis cluster data-plane interfaces on page 514](#)

**Output Fields** [Table 44 on page 514](#) lists the output fields for the `show chassis cluster data-plane interfaces` command. Output fields are listed in the approximate order in which they appear.

*Table 44: 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

---

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

**Syntax** `show chassis cluster data-plane statistics`

**Release Information** Command introduced in Junos OS Release 9.3.

**Description** Display information about chassis cluster data plane statistics.

**Required Privilege Level** view

**Related Documentation**

- [clear chassis cluster data-plane statistics on page 486](#)

**List of Sample Output** [show chassis cluster data-plane statistics on page 516](#)

**Output Fields** [Table 45 on page 516](#) 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 45: 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, SRX300, SRX320, SRX340, SRX345, 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 interfaces) in a chassis cluster.

**Required Privilege Level** view

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)
- [Ethernet Switching Feature Guide](#)

**List of Sample Output** [show chassis cluster ethernet-switching interfaces on page 518](#)

**Output Fields** [Table 46 on page 518](#) 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 46: show chassis cluster ethernet-switching interfaces Output Fields*

Field Name	Field Description
<b>swfab switch fabric interface-name</b>	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 switch fabric 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-7/0/9       up
  ge-7/0/10      up

```

## show chassis cluster ethernet-switching status

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

**Syntax** show chassis cluster ethernet-switching status

**Release Information** Command introduced in Junos OS Release 11.1.

**Description** Display the Ethernet switching status of the chassis cluster.

**Required Privilege Level** view

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)
- *Ethernet Switching Feature Guide*

**List of Sample Output** [show chassis cluster ethernet-switching status on page 520](#)

**Output Fields** [Table 47 on page 519](#) 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 47: show chassis cluster ethernet-switching status Output Fields*

Field Name	Field Description
<b>Cluster ID</b>	ID number (1-255) of a cluster. 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.
<b>Redundancy-Group</b>	You can create up to 128 redundancy groups 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.
<b>Status</b>	State of the redundancy group ( <b>Primary</b> , <b>Secondary</b> , <b>Lost</b> , or <b>Unavailable</b> ). <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 there is a 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>

Table 47: show chassis cluster ethernet-switching status Output Fields (continued)

Field Name	Field Description
<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> 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

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: 0
node0 1      primary      no    no    None
node1 1      secondary    no    no    None

Ethernet switching status:
  Probe state is UP. Both nodes are in single ethernet switching domain(s).
```



## show chassis cluster information

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

**Syntax** `show chassis cluster information`

**Release Information** Command introduced in Junos OS Release 12.1X47-D10.

**Description** Display chassis cluster messages. The messages indicate each node's health condition and details of the monitored failure.

**Required Privilege Level** view

**Related Documentation**

- [show chassis cluster status on page 542](#)

**List of Sample Output** [show chassis cluster information on page 521](#)  
[show chassis cluster information \(Monitoring Abnormal Case\) on page 522](#)  
[show chassis cluster information \(Preempt Delay Timer\) on page 524](#)

**Output Fields** [Table 48 on page 521](#) lists the output fields for the **show chassis cluster information** command. Output fields are listed in the approximate order in which they appear.

*Table 48: 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 (Monitoring Abnormal Case)

user@host&gt; 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

## Sample Output

## show chassis cluster information (Preempt Delay Timer)

user@host&gt; show chassis cluster information

node0:

## Redundancy Group Information:

Redundancy Group 0 , Current State: secondary, Weight: 255

Time	From	To	Reason
Aug 4 12:30:02	hold	secondary	Hold timer expired
Aug 4 12:30:05	secondary	primary	Only node present
Aug 4 14:19:58	primary	secondary-hold	Manual failover
Aug 4 14:24:58	secondary-hold	secondary	Ready to become secondary

Redundancy Group 1 , Current State: secondary, Weight: 255

Time	From	To	Reason
Aug 4 14:07:57	secondary	primary	Remote is in secondary hold
Aug 4 14:20:23	primary	secondary-hold	Monitor failed: IF
Aug 4 14:20:24	secondary-hold	secondary	Ready to become secondary
Aug 4 14:20:54	secondary	primary	Remote is in secondary hold
Aug 4 14:21:30	primary	secondary-hold	Monitor failed: IF
Aug 4 14:21:31	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: primary, Weight: 255

Time	From	To	Reason
Aug 4 12:33:47	hold	secondary	Hold timer expired
Aug 4 14:19:57	secondary	primary	Remote is in secondary hold

Redundancy Group 1 , Current State: primary, Weight: 255

Time	From	To	Reason
Aug 4 14:07:56	secondary-hold	secondary	Ready to become secondary
Aug 4 14:20:22	secondary	primary	Remote is in secondary hold

Aug 4 14:20:37 primary primary-preempt-hold Preempt (99/101)

Aug 4 14:20:52 primary-preempt-hold secondary-hold Primary preempt hold timer e

```
Aug  4 14:20:53 secondary-hold secondary    Ready to become secondary
Aug  4 14:21:28 secondary      primary      Remote yield (99/0)
```

```
Chassis cluster LED information:
Current LED color: Green
Last LED change reason: No failures
```

## 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 166](#)
- [NTP Time Synchronization on SRX Series Devices on page 291](#)
- [Example: Simplifying Network Management by Synchronizing the Primary and Backup Nodes with NTP on page 292](#)
- [request chassis cluster configuration-synchronize on page 493](#)

**List of Sample Output** [show chassis cluster information configuration-synchronization on page 527](#)

**Output Fields** [Table 49 on page 526](#) 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 49: show chassis cluster information configuration-synchronization Output Fields*

Field Name	Field Description
<b>Node name</b>	Node (device) in the chassis cluster ( <b>node0</b> or <b>node1</b> ).
<b>Status</b>	<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>
<b>Events</b>	The timestamp of the event, the automatic configuration synchronization status, and the number of synchronization attempts.

## Sample Output

### show chassis cluster information configuration-synchronization

```
user@host> show chassis cluster information configuration-synchronization

node0:
-----
Configuration Synchronization:
  Status:
    Activation status: Enabled
    Last sync operation: Auto-Sync
    Last sync result: Not needed
    Last sync mgd messages:
  Events:
    Feb 25 22:21:49.174 : Auto-Sync: Not needed
node1:
-----
Configuration Synchronization:
  Status:
    Activation status: Enabled
    Last sync operation: Auto-Sync
    Last sync result: Succeeded
    Last sync mgd messages:
      mgd: rcp: /config/juniper.conf: No such file or directory
      Network security daemon: warning: You have enabled/disabled inet6 flow.
      Network security daemon: You must reboot the system for your change to
take effect.
      Network security daemon: If you have deployed a cluster, be sure to reboot
all nodes.
      mgd: commit complete
  Events:
    Feb 25 23:02:33.467 : Auto-Sync: In progress. Attempt: 1
    Feb 25 23:03:13.200 : Auto-Sync: Succeeded. Attempt: 1
```

## show chassis cluster information issu

**Supported Platforms** [SRX1500, SRX5400, SRX5600, SRX5800](#)

**Syntax** `show chassis cluster information issu`

**Release Information** Command introduced in Junos OS Release 12.1X47-D10.

**Description** Display chassis cluster messages. The messages indicate the progress of the in-service software upgrade (ISSU).

**Required Privilege Level** view

**Related Documentation**

- [show chassis cluster status on page 542](#)

**List of Sample Output** [show chassis cluster information issu on page 528](#)

**Output Fields** [Table 50 on page 528](#) lists the output fields for the `show chassis cluster information issu` command. Output fields are listed in the approximate order in which they appear.

*Table 50: show chassis cluster information issu Output Fields*

Field Name	Field Description
Node name	Node (device) in the chassis cluster ( <code>node0</code> or <code>node1</code> ).
CS Prereq	Status of all cold synchronization prerequisites: <ul style="list-style-type: none"> <li>• if_state sync—Status of if_state synchronization.</li> <li>• fabric link—Status of fabric link synchronization.</li> <li>• policy data sync—Status of policy data synchronization.</li> <li>• cp ready—Status of the central point.</li> <li>• VPN data sync—Status of the VPN data synchronization.</li> </ul>
CS RTO sync	Status of cold synchronization runtime objects.
CS postreq	Status of cold synchronization postrequirements.

## Sample Output

### show chassis cluster information issu

```
user@host> show chassis cluster information issu
```

```
node0:
```

```
-----
Cold Synchronization Progress:
```

```
CS Prereq          10 of 10 SPU's completed
```



```
1. if_state sync          10 SPU's completed
2. fabric link            10 SPU's completed
3. policy data sync       10 SPU's completed
4. cp ready               10 SPU's completed
5. VPN data sync          10 SPU's completed
CS RTO sync               10 of 10 SPU's completed
CS Postreq                10 of 10 SPU's completed
```

node1:

-----  
Cold Synchronization Progress:

```
CS Prereq                 10 of 10 SPU's completed
1. if_state sync          10 SPU's completed
2. fabric link            10 SPU's completed
3. policy data sync       10 SPU's completed
4. cp ready               10 SPU's completed
5. VPN data sync          10 SPU's completed
CS RTO sync               10 of 10 SPU's completed
CS Postreq                10 of 10 SPU's completed
```

## show chassis cluster interfaces

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

**Syntax** `show chassis cluster interfaces`

**Release Information** Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0. Output changed to support control interfaces in Junos OS Release 11.2. Output changed to support redundant pseudo interfaces in Junos OS Release 12.1X44-D10. For SRX5000 line devices, output changed to support the internal security association (SA) option in Junos OS Release 12.1X45-D10. Output changed to support MACsec status on control and fabric interfaces in Junos OS Release 15.1X49-D60.

**Description** Display the status of the control interface in a chassis cluster configuration.

**Required Privilege Level** view

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

**List of Sample Output**

[show chassis cluster interfaces on page 531](#)  
[show chassis cluster interfaces \(SRX5000 line devices\) on page 532](#)  
[show chassis cluster interfaces on page 533](#)  
[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 533](#)

**Output Fields** [Table 51 on page 530](#) lists the output fields for the `show chassis cluster interfaces` command. Output fields are listed in the approximate order in which they appear.

