JUNIPER NETWORKS | Engineering Simplicity

Network Configuration Example

Configuring SRX Chassis Clusters for High Availability

Published
2023-09-13

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

About This Guide | iv

1 Configuring SRX Chassis Clusters for High Availability

About This Network Configuration Example | 2

Use Case for SRX Chassis Clusters | 2

Chassis Cluster Overview | 3

Example: Configuring an Active/Passive Cluster Deployment | 5

- Requirements | 5
- Overview | 6
- Configuration | 8
  - Verification | 15

Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster | 26

- Requirements | 27
- Overview | 27
- Configuration | 30
  - Verification | 41

Example: Configuring an Active/Active Layer 3 Cluster Deployment | 53

- Requirements | 53
- Overview | 54
- Configuration | 56
  - Verification | 63

About This Guide

This document describes different high availability deployment scenarios for high-end SRX Series devices. It also provides a step-by-step configuration example for each of the different scenarios.

1

CHAPTER

Configuring SRX Chassis Clusters for High Availability

About This Network Configuration Example | 2

Use Case for SRX Chassis Clusters | 2

Chassis Cluster Overview | 3

Example: Configuring an Active/Passive Cluster Deployment | 5

Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster | 26

Example: Configuring an Active/Active Layer 3 Cluster Deployment | 53

About This Network Configuration Example

This document describes different high availability deployment scenarios for high-end SRX Series devices. It also provides a step-by-step configuration example for each of the different scenarios.

Use Case for SRX Chassis Clusters

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

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

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

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

Using the SRX Chassis Cluster configuration described in "Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster" on page 26 will reduce your downtime and save you money.

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

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

Chassis Cluster Overview

IN THIS SECTION

- Control Plane and Data Plane | 4

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

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple Packet Forwarding Engines. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.
- Support for generic routing encapsulation (GRE) and IP-over-IP IPv4 tunnels used to route encapsulated IPv4/IPv6 traffic.

Control Plane and 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.

Clusters and nodes are identified in the following ways:

- A cluster is identified by a cluster ID (`cluster-id`) specified as a number 1 through 15.
- A cluster node is identified by a node ID (`node`) specified as a number from 0 to 1.

Chassis clustering of interfaces and services is provided 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 given time. When a redundancy group is said to be primary on a node, the objects on that node are active. Redundancy groups is a concept of Junos OS Services Redundancy Protocol (JSRP) clustering that is similar to virtual security interface (VSI) in Juniper Networks ScreenOS® Software. Basically, each node has an interface in the redundancy group, where only one interface is active at a time. Redundancy group 0 is always for the control plane, while redundancy group 1+ is always for the data plane ports.

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, Services Processing Unit (SPU) monitoring, failures, and manual failovers.

The following high-end SRX Series Services Gateways are supported:

- SRX1400
- SRX3400

- SRX3600
- SRX5600
- SRX5800

RELATED DOCUMENTATION

Use Case for SRX Chassis Clusters | 2
Example: Configuring an Active/Passive Cluster Deployment | 5
Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster | 26
Example: Configuring an Active/Active Layer 3 Cluster Deployment | 53

Example: Configuring an Active/Passive Cluster Deployment

IN THIS SECTION

- Requirements | 5
- Overview | 6
- Configuration | 8

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

Requirements

This example uses the following hardware and software components:

- Two Juniper Networks SRX5800 Services Gateways with identical hardware configurations running Junos OS Release 9.6 or later.

- One Juniper Networks MX240 3D Universal Edge Router running Junos OS Release 9.6 or later.
- One Juniper Networks EX8208 Ethernet Switch running Junos OS Release 9.6 or later.

NOTE: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

Before you begin:

- Physically connect the two SRX Services Gateways. This example is based on back-to-back connections (direct connection) for both the fabric and control ports.