*Table 51: 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><b>NOTE:</b> This field is available only on SRX5000 line 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 51: 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	

## 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
fab0			
fab1	xe-7/0/3	Up / Down	Disabled
fab1			

## Redundant-ethernet Information:

Name	Status	Redundancy-group
reth0	Up	1
reth1	Up	2
reth2	Down	Not configured
reth3	Down	Not configured
reth4	Down	Not configured
reth5	Down	Not configured
reth6	Down	Not configured
reth7	Down	Not configured
reth8	Down	Not configured
reth9	Down	Not configured
reth10	Down	Not configured
reth11	Down	Not configured

## Redundant-pseudo-interface Information:

Name	Status	Redundancy-group
lo0	Up	1

## Interface Monitoring:

Interface	Weight	Status	Redundancy-group
ge-0/1/9	100	Up	0
ge-0/1/9	100	Up	

## Sample Output

### show chassis cluster interfaces

```
user@host> show chassis cluster interfaces
```

The following output is specific to fabric monitoring failure:

Control link status: Up

Control interfaces:

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

Fabric link status: Down

Fabric interfaces:

Name	Child-interface	Status (Physical/Monitored)	Security
fab0	ge-0/0/2	Down / Down	Disabled
fab1	ge-9/0/2	Up / Up	Disabled

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

fab1	xe-10/2/7	Up / Down	Disabled
fab1			

**Redundant-ethernet Information:**

Name	Status	Redundancy-group
reth0	Up	Not configured
reth1	Down	1

**Redundant-pseudo-interface Information:**

Name	Status	Redundancy-group
lo0	Up	0

## show chassis cluster ip-monitoring status redundancy-group

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

**Syntax** `show chassis cluster ip-monitoring status  
<redundancy-group group-number>`

**Release Information** Command introduced in Junos OS Release 9.6. Support for global threshold, current threshold, and weight of each monitored IP address added in Junos OS Release 12.1X47-D10.

**Description** Display the status of all monitored IP addresses for a redundancy group.

- Options**
- `none`— Display the status of monitored IP addresses for all redundancy groups on the node.
  - `redundancy-group group-number`— Display the status of monitored IP addresses under the specified redundancy group.

**Required Privilege Level** view

**Related Documentation**

- [clear chassis cluster failover-count](#)

**List of Sample Output** [show chassis cluster ip-monitoring status on page 536](#)  
[show chassis cluster ip-monitoring status redundancy-group on page 537](#)

**Output Fields** [Table 52 on page 535](#) lists the output fields for the `show chassis cluster ip-monitoring status` command.

*Table 52: 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.

Table 52: show chassis cluster ip-monitoring status Output Fields (continued)

Field Name	Field Description
<b>Reason</b>	Explanation for the reported status. See <a href="#">Table 53 on page 536</a> .
<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 53: 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

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

**Syntax** `show chassis cluster statistics`

**Release Information** Command modified in Junos OS Release 9.0. Output changed to support dual control ports in Junos OS Release 10.0.

**Description** Display information about chassis cluster services and interfaces.

**Required Privilege Level** view

**Related Documentation**

- [clear chassis cluster statistics on page 491](#)

**List of Sample Output** [show chassis cluster statistics on page 539](#)  
[show chassis cluster statistics \(SRX5000 Line Devices\) on page 540](#)  
[show chassis cluster statistics \(SRX5000 Line Devices\) on page 541](#)

**Output Fields** [Table 54 on page 538](#) lists the output fields for the **show chassis cluster statistics** command. Output fields are listed in the approximate order in which they appear.

*Table 54: 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 54: 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** Support for monitoring failures added in Junos OS Release 12.1X47-D10.

**Description** Display the current status of the Chassis Cluster. You can use this command to check the status of chassis cluster nodes, redundancy groups, and failover status.

- 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 464](#)
  - [clear chassis cluster failover-count on page 487](#)
  - [request chassis cluster failover node on page 494](#)
  - [request chassis cluster failover reset on page 497](#)

**List of Sample Output** [show chassis cluster status on page 543](#)  
[show chassis cluster status with preemptive delay on page 544](#)  
[show chassis cluster status redundancy-group 1 on page 544](#)

**Output Fields** [Table 55 on page 542](#) lists the output fields for the `show chassis cluster status` command. Output fields are listed in the approximate order in which they appear.

*Table 55: show chassis cluster status Output Fields*

Field Name	Field Description
Cluster ID	ID number (1-15) of a cluster is applicable for releases upto Junos OS Release 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	You can create up to 128 redundancy groups 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 55: 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 there is a 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>: Primary state can be preempted based on priority.</li> <li>• <b>No</b>: Change in priority will not preempt the primary state.</li> </ul>
Manual failover	<ul style="list-style-type: none"> <li>• <b>Yes</b>: Primary state 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>: Primary state 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

### 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

### show chassis cluster status with preemptive delay

```
user@host> show chassis cluster status
```

```
Cluster ID: 1
Node  Priority Status          Preempt Manual  Monitor-failures
Redundancy group: 0, Failover count: 1
node0  200      primary      no      no      None
node1  100      secondary    no      no      None
Redundancy group: 1, Failover count: 3
node0  200      primary-preempt-hold yes no  None node1  100      secondary
              yes      no      None
```

## Sample Output

### 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 567](#)

**List of Sample Output** [show chassis environment on page 546](#)

**Output Fields** [Table 56 on page 545](#) lists the output fields for the **show chassis environment** command. Output fields are listed in the approximate order in which they appear.

*Table 56: 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 environment cb

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

**Syntax** `show chassis environment cb  
<slot>`

**Release Information** Command introduced in Junos OS Release 9.2.  
Starting with Junos OS Release 12.1X47-D15, the SRX5K-SCBE (SCB2) is introduced and starting with Junos OS Release 15.1X49-D10, the SRX5K-SCB3 (SCB3) with enhanced midplane is introduced.

**Description** Display environmental information about the Control Boards (CBs) installed on SRX Series devices.

**Options** `slot`—(Optional) Display environmental information about the specified CB.

**Required Privilege Level** view

**Related Documentation**

- [request chassis cb on page 492](#)

**List of Sample Output** [show chassis environment cb \(SRX5600 devices with SRX5K-SCB3 \[SCB3\] and Enhanced Midplanes\) on page 550](#)  
[show chassis environment cb node 1 \(SRX5600 devices with SRX5K-SCB3 \[SCB3\] and Enhanced Midplanes\) on page 550](#)

**Output Fields** [Table 57 on page 549](#) lists the output fields for the `show chassis environment cb` command. Output fields are listed in the approximate order in which they appear.

*Table 57: show chassis environment cb Output Fields*

Field Name	Field Description
<b>State</b>	Status of the CB. If two CBs are installed and online, one is functioning as the master, and the other is the standby. <ul style="list-style-type: none"> <li>• <b>Online</b>—CB is online and running.</li> <li>• <b>Offline</b>—CB is powered down.</li> </ul>
<b>Temperature</b>	Temperature in Celsius (C) and Fahrenheit (F) of the air flowing past the CB. <ul style="list-style-type: none"> <li>• <b>Temperature Intake</b>—Measures the temperature of the air intake to cool the power supplies.</li> <li>• <b>Temperature Exhaust</b>—Measures the temperature of the hot air exhaust.</li> </ul>
<b>Power</b>	Power required and measured on the CB. The left column displays the required power, in volts. The right column displays the measured power, in millivolts.

Table 57: show chassis environment cb Output Fields (continued)

Field Name	Field Description
<b>BUS Revision</b>	Revision level of the generic bus device.
<b>FPGA Revision</b>	Revision level of the field-programmable gate array (FPGA).
<b>PMBus device</b>	<p>Enhanced SCB on SRX Series devices allows the system to save power by supplying only the amount of voltage that is required. Configurable PMBus devices are used to provide the voltage for each individual device. There is one PMBus device for each XF ASIC so that the output can be customized to each device. The following PMBus device information is displayed for devices with Enhanced MX SCB:</p> <ul style="list-style-type: none"> <li>• Expected voltage</li> <li>• Measured voltage</li> <li>• Measured current</li> <li>• Calculated power</li> </ul>

## Sample Output

### show chassis environment cb (SRX5600 devices with SRX5K-SCB3 [SCB3] and Enhanced Midplanes)

```

user@host> show chassis environment cb node 0
node0:
-----
CB 0 status:
State                Online Master
Temperature           34 degrees C / 93 degrees F
Power 1
  1.0 V                1002
  1.2 V                1198
  1.5 V                1501
  1.8 V                1801
  2.5 V                2507
  3.3 V                3300
  5.0 V                5014
  5.0 V RE             4982
  12.0 V               11988
  12.0 V RE            11930
Power 2
  4.6 V bias MidPlane  4801
  11.3 V bias PEM      11292
  11.3 V bias FPD      11272
  11.3 V bias POE 0    11214
  11.3 V bias POE 1    11253
Bus Revision          96
FPGA Revision         16
PMBus device          Expected voltage  Measured voltage  Measured current  Calculated power
XF ASIC A             1033 mV         1033 mV         15500 mA          16011 mW
XF ASIC B             1034 mV         1033 mV         15000 mA          15495 mW

```

### show chassis environment cb node 1 (SRX5600 devices with SRX5K-SCB3 [SCB3] and Enhanced Midplanes)

```

user@host> show chassis environment cb node 1

```

```
node1:
```

```
-----
CB 0 status:
```

State	Online Master			
Temperature	35 degrees C / 95 degrees F			
Power 1				
1.0 V	1002			
1.2 V	1198			
1.5 V	1504			
1.8 V	1801			
2.5 V	2507			
3.3 V	3325			
5.0 V	5014			
5.0 V RE	4943			
12.0 V	12007			
12.0 V RE	12007			
Power 2				
4.6 V bias MidPlane	4814			
11.3 V bias PEM	11272			
11.3 V bias FPD	11330			
11.3 V bias POE 0	11176			
11.3 V bias POE 1	11292			
Bus Revision	96			
FPGA Revision	16			
PMBus	Expected	Measured	Measured	Calculated
device	voltage	voltage	current	power
XF ASIC A	958 mV	959 mV	13500 mA	12946 mW
XF ASIC B	1033 mV	1031 mV	16500 mA	17011 mW

## show chassis ethernet-switch

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

**Syntax** `show chassis ethernet-switch`

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

**Description** Display information about the ports on the Control Board (CB) Ethernet switch on an SRX Series device.

**Required Privilege Level** view

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)

**List of Sample Output** [show chassis ethernet-switch on page 552](#)

**Output Fields** [Table 58 on page 552](#) lists the output fields for the `show chassis ethernet-switch` command. Output fields are listed in the approximate order in which they appear.

*Table 58: show chassis ethernet-switch Output Fields*

Field Name	Field Description
Link is good on port n connected to device	Information about the link between each port on the CB's Ethernet switch and one of the following devices:
or	<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>
Link is good on Fast Ethernet port n connected to device	
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

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

**Syntax** show chassis fabric plane

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

**Description** Show state of fabric management plane.

**Required Privilege Level** view

**Related Documentation**

- [show chassis fabric plane-location on page 562](#)

**List of Sample Output** [show chassis fabric plane\(SRX5600 and SRX5800 Devices with SRX5000 Line SCB II \[SRX5K-SCBE\] and SRX5K-RE-1800X4\) on page 557](#)

**Output Fields** [Table 59 on page 556](#) lists the output fields for the **show chassis fabric plane** command. Output fields are listed in the approximate order in which they appear.

*Table 59: 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 59: 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

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

**Syntax** show chassis fabric plane-location

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

**Description** Show fabric plane location.

**Required Privilege Level** view

**Related Documentation** • [show chassis fabric plane on page 556](#)

**List of Sample Output** [show chassis fabric plane-location\(SRX5600 and SRX5800 Devices with SRX5000 Line SCB II \[SRX5K-SCBE\] and SRX5K-RE-1800X4\) on page 562](#)

**Output Fields** [Table 60 on page 562](#) lists the output fields for the **show chassis fabric plane-location** command. Output fields are listed in the approximate order in which they appear.

*Table 60: 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 556](#)
- [show chassis fabric plane-location on page 562](#)

**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 565](#)

**Output Fields** [Table 61 on page 564](#) lists the output fields for the `show chassis fabric summary` command. Output fields are listed in the approximate order in which they appear.

*Table 61: show chassis fabric summary Output Fields*

Field Name	Field Description
Plane	Plane number.

---

Table 61: 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 fpc</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)

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

**Syntax** `show chassis hardware`  
`<clei-models | detail | extensive | models | node ( node-id | all | local | primary )>`

**Release Information** Command introduced in Junos OS Release 9.2. Command modified in Junos OS Release 9.2 to include **node** option.

**Description** Display chassis hardware information.

- Options**
- **clei-models**—(Optional) Display Common Language Equipment Identifier Code (CLEI) barcode and model number for orderable field-replaceable units (FRUs).
  - **detail | extensive**—(Optional) Display the specified level of output.
  - **models**—(Optional) Display model numbers and part numbers for orderable FRUs.
  - **node**—(Optional) For chassis cluster configurations, display chassis hardware information on a specific node (device) in the cluster.
    - **node-id**—Identification number of the node. It can be 0 or 1.
    - **local**—Display information about the local node.
    - **primary**—Display information about the primary node.

**Required Privilege Level** view

**Related Documentation**

- *Juniper Networks Devices Processing Overview*
- *Interface Naming Conventions*

**Output Fields** [Table 62 on page 567](#) lists the output fields for the **show chassis hardware** command. Output fields are listed in the approximate order in which they appear.

*Table 62: show chassis hardware Output Fields*

Field Name	Field Description
Item	Chassis component—Information about the backplane; power supplies; fan trays; Routing Engine; each Physical Interface Module (PIM)—reported as FPC and PIC—and each fan, blower, and impeller.
Version	Revision level of the chassis component.
Part Number	Part number for the chassis component.

*Table 62: show chassis hardware Output Fields (continued)*

Field Name	Field Description
<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.
<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 62: show chassis hardware Output Fields (continued)

Field Name	Field Description
Description	<p>Brief description of the hardware item:</p> <ul style="list-style-type: none"> <li>Type of power supply.</li> <li>Switch Control Board (SCB)</li> </ul> <p>Starting with Junos OS Release 12.1X47-D15 and Junos OS Release 17.3R1, the SRX5K-SCBE (SCB2) is introduced.</p> <ul style="list-style-type: none"> <li>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>With an SCB2, a third SCB is supported. If a third SCB is plugged in, it provides intra-chassis fabric redundancy.</li> <li>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>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>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> <p>Starting with Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the SRX5K-SCB3 (SCB3) with enhanced midplane is introduced.</p> <ul style="list-style-type: none"> <li>All existing SCB software that is supported by SCB2 is supported on SCB3.</li> <li>SRX5K-RE-1800X4 (RE2). Mixed Routing Engine use is not supported.</li> <li>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>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>SCB3 supports up to 400 Gbps per slot with old midplanes and up to 500 Gbps per slot with new midplanes.</li> <li>SCB3 supports fabric intra-chassis redundancy.</li> <li>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>SCB3 has a second external Ethernet port.</li> <li>Fabric bandwidth increasing mode is not supported.</li> </ul>

Table 62: 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           <p>Starting with Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the SRX5K-MPC3-100G10G (IOC3) and the SRX5K-MPC3-40G10G (IOC3) are introduced.</p> <ul style="list-style-type: none"> <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> <p><b>NOTE:</b> Fabric bandwidth increasing mode is not supported on IOC3.</p> </li> <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 and Junos OS Release 17.3R1, 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> <p><b>NOTE:</b> The RE2 provides significantly better performance than the previously used Routing Engine, even with a single core.</p> </li> </ul>

## show chassis hardware

### show chassis hardware

```

user@host> show chassis hardware
Hardware inventory:
Item          Version  Part number  Serial number  Description
Chassis                               CM0715AK0021  SRX1500
Midplane                               ACMA4255      SRX1500
CB 0          REV 08    711-053838   ACMA7529      CPU Board SRX700E
Routing Engine 0 BUILTIN    BUILTIN      SRX Routing Engine
FPC 0         REV 07    711-053832   ACMA3311      FEB
  PIC 0       BUILTIN    BUILTIN      12x1G-T-4x1G-SFP-4x10G
    Xcvr 12    REV 01    740-014132   61521013     SFP-T
    Xcvr 13    REV 02    740-013111   A281604      SFP-T
    Xcvr 14    REV 02    740-011613   NRN30NV      SFP-SX
    Xcvr 15    REV 02    740-011613   NRN2PWV      SFP-SX
    Xcvr 16    REV 01    740-021308   AJA17B5      SFP+-10G-SR
    Xcvr 17    REV 01    740-021308   MSP056B      SFP+-10G-SR
    Xcvr 18    REV 01    740-031980   AS920WJ      SFP+-10G-SR
    Xcvr 19    REV 01    740-031980   AS92W5N      SFP+-10G-SR
Power Supply 0 REV 01    740-055217   1EDP42500JZ  PS 400W 90-264V AC in
Fan Tray 0                               SRX1500 0, Front to Back
  Airflow - AFO
Fan Tray 1                               SRX1500 1, Front to Back
  Airflow - AFO
Fan Tray 2                               SRX1500 2, Front to Back
  Airflow - AFO
Fan Tray 3                               SRX1500 3, Front to Back
  Airflow - AFO

```

### show chassis hardware (SRX5600 and SRX5800 devices for SRX5K-MPC)

```

user@host> show chassis hardware
Hardware inventory:
Item          Version  Part number  Serial number  Description
Chassis                               JN12170EAAGA  SRX 5800
Midplane                               ACAX3849      SRX 5800 Backplane
FPM Board      REV 01    710-041799   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	C831XJ08S	XFP-10G-SR
PIC 2		BUILTIN	BUILTIN	1x 10GE(LAN/WAN) RichQ

PIC 3		BUILTIN	BUILTIN	1x 10GE(LAN/WAN) RichQ
FPC 3	REV 22	750-043157	ZH8189	SRX5k IOC II
CPU	REV 06	711-043360	YX3912	SRX5k MPC PMB
MIC 0	REV 01	750-055732	CACF9115	20x 1GE(LAN) SFP
PIC 0		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 2	REV 02	740-013111	B358549	SFP-T
Xcvr 9	REV 02	740-011613	PNB1FQS	SFP-SX
PIC 1		BUILTIN	BUILTIN	10x 1GE(LAN) SFP
Xcvr 9	REV 02	740-011613	PNB1FFF	SFP-SX
FPC 5	REV 01	750-027945	JW9665	SRX5k FIOC
CPU				
FPC 8	REV 08	750-023996	XA7234	SRX5k SPC
CPU	REV 02	710-024633	XA1599	SRX5k DPC PMB
PIC 0		BUILTIN	BUILTIN	SPU Cp-Flow
PIC 1		BUILTIN	BUILTIN	SPU Flow
Fan Tray 0	REV 03	740-014971	TP0902	Fan Tray
Fan Tray 1	REV 01	740-014971	TP0121	Fan Tray

### show chassis hardware

(SRX5600 and SRX5800 devices with SRX5000 line SRX5K-SCBE [SCB2] and SRX5K-RE-1800X4 [RE2])