Overview

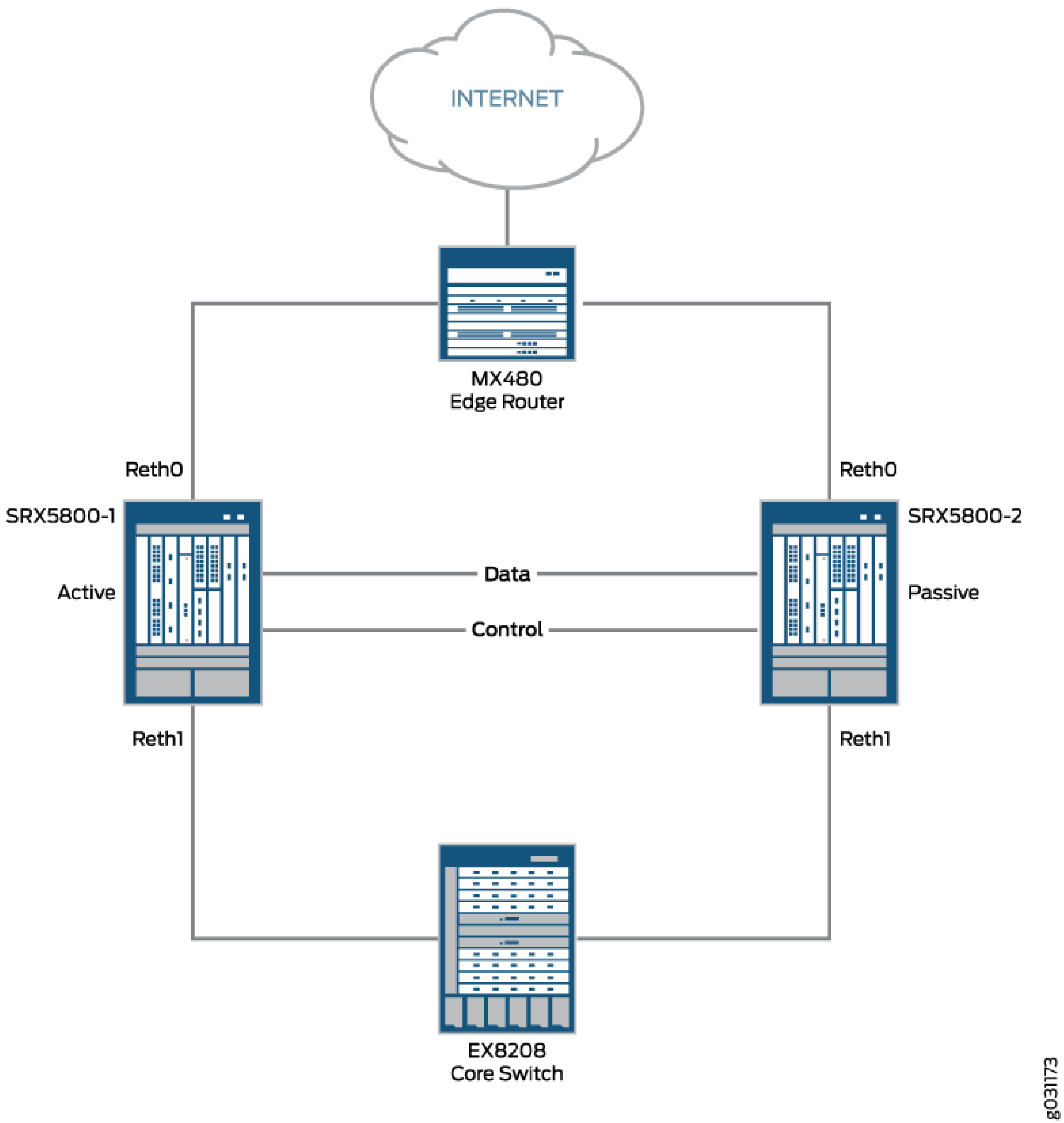
This example shows how to set up basic active/passive chassis clustering on a pair of high-end SRX Series devices. The basic active/passive example is the most common type of chassis cluster.