```
user@host> show chassis hardware
node0:
```

#### Hardware inventory:

Item	Version	Part number	Serial number	Description
Chassis			JN1251EA1AGB	SRX5600
Midplane	REV 01	760-063936	ACRE2657	Enhanced SRX5600 Midplane
FPM Board	REV 01	710-024631	CABY3551	Front Panel Display
PEM 0	Rev 03	740-034701	QCS13380901P	PS 1.4-2.6kW; 90-264V
AC in				
PEM 1	Rev 03	740-034701	QCS133809019	PS 1.4-2.6kW; 90-264V
AC in				
Routing Engine 0	REV 02	740-056658	9009210105	SRX5k RE-1800X4
Routing Engine 1	REV 02	740-056658	9013115551	SRX5k RE-1800X4
CB 0	REV 01	750-062257	CADW3663	SRX5k SCB3
CB 1	REV 01	750-062257	CADZ3263	SRX5k SCB3
FPC 0	REV 18	750-054877	CABG6043	SRX5k SPC II
CPU		BUILTIN	BUILTIN	SRX5k DPC PPC
PIC 0		BUILTIN	BUILTIN	SPU Cp
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
FPC 1	REV 01	750-062243	CAEE5918	SRX5k IOC3 24XGE+6XLG
CPU	REV 02	711-062244	CADX8509	RMPC PMB
PIC 0		BUILTIN	BUILTIN	12x 10GE SFP+
Xcvr 0	REV 01	740-031980	273363A01891	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	273363A01915	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	ANA0BK6	SFP+-10G-SR
Xcvr 3	REV 01	740-031980	AP407GA	SFP+-10G-SR
Xcvr 9	REV 01	740-021308	MUC20G1	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	12x 10GE SFP+
PIC 2		BUILTIN	BUILTIN	3x 40GE QSFP+
PIC 3		BUILTIN	BUILTIN	3x 40GE QSFP+
WAN MEZZ	REV 15	750-049136	CAEE5845	MPC5E 24XGE OTN Mezz
FPC 3	REV 11	750-043157	CACL7452	SRX5k IOC II
CPU	REV 04	711-043360	CACP1977	SRX5k MPC PMB
MIC 0	REV 04	750-049488	CABL4759	10x 10GE SFP+
PIC 0		BUILTIN	BUILTIN	10x 10GE SFP+
Xcvr 0	REV 01	740-021308	CF36KM0SY	SFP+-10G-SR
Xcvr 1	REV 01	740-021308	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	RMPD PMB
PIC 0		BUILTIN	BUILTIN	12x 10GE SFP+
Xcvr 0	REV 01	740-031980	AP41BLH	SFP+-10G-SR
Xcvr 1	REV 01	740-031980	AQ400SL	SFP+-10G-SR
Xcvr 2	REV 01	740-031980	AP422LJ	SFP+-10G-SR
Xcvr 3	REV 01	740-021308	AMG0RBT	SFP+-10G-SR
Xcvr 9	REV 01	740-021308	MUC2FRG	SFP+-10G-SR
PIC 1		BUILTIN	BUILTIN	12x 10GE SFP+
PIC 2		BUILTIN	BUILTIN	3x 40GE QSFP+
PIC 3		BUILTIN	BUILTIN	3x 40GE QSFP+
WAN MEZZ	REV 15	750-049136	CAEA4837	MPC5E 24XGE OTN Mezz
FPC 3	REV 11	750-043157	CACA8784	SRX5k IOC II
CPU	REV 04	711-043360	CACA8820	SRX5k MPC PMB
MIC 0	REV 05	750-049488	CADF0521	10x 10GE SFP+
PIC 0		BUILTIN	BUILTIN	10x 10GE SFP+
Xcvr 0	REV 01	740-030658	AD1130A00PV	SFP+-10G-USR
Xcvr 1	REV 01	740-031980	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	REV 02	710-017254	KD9027	Front Panel Display
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	SRX5k DPC PPC
PIC 0		BUILTIN	BUILTIN	SPU Cp
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
FPC 2	REV 01	750-062243	CAEE5924	SRX5k IOC3 24XGE+6XLG
CPU	REV 01	711-062244	CAEB4890	SRX5k IOC3 PMB
PIC 0		BUILTIN	BUILTIN	12x 10GE SFP+
PIC 1		BUILTIN	BUILTIN	12x 10GE SFP+
PIC 2		BUILTIN	BUILTIN	3x 40GE QSFP+
Xcvr 0	REV 01	740-038623	MOC13156230449	QSFP+-40G-CU1M
Xcvr 2	REV 01	740-038623	MOC13156230449	QSFP+-40G-CU1M
PIC 3		BUILTIN	BUILTIN	3x 40GE QSFP+
WAN MEZZ	REV 01	750-062682	CAEE5817	24x 10GE SFP+ Mezz
FPC 4	REV 11	750-043157	CACY1595	SRX5k IOC II
CPU	REV 04	711-043360	CACZ8879	SRX5k MPC PMB
MIC 1	REV 04	750-049488	CACM6062	10x 10GE SFP+
PIC 2		BUILTIN	BUILTIN	10x 10GE SFP+
Xcvr 7	REV 01	740-021308	AD1439301TU	SFP+-10G-SR
Xcvr 8	REV 01	740-021308	AD1439301SD	SFP+-10G-SR
Xcvr 9	REV 01	740-021308	AD1439301TS	SFP+-10G-SR
FPC 5	REV 05	750-044175	ZZ1371	SRX5k SPC II
CPU		BUILTIN	BUILTIN	SRX5k DPC PPC
PIC 0		BUILTIN	BUILTIN	SPU Flow
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
Fan Tray				Enhanced Fan Tray

```
node1:
```

```
-----
Hardware inventory:
```

Item	Version	Part number	Serial number	Description
Chassis			JN124FEC0AGB	SRX5600
Midplane	REV 01	760-063936	ACRE2946	Enhanced SRX5600 Midplane
FPM Board	test	710-017254	test	Front Panel Display
PEM 0	Rev 01	740-038514	QCS114111003	DC 2.6kW Power Entry

Module				
PEM 1	Rev 01	740-038514	QCS12031100J	DC 2.6kW Power Entry
Module				
Routing Engine 0	REV 01	740-056658	9009186342	SRX5k RE-1800X4
CB 0	REV 01	750-062257	CAEB8178	SRX5k SCB3
FPC 0	REV 07	750-044175	CAAD0769	SRX5k SPC II
CPU		BUILTIN	BUILTIN	SRX5k DPC PPC
PIC 0		BUILTIN	BUILTIN	SPU Cp
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
FPC 4	REV 11	750-043157	CACY1592	SRX5k IOC II
CPU	REV 04	711-043360	CACZ8831	SRX5k MPC PMB
MIC 1	REV 04	750-049488	CACN0239	10x 10GE SFP+
PIC 2		BUILTIN	BUILTIN	10x 10GE SFP+
Xcvr 7	REV 01	740-031980	ARN23HW	SFP+-10G-SR
Xcvr 8	REV 01	740-031980	ARN2FVW	SFP+-10G-SR
Xcvr 9	REV 01	740-031980	ARN2YVM	SFP+-10G-SR
FPC 5	REV 10	750-056758	CADA8736	SRX5k SPC II
CPU		BUILTIN	BUILTIN	SRX5k DPC PPC
PIC 0		BUILTIN	BUILTIN	SPU Flow
PIC 1		BUILTIN	BUILTIN	SPU Flow
PIC 2		BUILTIN	BUILTIN	SPU Flow
PIC 3		BUILTIN	BUILTIN	SPU Flow
Fan Tray				Enhanced Fan Tray

## show chassis hardware (SRX4200)

```
user@host> show chassis hardware
```

Hardware inventory:

Item	Version	Part number	Serial number	Description
Chassis			DK2816AR0020	SRX4200
Mainboard	REV 01	650-071675	16061032317	SRX4200
Routing Engine 0		BUILTIN	BUILTIN	SRX Routing Engine
FPC 0		BUILTIN	BUILTIN	FEB
PIC 0		BUILTIN	BUILTIN	8x10G-SFP
Xcvr 0	REV 01	740-038153	MOC11511530020	SFP+-10G-CU3M
Xcvr 1	REV 01	740-038153	MOC11511530020	SFP+-10G-CU3M
Xcvr 2	REV 01	740-038153	MOC11511530020	SFP+-10G-CU3M
Xcvr 3	REV 01	740-038153	MOC11511530020	SFP+-10G-CU3M
Xcvr 4	REV 01	740-021308	04DZ06A00364	SFP+-10G-SR
Xcvr 5	REV 01	740-031980	233363A03066	SFP+-10G-SR
Xcvr 6	REV 01	740-021308	AL70SWE	SFP+-10G-SR
Xcvr 7	REV 01	740-031980	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 423](#)
- [request system snapshot \(Maintenance\)](#)

**List of Sample Output** [show chassis routing-engine \(Sample 1 - SRX550M\) on page 579](#)  
[show chassis routing-engine \(Sample 2 - vSRX\) on page 579](#)

**Output Fields** [Table 63 on page 578](#) lists the output fields for the **show chassis routing-engine** command. Output fields are listed in the approximate order in which they appear.

*Table 63: 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.  <i>NOTE:</i> Starting with Junos OS Release 15.1x49-D70 and Junos OS Release 17.3R1, there is a change in the method for calculating the memory utilization by a Routing Engine. The inactive memory is now subtracted from the total available memory. There is thus, a decrease in the reported value for used memory; as the inactive memory is now considered as free.
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.

Table 63: show chassis routing-engine Output Fields (continued)

Field Name	Field Description
<b>Interrupt</b>	Current CPU utilization in interrupt mode on the control plane core.
<b>Idle</b>	Current CPU utilization in idle mode on the control plane core.
<b>Model</b>	Routing Engine model.
<b>Start time</b>	Routing Engine start time.
<b>Uptime</b>	Length of time the Routing Engine has been up (running) since the last start.
<b>Last reboot reason</b>	Reason for the last reboot of the Routing Engine.
<b>Load averages</b>	The average number of threads waiting in the run queue or currently executing over 1-, 5-, and 15-minute periods.

## Sample Output

### show chassis routing-engine (Sample 1 - SRX550M)

```

user@host> show chassis routing-engine
Routing Engine status:
  Temperature           38 degrees C / 100 degrees F
  CPU temperature       36 degrees C / 96 degrees F
  Total memory          512 MB Max   435 MB used ( 85 percent)
  Control plane memory  344 MB Max   296 MB used ( 86 percent)
  Data plane memory     168 MB Max   138 MB used ( 82 percent)
  CPU utilization:
    User                 8 percent
    Background           0 percent
    Kernel               4 percent
    Interrupt            0 percent
    Idle                 88 percent
  Model                 RE-SRX5500-LOWMEM
  Serial ID             AAAP8652
  Start time            2009-09-21 00:04:54 PDT
  Uptime                52 minutes, 47 seconds
  Last reboot reason     0x200:chassis control reset
  Load averages:        1 minute   5 minute  15 minute
                        0.12       0.15     0.10

```

## Sample Output

### show chassis routing-engine (Sample 2 - vSRX)

```

user@host> show chassis routing-engine
Routing Engine status:
  Total memory          1024 MB Max   358 MB used ( 35 percent)
  Control plane memory  1024 MB Max   358 MB used ( 35 percent)
  5 sec CPU utilization:
    User                 2 percent
    Background           0 percent
    Kernel               4 percent

```

Interrupt	6 percent
Idle	88 percent
Model	VSRX RE
Start time	2015-03-03 07:04:18 UTC
Uptime	2 days, 11 hours, 51 minutes, 11 seconds
Last reboot reason	Router rebooted after a normal shutdown.
Load averages:	1 minute    5 minute    15 minute
	0.07            0.04            0.06

## show configuration chassis cluster traceoptions

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

**Syntax** show configuration chassis cluster traceoptions

**Release Information** Command introduced in Junos OS Release 12.1.

**Description** Display tracing options for the chassis cluster redundancy process.

**Required Privilege Level** view

**Related Documentation**

- [cluster \(Chassis\) on page 423](#)
- [traceoptions \(Chassis Cluster\) on page 476](#)

**List of Sample Output** [show configuration chassis cluster traceoptions on page 581](#)

**Output Fields** [Table 64 on page 581](#) 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 64: 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;
```

## set date ntp

---

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

**Syntax**

```
set date ntp {  
    server <server>;  
    force;  
    key <key>;  
    source-address <source-address>;  
}
```

**Release Information** Command introduced in Junos OS Release 15.1X49-D70.

**Description** Set the date and local time. If reject mode is enabled and the system rejected the update from the NTP server because it exceeds the configured threshold value, an administrator has two options to overrule the reject mode action: manually set the date and time in *YYYYMMDDhhmm.ss* format, or force synchronization of device time with the NTP server update by specifying the **force** option.

**Options** **ntp**—Use a NTP server to synchronize the current date and time setting on the SRX series devices.

**Server <server>**—Specify the IP address of one or more NTP servers.

**force** —Force system date and time to update to NTP server values. The device date and time are synchronized with the NTP proposed date and time even if reject is set as the action and the difference between the device time and NTP proposed time exceeds the default or the configured threshold value.

**key <key>**—Specify a key number to authenticate the NTP server used to synchronize the date and time. You must specify the same key number used to authenticate the server, configured at the **[edit system ntp authentication-key number]** hierarchy level.

**source-address <source-address>**—Specify the source address that the SRX Series devices use to contact the remote NTP server.

**Required Privilege Level** view

**Related Documentation**

- [show system ntp threshold on page 598](#)
- [ntp on page 456](#)
- [ntp threshold on page 457](#)
- [NTP Time Synchronization on SRX Series Devices on page 291](#)

**List of Sample Output** [set date ntp force on page 583](#)

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

## Sample Output

### set date ntp force

```
user@host> set date ntp force
18 Jul 16:52:43 ntpdate[3319]: NTP update request has been accepted, The time
offset is 147605840.624994 sec from the time server 66.129.255.62 which is larger
than the maximum threshold of 400 sec allowed.
```

## show interfaces (Gigabit Ethernet)

---

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

**Syntax** `show interfaces ge-fpc /pic/port`  
`<brief | detail | extensive | terse>`  
`<descriptions>`  
`<media>`  
`<snmp-index snmp-index>`  
`<statistics>`

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

**Description** Display status information about the specified Gigabit Ethernet interface.

**Options** `ge-fpc/pic/port`—Display standard information about the specified Gigabit Ethernet interface.

`brief | detail | extensive | terse`—(Optional) Display the specified level of output.

`descriptions`—(Optional) Display interface description strings.

`media`—(Optional) Display media-specific information about network interfaces.

`snmp-index snmp-index`—(Optional) Display information for the specified SNMP index of the interface.

`statistics`—(Optional) Display static interface statistics.

**Additional Information** In a logical system, this command displays information only about the logical interfaces and not about the physical interfaces.

**Required Privilege Level** view

**Related Documentation**

- [Understanding Media Access Control Security \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)
- [show security mka sessions \(SRX Series Device on page 607](#)
- [show security macsec statistics \(SRX Series Devices\) on page 601](#)

**List of Sample Output** [show interfaces \(Gigabit Ethernet\) \(for Fabric\) on page 594](#)  
[show interfaces detail for Fabric on page 595](#)



**Output Fields** Table 65 on page 585 describes the output fields for the **show interfaces** (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 Table 66 on page 594.

*Table 65: 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

Table 65: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
CoS queues	Number of CoS queues configured.	detail none
Hold-times	Current interface hold-time up and hold-time down, in milliseconds (ms).	
Current address	Configured MAC address.	detail none
Hardware address	Hardware MAC address.	detail none
Last flapped	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> .	detail none
Input Rate	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
Output Rate	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
Statistics last cleared	Time when the statistics for the interface were last set to zero.	detail extensive
Egress account overhead	Layer 2 overhead in bytes that is accounted in the interface statistics for egress traffic.	detail extensive
Ingress account overhead	Layer 2 overhead in bytes that is accounted in the interface statistics for ingress traffic.	detail extensive
Traffic statistics	<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>	detail

Table 65: 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 40x 1GE R, DPCE 20x 1GE + 2x 10GE R, or DPCE 40x 1GE 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 65: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
MACSec statistics	<p>Output</p> <ul style="list-style-type: none"> <li>Secure Channel Transmitted:</li> <li>Protected Packets, Encrypted Packets, Protected Bytes, Encrypted Bytes</li> </ul> <p>Input</p> <ul style="list-style-type: none"> <li>Secure Channel Received:</li> <li>Accepted Packets, Validated Bytes, Decrypted Bytes</li> </ul>	
OTN FEC statistics	<p>The forward error correction (FEC) counters provide the following statistics:</p> <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>	detail
PCS statistics	<p>(10-Gigabit Ethernet interfaces) Displays Physical Coding Sublayer (PCS) fault conditions from the WAN PHY or the LAN PHY device.</p> <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>	detail extensive

Table 65: 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 65: 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 65: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
<b>WIS section</b>	<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>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>	<p>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.</p> <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 65: 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</b> none
<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</b> none
<b>Protocol</b>	Protocol family. .	<b>detail</b> none
<b>MTU</b>	Maximum transmission unit size on the logical interface.	<b>detail</b> none
<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</b> none
<b>Service Name Table</b>	Name of the service name table for the interface configured with a PPPoE family.	<b>detail</b> none
<b>Max Sessions</b>	Maximum number of PPPoE logical interfaces that can be activated on the underlying interface.	<b>detail</b> none
<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</b> none
<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</b> none
<b>AC Name</b>	Name of the access concentrator.	<b>detail</b> none
<b>Maximum labels</b>	Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.	<b>detail</b> none



Table 65: 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>protocol-family</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 65: *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 66: *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>	<p>Traffic statistics:</p> <p>Input bytes: 496 bytes per packet, representing the Layer 2 packet</p> <p>MAC statistics:</p> <p>Received octets: 500 bytes per packet, representing the Layer 2 packet + 4 bytes</p>	The additional 4 bytes are for the CRC.
Inbound logical interface	<b>show interfaces ge-0/3/0.50 extensive</b>	<p>Traffic statistics:</p> <p>Input bytes: 478 bytes per packet, representing the Layer 3 packet</p>	
Outbound physical interface	<b>show interfaces ge-0/0/0 extensive</b>	<p>Traffic statistics:</p> <p>Input bytes: 490 bytes per packet, representing the Layer 3 packet + 12 bytes</p> <p>MAC statistics:</p> <p>Received octets: 478 bytes per packet, representing the Layer 3 packet</p>	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>	<p>Traffic statistics:</p> <p>Input bytes: 478 bytes per packet, representing the Layer 3 packet</p>	

## 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:

```

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:
0             Mapped forwarding classes
1             best-effort
2             expedited-forwarding
3             assured-forwarding
              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      2328 bps
Output bytes : 18277954     2264 bps
Input packets: 133917       1 pps
Output packets: 128737      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 system ntp threshold

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

**Syntax** `show system ntp threshold`

**Release Information** Command introduced in Junos OS Release 15.1X49-D70.

**Description** Display the current threshold and reject mode configured information.

**Required Privilege Level** view

**Related Documentation**

- [set date ntp on page 582](#)
- [ntp threshold on page 457](#)
- [ntp on page 456](#)
- [NTP Time Synchronization on SRX Series Devices on page 291](#)

**List of Sample Output** [show system ntp threshold on page 598](#)

**Output Fields** lists the output fields for the [Table 67 on page 598](#) `show system ntp threshold` command. Output fields are listed in the approximate order in which they appear.

*Table 67: show system ntp threshold Output Fields*

Field Name	Field Description
NTP threshold	Assign a threshold value for Network Time Protocol (NTP) adjustment that is outside of the acceptable NTP update and specify whether to accept or reject NTP synchronization when the proposed time from the NTP server exceeds the configured threshold value.
Success Criteria	Verifies the NTP threshold and provide the status of NTP adjustment mode (accept or reject).

## Sample Output

### show system ntp threshold