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 should the active device become inactive.

Figure 1 on page 7 shows the topology used in this example.

Figure 1: Basic Active/Passive Chassis Clustering Topology on a Pair of High-End SRX Series Devices



Configuration

IN THIS SECTION

- Verification | 15

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*.

To configure this example, perform the following procedures:

Configuring the Control Ports

Step-by-Step Procedure

Select FPC 1/13, because the central point (CP) is always on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the central point (for this example, use the SPC in slot 1). You must enter the operational mode commands on both devices.

NOTE: Control port configuration is required for SRX5600 and SRX5800 Services Gateways. No control port configuration is needed for SRX1400, SRX3400, or SRX3600 devices.

To configure the control port for each device and commit the configuration:

1. Configure the control port for SRX5800-1 (node 0) and commit the configuration.

```
user@SRX5800-1# set chassis cluster control-ports fpc 1 port 0
user@SRX5800-1# set chassis cluster control-ports fpc 13 port 0
user@SRX5800-1# commit
```

2. Configure the control port for SRX5800-2 (node 1) and commit the configuration.

```
user@SRX5800-2# set chassis cluster control-ports fpc 1 port 0
user@SRX5800-2# set chassis cluster control-ports fpc 13 port 0
user@SRX5800-2# commit
```

Enabling Cluster Mode

Step-by-Step Procedure

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.

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. When the system boots, both the nodes come up as a cluster.

> **NOTE**: Since there is only a single cluster on the segments, the example uses cluster ID 1 with Device SRX5800-1 as node 0 and Device SRX5800-2 as node 1.

To set the two devices in cluster mode:

1. Enable cluster mode on the SRX5800-1 (node 0).

```
user@SRX5800-1> set chassis cluster cluster-id 1 node 0 reboot
```

2. Enable cluster mode on the SRX5800-2 (node 1).

```
user@SRX5800-2> set chassis cluster cluster-id 1 node 1 reboot
```

> **BEST PRACTICE**: It is a best practice to place each SRX cluster in its own broadcast domain (BD) with a unique cluster ID for each SRX cluster. This can be achieved with back-to-back connections, or if a switch is used to connect the SRX devices, by assigning a unique VLAN to the switch ports used by each cluster.

If two or more SRX clusters share a broadcast domain you can expect to see heart beat errors increment in the output of a `show chassis cluster information detail` command. This error is the result of a cluster receiving control traffic that is intended for a different cluster. Using back-to-back connections, or a per-cluster VLAN, avoids this issue by placing each cluster in its own BD.

The cluster ID must be the same on both devices that form a cluster, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 0 through 255. Setting a cluster ID to 0 is equivalent to disabling a cluster. A cluster ID greater than 15 can only be set when the fabric and control link interfaces are connected back-to-back or connected on separate VLANs.

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

Configuring Cluster Mode

Step-by-Step Procedure

NOTE: In cluster mode, the cluster is synchronized between the nodes when you execute a `commit` command. All commands are applied to both nodes regardless of which device the command is configured on.

To configure an active/passive chassis cluster on a pair of high-end SRX Series devices:

1. Configure the fabric (data) ports of the cluster that are used to pass real-time objects (RTOs) in active/passive mode. This example uses one of the 1-Gigabit Ethernet ports. Define two fabric interfaces, one on each chassis, to connect together.

```
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 SRX Services Gateway chassis cluster configuration is contained within a single common configuration, use the Junos OS node-specific configuration method called groups to assign some elements of the configuration to a specific member only.

   The `set apply-groups ${node}` command uses the node variable to define how the groups are applied to the nodes. Each node recognizes its number and accepts the configuration accordingly. You must also configure out-of-band management on the fxp0 interface of the SRX5800 using separate IP addresses for the individual control planes of the cluster.