```
user@host> show system ntp threshold
NTP threshold: 400 sec
NTP adjustment reject mode is enabled
Success Criteria: verify threshold and reject mode can appear after user
configuration.
```

## 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 605](#)

**List of Sample Output** [show security macsec connections on page 600](#)

**Output Fields** [Table 68 on page 599](#) lists the output fields for the **show security macsec connections** command. Output fields are listed in the approximate order in which they appear.

*Table 68: 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.

*Table 68: show security macsec connections Output Fields (continued)*

Field Name	Field Description
<b>Replay protect</b>	<p>Replay protection setting. Replay protection is enabled when this output is <b>on</b> and disabled when this output is <b>off</b>.</p> <p>You can enable replay protection using the <b>replay-protect</b> statement in the connectivity association.</p>

## 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 (SRX Series Devices)

**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 \(MACsec\) on page 370](#)
  - [Configuring Media Access Control Security \(MACsec\) on page 373](#)
  - [macsec on page 450](#)
  - [show interfaces \(Gigabit Ethernet\) on page 584](#)
  - [show security mka sessions \(SRX Series Device on page 607](#)

**List of Sample Output** [show security macsec statistics interface on page 604](#)

**Output Fields** [Table 69 on page 602](#) 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 69: 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>	<p>Total number of packets transmitted out of the interface in the secure channel that were secured and encrypted using MACsec.</p> <p>Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).</p>	All levels
<b>Encrypted bytes</b>	<p>Total number of bytes transmitted out of the interface in the secure channel that were secured and encrypted using MACsec.</p> <p>Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).</p>	All levels
<b>Protected packets</b>	<p>Total number of packets transmitted out of the interface in the secure channel that were secured but not encrypted using MACsec.</p> <p>Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).</p>	All levels
<b>Protected bytes</b>	<p>Total number of bytes transmitted out of the interface in the secure channel that were secured but not encrypted using MACsec.</p> <p>Data packets are sent in the secure channel when MACsec is enabled, and are secured using a connectivity association key (CAK).</p>	All levels
<b>Fields for Secure Association transmitted</b>		
<b>Encrypted packets</b>	<p>Total number of packets transmitted out of the interface in the connectivity association that were secured and encrypted using MACsec.</p> <p>The total includes the data packets transmitted in the secure channel and the control packets secured using a connectivity association key (CAK).</p>	All levels
<b>Protected packets</b>	<p>Total number of packets transmitted out of the interface in the connectivity association that were secured but not encrypted using MACsec.</p> <p>The total includes the data packets transmitted in the secure channel and the control packets secured using a connectivity association key (CAK).</p>	All levels
<b>Fields for Secure Channel received</b>		

Table 69: 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

```
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 \(SRX Series Devices\)](#).

**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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)
- [show interfaces \(Gigabit Ethernet\) on page 584](#)
- [show security macsec statistics \(SRX Series Devices\) on page 601](#)

**List of Sample Output** [show security mka statistics on page 606](#)

**Output Fields** [Table 70 on page 605](#) lists the output fields for the `show security mka statistics` command. Output fields are listed in the approximate order in which they appear.

*Table 70: show security mka statistics Output Fields*

Field Name	Field Description
Received packets	<p>Number of received MKA control packets.</p> <p>This counter increments for received MKA control packets only. This counter does not increment when data packets are received.</p>
Transmitted packets	<p>Number of transmitted MKA packets</p> <p>This counter increments for transmitted MKA control packets only. This counter does not increment when data packets are transmitted.</p>
Version mismatch packets	Number of version mismatch packets.

Table 70: 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 (SRX Series 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 \(MACsec\) on page 370](#)
- [Configuring Media Access Control Security \(MACsec\) on page 373](#)
- [macsec on page 450](#)
- [show interfaces \(Gigabit Ethernet\) on page 584](#)
- [show security macsec statistics \(SRX Series Devices\) on page 601](#)

**List of Sample Output** [show security mka sessions on page 608](#)

**Output Fields** [Table 71 on page 607](#) lists the output fields for the **show security mka sessions** command. Output fields are listed in the approximate order in which they appear.

*Table 71: 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 <b>cak</b> 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.

Table 71: show security mka sessions Output Fields (continued)

Field Name	Field Description
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> .
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

```
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

```



## show security internal-security-association

**Supported Platforms** [SRX5400, SRX5600, SRX5800](#)

**Syntax** `show security internal-security-association`

**Release Information** Command introduced in Junos OS Release 12.1X47-D10.

**Description** Provide secure login by enabling the internal security association in a chassis cluster configuration.

**Required Privilege Level** view

**Related Documentation**

- Chassis Cluster Feature Guide for SRX Series Devices*

**List of Sample Output** [show security internal-security-association on page 609](#)

**Output Fields** [Table 72 on page 609](#) lists the output fields for the **show security internal-security-association** command. Output fields are listed in the approximate order in which they appear.

*Table 72: show security internal-security-association Output Fields*

Field Name	Field Description
Internal SA Status	State of the internal SA option on the chassis cluster control link: <b>enabled</b> or <b>disabled</b> .
Iked Encryption Status	State of the iked encryption.

## Sample Output

### show security internal-security-association

```
user@host>show security internal-security-association
```

```
node0:
```

```
-----
Internal SA Status : Enabled
Iked Encryption Status : Enabled
```

```
node1:
```

```
-----
Internal SA Status : Enabled
Iked Encryption Status : Enabled
```

## show system license (View)

**Supported Platforms** [SRX Series, vSRX](#)

**Syntax** `show system license`  
`<installed | keys | status | usage>`

**Release Information** Command introduced in Junos OS Release 9.5. Logical system status option added in Junos OS Release 11.2.

**Description** Display licenses and information about how licenses are used.

**Options** **none**—Display all license information.

**installed**—(Optional) Display installed licenses only.

**keys**—(Optional) Display a list of license keys. Use this information to verify that each expected license key is present.

**status**—(Optional) Display license status for a specified logical system or for all logical systems.

**usage**—(Optional) Display the state of licensed features.

**Required Privilege Level** view

**Related Documentation**

- [Adding New Licenses \(CLI Procedure\)](#)

**List of Sample Output**

- [show system license on page 611](#)
- [show system license installed on page 611](#)
- [show system license keys on page 612](#)
- [show system license usage on page 612](#)
- [show system license status logical-system all on page 612](#)

**Output Fields** [Table 73 on page 610](#) lists the output fields for the **show system license** command. Output fields are listed in the approximate order in which they appear.

*Table 73: show system license Output Fields*

Field Name	Field Description
<b>Feature name</b>	Name assigned to the configured feature. You use this information to verify that all the features for which you installed licenses are present.
<b>Licenses used</b>	Number of licenses used by the device. You use this information to verify that the number of licenses used matches the number configured. If a licensed feature is configured, the feature is considered used.

Table 73: show system license Output Fields (continued)

Field Name	Field Description
Licenses installed	Information about the installed license key: <ul style="list-style-type: none"> <li>• <b>License identifier</b>—Identifier associated with a license key.</li> <li>• <b>License version</b>—Version of a license. The version indicates how the license is validated, the type of signature, and the signer of the license key.</li> <li>• <b>Valid for device</b>—Device that can use a license key.</li> <li>• <b>Features</b>—Feature associated with a license.</li> </ul>
Licenses needed	Number of licenses required for features being used but not yet properly licensed.
Expiry	Time remaining in the grace period before a license is required for a feature being used.
Logical system license status	Displays whether a license is enabled for a logical system.

## Sample Output

### show system license

```
user@host> show system license
```

License usage:

Feature name	Licenses used	Licenses installed	Licenses needed	Expiry
av_key_kaspersky_engine 01:00:00 IST	1	1	0	2012-03-30
wf_key_surfcontrol_cpa 01:00:00 IST	0	1	0	2012-03-30
dynamic-vpn	0	1	0	permanent
ax411-wlan-ap	0	2	0	permanent

Licenses installed:

License identifier: JUNOS301998

License version: 2

Valid for device: AG4909AA0080

Features:

av\_key\_kaspersky\_engine - Kaspersky AV

date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST

License identifier: JUNOS302000

License version: 2

Valid for device: AG4909AA0080

Features:

wf\_key\_surfcontrol\_cpa - Web Filtering

date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST

### show system license installed

```
user@host> show system license installed
```

License identifier: JUNOS301998

License version: 2

Valid for device: AG4909AA0080

## Features:

av\_key\_kaspersky\_engine - Kaspersky AV  
 date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST

License identifier: JUNOS302000

License version: 2

Valid for device: AG4909AA0080

## Features:

wf\_key\_surfcontrol\_cpa - Web Filtering  
 date-based, 2011-03-30 01:00:00 IST - 2012-03-30 01:00:00 IST

## show system license keys

user@host> show system license keys

```
XXXXXXXXXX xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx
          xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx
          xxxxxx xxxxxx xxx
```

## show system license usage

user@host> show system license usage

Feature name	Licenses used	Licenses installed	Licenses needed	Expiry
av_key_kaspersky_engine 01:00:00 IST	1	1	0	2012-03-30
wf_key_surfcontrol_cpa 01:00:00 IST	0	1	0	2012-03-30
dynamic-vpn	0	1	0	permanent
ax411-wlan-ap	0	2	0	permanent

## show system license status logical-system all

user@host> show system license status logical-system all  
 Logical system license status:

logical system name	license status
root-logical-system	enabled
LSYS0	enabled
LSYS1	enabled
LSYS2	enabled

## show vrrp

**Supported Platforms** [SRX Series](#)

**Syntax** `show vrrp`  
`<brief | detail | extensive | summary>`  
`<interface interface-name>`  
`<logical-system logical-system-name>`  
`<nsr>`  
`<track track-interfaces>`

**Release Information** Statement introduced in Junos OS Release 18.1R1 for the SRX Series devices.

**Description** Display information and status about VRRP groups.

**Options** **none**—(Same as brief) Display brief status information about all VRRP interfaces.

**brief | detail | extensive | summary**—(Optional) Display the specified level of output.

**interface *interface-name***—(Optional) Display information and status about the specified VRRP interface.

**logical-system**—(Optional) Display information and status about the specified logical system.

**nsr**—(Optional) Display information and status about the master routing engine.

**track *track-interfaces***—(Optional) Display information and status about VRRP track interfaces.

**Required Privilege Level** view

**Related Documentation**

- [Configuring VRRP for IPv6 \(CLI Procedure\)](#)
- [Understanding VRRP on SRX Series Devices on page 279](#)
- [Example: Configuring VRRP/VRRPv3 on Chassis Cluster Redundant Ethernet Interfaces on page 282](#)

**List of Sample Output**

- [show vrrp on page 618](#)
- [show vrrp brief on page 618](#)
- [show vrrp detail \(IPv6\) on page 618](#)
- [show vrrp detail \(Route Track\) on page 619](#)
- [show vrrp extensive on page 619](#)
- [show vrrp interface on page 621](#)
- [show vrrp summary on page 621](#)
- [show vrrp track detail on page 621](#)
- [show vrrp track summary on page 622](#)

**Output Fields** Table 74 on page 614 lists the output fields for the **show vrrp** command. Output fields are listed in the approximate order in which they appear.