```
user@host# set groups node0
user@host# set groups node1
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 0.0.0.0/0
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 0.0.0.0/0
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 any active/passive mode cluster, only redundancy group 0 and redundancy group 1 need to be configured. This example uses two reth interfaces; both reth interfaces are members of redundancy group 1. Besides redundancy groups, you must also define:

   - Redundant Ethernet Interface count—Configure how many redundant Ethernet interfaces (reth) can possibly be configured so that the system can allocate the appropriate resources for it.

   - Priority for control plane and data plane—Define which device has priority (for chassis cluster, high priority is preferred) for the control plane, and which device is preferred to be active for the data plane.

     NOTE: In active/passive or active/active mode, the control plane (redundancy group 0) can be active on a chassis different from the data plane (redundancy group 1+ and groups) chassis. However, for this example, we recommend having both the control and data plane active on the same chassis member.

```
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 occurs with data plane failover. In case of control plane failover, all the daemons are restarted on the new node. This promotes a seamless transition to the new node without any packet loss.

   Define the following items:

   - Membership information of the member interfaces to the reth interface.

     ```
     user@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
     ```

   - Which redundancy group the reth interface is a member of. For this active/passive example, it is always 1.

     ```
     user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
     user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
     ```

   - The reth interface information such as the IP address of the interface.

     ```
     user@host# set interfaces reth0 unit 0 family inet address 1.1.1.1/24
     user@host# set interfaces reth1 unit 0 family inet address 2.2.2.1/24
     ```

5. Configure the chassis cluster behavior in case of a failure.

   Each interface is configured with a weight value that is deducted from the redundancy group threshold of 255 upon a link loss. The failover threshold is hard coded at 255 and cannot be changed. You can alter an interface link's weights to determine the impact on the chassis failover.

   When a redundancy group threshold reaches 0, that redundancy group fails over to the secondary node.

   Enter the following commands on the SRX5800-1:

   ```
   user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
   user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255
   user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
   ```

```
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
user@SRX5800-1# set chassis cluster control-link-recovery
```

This step completes the chassis cluster configuration part of the active/passive mode example for the SRX5800. The rest of this procedure describes how to configure the zone, virtual router, routing, the EX8208, and the MX480 to complete the deployment scenario.

Configuring Zones, Virtual Routers, and Routes

Step-by-Step Procedure

Configure and connect the reth interfaces to the appropriate zones and virtual routers. For this example, leave the reth0 and reth1 interfaces in the virtual router (default) and the routing table (inet.0), which does not require any additional configuration.

To configure zones, virtual routers, and routes:

1. Configure reth interfaces to the appropriate zones and virtual routers.

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

2. Configure static routes to the other network devices.

```
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
```

Configuring the EX8208

Step-by-Step Procedure

For the EX8208, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive full mesh example for the SRX5800, most notably the VLANs, routing, and interface configuration.

To configure the EX8208:

1. Configure the interfaces.

```
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
```

2. Configure the VLANs.

```
user@host# set vlans SRX5800 vlan-id 50
user@host# set vlans SRX5800 l3-interface vlan.50
```

3. Configure a static route.

```
user@host# set routing-options static route 0.0.0.0/0 next-hop 2.2.2..1/24
```

Configuring the MX240

Step-by-Step Procedure

For the MX240, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800, most notably you must use an integrated routing and bridging (IRB) interface within a virtual switch instance on the switch.

To configure the MX240:

1. Configure the interfaces.

```
user@host# set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family bridge
user@host# set interfaces xe-2/0/0 encapsulation ethernet-bridge unit 0 family bridge
user@host# set interfaces irb unit 0 family inet address 1.1.1.254/24
```

2. Configure the static routes.

```
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
```

3. Configure the bridge domains.