*Table 74: show vrrp Output Fields*

Field Name	Field Description	Level of Output
<b>Interface</b>	Name of the logical interface.	none, brief, extensive, summary
<b>Interface index</b>	Physical interface index number, which reflects its initialization sequence.	extensive
<b>Groups</b>	Total number of VRRP groups configured on the interface.	extensive
<b>Active</b>	Total number of VRRP groups that are active (that is, whose interface state is either up or down).	extensive
<b>Interface VRRP PDU statistics</b>	Nonerrored statistics for the logical interface: <ul style="list-style-type: none"> <li>• <b>Advertisement sent</b>—Number of VRRP advertisement protocol data units (PDUs) that the interface has transmitted.</li> <li>• <b>Advertisement received</b>—Number of VRRP advertisement PDUs received by the interface.</li> <li>• <b>Packets received</b>—Number of VRRP packets received for VRRP groups on the interface.</li> <li>• <b>No group match received</b>—Number of VRRP packets received for VRRP groups that do not exist on the interface.</li> </ul>	extensive
<b>Interface VRRP PDU error statistics</b>	Errored statistics for the logical interface: <ul style="list-style-type: none"> <li>• <b>Invalid IPAH next type received</b>—Number of packets received that use the IP Authentication Header protocol (IPAH) and that do not encapsulate VRRP packets.</li> <li>• <b>Invalid VRRP ttl value received</b>—Number of packets received whose IP time-to-live (TTL) value is not 255.</li> <li>• <b>Invalid VRRP version received</b>—Number of packets received whose VRRP version is not 2.</li> <li>• <b>Invalid VRRP pdu type received</b>—Number of packets received whose VRRP PDU type is not 1.</li> <li>• <b>Invalid VRRP authentication type received</b>—Number of packets received whose VRRP authentication is not none, simple, or md5.</li> <li>• <b>Invalid VRRP IP count received</b>—Number of packets received whose VRRP IP count exceeds 8.</li> <li>• <b>Invalid VRRP checksum received</b>—Number of packets received whose VRRP checksum does not match the calculated value.</li> </ul>	extensive
<b>Physical interface</b>	Name of the physical interface.	detail, extensive
<b>Unit</b>	Logical unit number.	All levels
<b>Address</b>	Address of the physical interface.	none, brief, detail, extensive
<b>Index</b>	Physical interface index number, which reflects its initialization sequence.	detail, extensive

Table 74: show vrrp Output Fields (continued)

Field Name	Field Description	Level of Output
SNMP ifIndex	SNMP index number for the physical interface.	detail, extensive
VRRP-Traps	Status of VRRP traps: <b>Enabled</b> or <b>Disabled</b> .	detail, extensive
Type and Address	Identifier for the address and the address itself: <ul style="list-style-type: none"> <li><b>lcl</b>—Configured local interface address.</li> <li><b>mas</b>—Address of the master device. This address is displayed only when the local interface is acting as a backup device.</li> <li><b>vip</b>—Configured virtual IP addresses.</li> </ul>	none, brief, summary
Interface state or Int state	State of the physical interface: <ul style="list-style-type: none"> <li><b>down</b>—The device is present and the link is unavailable.</li> <li><b>not present</b>—The interface is configured, but no physical device is present.</li> <li><b>unknown</b>—The VRRP process has not had time to query the kernel about the state of the interface.</li> <li><b>up</b>—The device is present and the link is established.</li> </ul>	none, brief, extensive, summary
Group	VRRP group number.	none, brief, extensive, summary
State	VRRP state: <ul style="list-style-type: none"> <li><b>backup</b>—The interface is acting as the backup device interface.</li> <li><b>bringup</b>—VRRP is just starting, and the physical device is not yet present.</li> <li><b>idle</b>—VRRP is configured on the interface and is disabled. This can occur when VRRP is first enabled on an interface whose link is established.</li> <li><b>initializing</b>—VRRP is initializing.</li> <li><b>master</b>—The interface is acting as the master device interface.</li> <li><b>transition</b>—The interface is changing between being the backup and being the master device.</li> </ul>	extensive
Priority	Configured VRRP priority for the interface.	detail, extensive
Advertisement interval	Configured VRRP advertisement interval.	detail, extensive
Authentication type	Configured VRRP authentication type: <b>none</b> , <b>simple</b> , or <b>md5</b> .	detail, extensive
Preempt	Whether preemption is allowed on the interface: <b>yes</b> or <b>no</b> .	detail, extensive
Accept-data mode	Whether the interface is configured to accept packets destined for the virtual IP address: <b>yes</b> or <b>no</b> .	detail, extensive
VIP count	Number of virtual IP addresses that have been configured on the interface.	detail, extensive
VIP	List of virtual IP addresses configured on the interface.	detail, extensive

Table 74: show vrrp Output Fields (continued)

Field Name	Field Description	Level of Output
Advertisement timer	Time until the advertisement timer expires.	detail, extensive
Master router	IP address of the interface that is acting as the master. If the VRRP interface is down, the output is N/A.	detail, extensive
Virtual router uptime	Time that the virtual device has been up.	detail, extensive
Master router uptime	Time that the master device has been up.	detail, extensive
Virtual MAC	MAC address associated with the virtual IP address.	detail, extensive
Tracking	Whether tracking is <b>enabled</b> or <b>disabled</b> .	detail, extensive
Current priority	Current operational priority for being the VRRP master.	detail, extensive
Configured priority	Configured base priority for being the VRRP master.	detail, extensive
Priority hold-time	Minimum time interval, in seconds, between successive changes to the current priority. <b>Disabled</b> indicates no minimum interval.	detail, extensive
Remaining-time	( <b>track</b> option only) Displays the time remaining in the priority hold-time interval.	detail
Interface tracking	Whether interface tracking is enabled or disabled. When enabled, the output also displays the number of tracked interfaces.	detail extensive
Interface/Tracked interface	Name of the tracked interface.	detail extensive
Int state/Interface state	Current operational state of the tracked interface: <b>up</b> or <b>down</b> .	detail, extensive
Int speed/Speed	Current operational speed, in bits per second, of the tracked interface.	detail, extensive
Incurred priority cost	Operational priority cost incurred due to the state and speed of this tracked interface. This cost is applied to the configured priority to obtain the current priority.	detail, extensive
Threshold	Speed below which the corresponding priority cost is incurred. In other words, when the speed of the interface drops below the threshold speed, the corresponding priority cost is incurred.  An entry of <b>down</b> means that the corresponding priority cost is incurred when the interface is down.	detail, extensive
Route tracking	Whether route tracking is enabled or disabled. When enabled, the output also displays the number of tracked routes.	detail, extensive



Table 74: show vrrp Output Fields (continued)

Field Name	Field Description	Level of Output
Route count	The number of routes being tracked.	detail, extensive
Route	The IP address of the route being tracked.	detail, extensive
VRF name	The VPN routing and forwarding (VRF) routing instance that the tracked route is in.	detail, extensive
Route state	The state of the route being tracked: <b>up</b> , <b>down</b> , or <b>unknown</b> .	detail, extensive
Priority cost	Configured priority cost. This value is incurred when the interface speed drops below the corresponding threshold or when the tracked route goes down.	detail, extensive
Active	Whether the threshold is active (*). If the threshold is active, the corresponding priority cost is incurred.	detail, extensive
Group VRRP PDU statistics	Number of VRRP advertisements sent and received by the group.	extensive
Group VRRP PDU error statistics	Errored statistics for the VRRP group: <ul style="list-style-type: none"> <li>• <b>Bad authentication type received</b>—Number of VRRP PDUs received with an invalid authentication type. The received authentication can be <b>none</b>, <b>simple</b>, or <b>md5</b> and must be the same for all devices in the VRRP group.</li> <li>• <b>Bad password received</b>—Number of VRRP PDUs received with an invalid key (password). The password for simple authentication must be the same for all devices in the VRRP group.</li> <li>• <b>Bad MD5 digest received</b>—Number of VRRP PDUs received for which the MD5 digest computed from the VRRP PDU differs from the digest expected by the VRRP instance configured on the device.</li> <li>• <b>Bad advertisement timer received</b>—Number of VRRP PDUs received with an advertisement time interval that is inconsistent with the one in use among the devices in the VRRP group.</li> <li>• <b>Bad VIP count received</b>—Number of VRRP PDUs whose virtual IP address counts differ from the count that has been configured on the VRRP instance.</li> <li>• <b>Bad VIPADDR received</b>—Number of VRRP PDUs whose virtual IP addresses differ from the list of virtual IP addresses configured on the VRRP instance.</li> </ul>	extensive
Group state transition statistics	State transition statistics for the VRRP group: <ul style="list-style-type: none"> <li>• <b>Idle to master transitions</b>—Number of times that the VRRP instance transitioned from the idle state to the master state.</li> <li>• <b>Idle to backup transitions</b>—Number of times that the VRRP instance transitioned from the idle state to the backup state.</li> <li>• <b>Backup to master transitions</b>—Number of times that the VRRP instance transitioned from the backup state to the master state.</li> <li>• <b>Master to backup transitions</b>—Number of times that the VRRP instance transitioned from the master state to the backup state.</li> </ul>	extensive
Vlan-id	ID of Vlan	detail

Table 74: show vrrp Output Fields (continued)

Field Name	Field Description	Level of Output
VR state	VRRP information: <ul style="list-style-type: none"> <li>• <b>backup</b>—The interface is acting as the backup device interface.</li> <li>• <b>bringup</b>—VRRP is just starting, and the physical device is not yet present.</li> <li>• <b>idle</b>—VRRP is configured on the interface and is disabled. This can occur when VRRP is first enabled on an interface whose link is established.</li> <li>• <b>initializing</b>—VRRP is initializing.</li> <li>• <b>master</b>—The interface is acting as the master device interface.</li> <li>• <b>transition</b>—The interface is changing between being the backup and being the master device.</li> </ul>	none, brief
Timer	VRRP timer information: <ul style="list-style-type: none"> <li>• <b>A</b>—Time, in seconds, until the advertisement timer expires.</li> <li>• <b>D</b>—Time, in seconds, until the Master is Dead timer expires.</li> </ul>	none, brief

## Sample Output

### show vrrp

```

user@host> show vrrp
Interface      State      Group  VR state  Timer  Type  Address
ge-0/0/0.121   up         1      master    A 1.052 1c1
ge2001:db8::12:1:1:1
                                     vip
ge2001:db8:0:1:12:1:1:99
                                     vip
ge2001:db8::12:1:1:99
ge-0/0/2.131   up         1      master    A 0.364 1c1
ge2001:db8::13:1:1:1
                                     vip
ge2001:db8:0:1:13:1:1:99
                                     vip
ge2001:db8::13:1:1:99

```

### show vrrp brief

The output for the **show vrrp brief** command is identical to that for the **show vrrp** command. For sample output, see [show vrrp on page 618](#).

### show vrrp detail (IPv6)

```

user@host> show vrrp detail
Physical interface: ge-0/0/0, Unit: 121, Vlan-id: 212, Address:
ge2001:db8::12:1:1:1/120
  Index: 67, SNMP ifIndex: 45, VRRP-Traps: enabled
  Interface state: up, Group: 1, State: master
  Priority: 200, Advertisement interval: 1, Authentication type: none
  Preempt: yes, Accept-data mode: no, VIP count: 2, VIP: ge2001:db8:0:1:12:1:1:99,
ge2001:db8::12:1:1:99
  Advertisement timer: 1.121s, Master router: ge2001:db8:0:1:12:1:1:1
  Virtual router uptime: 00:03:47, Master router uptime: 00:03:41

```

Virtual MAC: 00:00:5e:00:02:01  
Tracking: disabled

Physical interface: ge-0/0/2, Unit: 131, Vlan-id: 213, Address:  
ge2001:db8::13:1:1:1/120  
Index: 69, SNMP ifIndex: 47, VRRP-Traps: enabled  
Interface state: up, Group: 1, State: master  
Priority: 200, Advertisement interval: 1, Authentication type: none  
Preempt: yes, Accept-data mode: no, VIP count: 2, VIP: ge2001:db8:0:1:13:1:1:99,  
ge2001:db8::13:1:1:99  
Advertisement timer: 0.327s, Master router: ge2001:db8:0:1:13:1:1:1  
Virtual router uptime: 00:03:47, Master router uptime: 00:03:41  
Virtual MAC: 00:00:5e:00:02:01  
Tracking: disabled

### show vrrp detail (Route Track)

```
user@host> show vrrp detail
Physical interface: ge-1/1/0, Unit: 0, Address: 192.0.2.30/24
Index: 67, SNMP ifIndex: 379, VRRP-Traps: enabled
Interface state: up, Group: 100, State: master
Priority: 150, Advertisement interval: 1, Authentication type: none
Preempt: yes, Accept-data mode: no, VIP count: 1, VIP: 192.0.2.100
Advertisement timer: 1.218s, Master router: 192.0.2.30
Virtual router uptime: 00:04:28, Master router uptime: 00:00:13
Virtual MAC: 00:00:5e:00:01:64
Tracking: enabled
  Current priority: 150, Configured priority: 150
  Priority hold-time: disabled
  Interface tracking: disabled
  Route tracking: enabled, Route count: 1
    Route      VRF name    Route state  Priority cost
    198.51.100.0/24  default      up           30
```

### show vrrp extensive

```
user@host> show vrrp extensive
Interface: ge-0/0/0.121, Interface index: 67, Groups: 1, Active : 1
  Interface VRRP PDU statistics
    Advertisement sent           : 188
    Advertisement received        : 0
    Packets received              : 0
    No group match received       : 0
  Interface VRRP PDU error statistics
    Invalid IPAH next type received : 0
    Invalid VRRP TTL value received : 0
    Invalid VRRP version received  : 0
    Invalid VRRP PDU type received  : 0
    Invalid VRRP authentication type received: 0
    Invalid VRRP IP count received  : 0
    Invalid VRRP checksum received  : 0

Physical interface: ge-0/0/0, Unit: 121, Vlan-id: 212, Address:
ge2001:db8::12:1:1:1/120
Index: 67, SNMP ifIndex: 45, VRRP-Traps: enabled
Interface state: up, Group: 1, State: master
Priority: 200, Advertisement interval: 1, Authentication type: none
Preempt: yes, Accept-data mode: no, VIP count: 2, VIP: ge2001:db8:0:1:12:1:1:99,
ge2001:db8::12:1:1:99
Advertisement timer: 1.034s, Master router: ge2001:db8:0:1:12:1:1:1
```

```

Virtual router uptime: 00:04:04, Master router uptime: 00:03:58
Virtual MAC: 00:00:5e:00:02:01
Tracking: disabled
Group VRRP PDU statistics
  Advertisement sent      :      188
  Advertisement received  :         0
Group VRRP PDU error statistics
  Bad authentication type received:      0
  Bad password received      :         0
  Bad MD5 digest received    :         0
  Bad advertisement timer received:      0
  Bad VIP count received     :         0
  Bad VIPADDR received      :         0
Group state transition statistics
  Idle to master transitions :         0
  Idle to backup transitions :         1
  Backup to master transitions :         1
  Master to backup transitions :         0

Interface: ge-0/0/2.131, Interface index: 69, Groups: 1, Active : 1
Interface VRRP PDU statistics
  Advertisement sent      :      186
  Advertisement received  :         0
  Packets received        :         0
  No group match received :         0
Interface VRRP PDU error statistics
  Invalid IPAH next type received :      0
  Invalid VRRP TTL value received :      0
  Invalid VRRP version received   :      0
  Invalid VRRP PDU type received  :      0
  Invalid VRRP authentication type received:      0
  Invalid VRRP IP count received  :      0
  Invalid VRRP checksum received  :      0

Physical interface: ge-0/0/2, Unit: 131, Vlan-id: 213, Address:
ge2001:db8::13:1:1:1/120
Index: 69, SNMP ifIndex: 47, VRRP-Traps: enabled
Interface state: up, Group: 1, State: master
Priority: 200, Advertisement interval: 1, Authentication type: none
Preempt: yes, Accept-data mode: no, VIP count: 2, VIP: ge2001:db8:0:1:13:1:1:99,
ge2001:db8::13:1:1:99
Advertisement timer: 0.396s, Master router: ge2001:db8:0:1:13:1:1:1
Virtual router uptime: 00:04:04, Master router uptime: 00:03:58
Virtual MAC: 00:00:5e:00:02:01
Tracking: disabled
Group VRRP PDU statistics
  Advertisement sent      :      186
  Advertisement received  :         0
Group VRRP PDU error statistics
  Bad authentication type received:      0
  Bad password received      :         0
  Bad MD5 digest received    :         0
  Bad advertisement timer received:      0
  Bad VIP count received     :         0
  Bad VIPADDR received      :         0
Group state transition statistics
  Idle to master transitions :         0
  Idle to backup transitions :         1
  Backup to master transitions :         1
  Master to backup transitions :         0

```

**show vrrp interface**

```

user@host> show vrrp interface
Interface: ge-0/0/0.121, Interface index: 67, Groups: 1, Active : 1
  Interface VRRP PDU statistics
    Advertisement sent           :          205
    Advertisement received       :           0
    Packets received             :           0
    No group match received      :           0
  Interface VRRP PDU error statistics
    Invalid IPAH next type received :           0
    Invalid VRRP TTL value received :           0
    Invalid VRRP version received  :           0
    Invalid VRRP PDU type received :           0
    Invalid VRRP authentication type received:           0
    Invalid VRRP IP count received :           0
    Invalid VRRP checksum received :           0

Physical interface: ge-0/0/0, Unit: 121, Vlan-id: 212, Address:
ge2001:db8::12:1:1:1/120
  Index: 67, SNMP ifIndex: 45, VRRP-Traps: enabled
  Interface state: up, Group: 1, State: master
  Priority: 200, Advertisement interval: 1, Authentication type: none
  Preempt: yes, Accept-data mode: no, VIP count: 2, VIP: ge2001:db8:0:1:12:1:1:99,
gec2001:db8::12:1:1:99
  Advertisement timer: 0.789s, Master router: ge2001:db8:0:1:12:1:1:1
  Virtual router uptime: 00:04:26, Master router uptime: 00:04:20
  Virtual MAC: 00:00:5e:00:02:01
  Tracking: disabled
  Group VRRP PDU statistics
    Advertisement sent           :          205
    Advertisement received       :           0
  Group VRRP PDU error statistics
    Bad authentication type received:           0
    Bad password received         :           0
    Bad MD5 digest received       :           0
    Bad advertisement timer received:           0
    Bad VIP count received        :           0
    Bad VIPADDR received          :           0
  Group state transition statistics
    Idle to master transitions    :           0
    Idle to backup transitions    :           1
    Backup to master transitions  :           1
    Master to backup transitions  :           0

```

**show vrrp summary**

```

user@host> show vrrp summary
Interface  State      Group  VR state  Type  Address
ge-3/2/6.0  up         1      backup    1c1   10.57.0.2
                                vip    10.57.0.100

```

**show vrrp track detail**

```

user@host> show vrrp track detail
Tracked interface: ae1.211
  State: up, Speed: 400m
  Incurred priority cost: 0
  Threshold  Priority cost  Active
  400m       10

```

```
300m      60
200m      110
100m      160
down      190
Tracking VRRP interface: ae0.210, Group: 1
VR State: master
Current priority: 200, Configured priority: 200
Priority hold-time: disabled,    Remaining-time: 50.351
```

#### show vrrp track summary

```
user@host> show vrrp track summary
```

Track if	State	Speed	VRRP if	Group	VR State	Current priority
ae1.211	up	400m	ae0.210	1	master	200