```
user@host# set bridge-domains SRX5800 vlan-id X
user@host# set bridge-domains SRX5800 domain-type bridge routing-interface irb.0
user@host# set bridge-domains SRX5800 domain-type bridge interface xe-1/0/0
user@host# set bridge-domains SRX5800-1 domain-type bridge interface xe-2/0/0
```

Configuring Miscellaneous Settings

Step-by-Step Procedure

This active/passive mode example for the SRX5800 does not describe in detail miscellaneous configurations such as how to configure NAT, security policies, or VPNs. They are essentially the same as they would be for standalone configurations.

However, if you are performing proxy ARP in chassis cluster configurations, you must apply the proxy ARP configurations to the reth interfaces rather than the member interfaces because the RETH interfaces hold the logical configurations.

You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

Verification

IN THIS SECTION

- Verifying Chassis Cluster Status | 16
- Verifying Chassis Cluster Interfaces | 16
- Verifying Chassis Cluster Statistics | 17
- Verifying Chassis Cluster Control Plane Statistics | 19
- Verifying Chassis Cluster Data Plane Statistics | 19
- Verifying Chassis Cluster Redundancy Group Status | 20
- Troubleshooting with Logs | 21

Confirm that the configuration is working properly.

Verifying Chassis Cluster Status

Purpose

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

Action

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

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

Redundancy group: 0 , Failover count: 1
    node0                  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
```

Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

Verifying Chassis Cluster Interfaces

Purpose

Verify information about chassis cluster interfaces.

Action

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

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

```
Control link 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
```

Meaning

The sample output shows each interface's status, weight value, and the redundancy group to which that interface belongs.

Verifying Chassis Cluster Statistics

Purpose

Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of real-time objects (RTOs) sent and received for services.

Action

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

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

Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
```

```
        Probes sent: 258681
        Probes received: 258681
Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          6            0
    Session create                            161          0
    Session close                             148          0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPSec VPN                                 0            0
    Firewall user authentication              0            0
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
    RPC ALG                                   0            0
    RTSP ALG                                  0            0
    RAS ALG                                   0            0
    MAC address learning                      0            0
    GPRS GTP                                  0            0
```

Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

Verifying Chassis Cluster Control Plane Statistics

Purpose

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

Action

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

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

Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258681
        Probes received: 258681
```

Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

Verifying Chassis Cluster Data Plane Statistics

Purpose

Verify information about the number of real-time objects (RTOs) sent and received for services.

Action

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

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

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

Meaning

The sample output shows the number of RTOs sent and received for various services.

Verifying Chassis Cluster Redundancy Group Status

Purpose

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

Action

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

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

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

Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

Troubleshooting with Logs

Purpose

Look at the system log files to identify any chassis cluster issues. You should look at the system log files on both nodes.

Action

From operational mode, enter these `show log` commands.

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

Results

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

```
version x.xx.x
groups {
    node0 {
        system {
            host-name SRX5800-1;
            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 SRX5800-2;
            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 "$1$zTMjraKG$qU8rjxoHzC6Y/WDmYpR9r.";
    }
    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.

RELATED DOCUMENTATION

Chassis Cluster Overview | 3
Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster | 26
Example: Configuring an Active/Active Layer 3 Cluster Deployment | 53

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

IN THIS SECTION

- Requirements | 27
- Overview | 27

- Configuration | 30

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

Requirements

This example uses the following hardware and software components:

- Two Juniper Networks SRX5800 Services Gateways with identical hardware configurations running Junos OS Release 9.6 or later.
- Two Juniper Networks MX480 3D Universal Edge Routers running Junos OS Release 9.6 or later.
- Two Juniper Networks EX8208 Ethernet Switches running Junos OS Release 9.6 or later.

> **NOTE**: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

Before you begin:

- Physically connect the two SRX Services Gateways (back-to-back for the fabric and control ports).

Overview

This example shows how to set up basic active/passive full mesh chassis clustering on a pair of high-end SRX Series devices. Full mesh active/passive clustering allows you to set up an environment that does not have a single point of failure, not only on the SRX Series devices but also on the surrounding network devices. The main difference in the full mesh deployment described in this example and the basic active/passive deployment described in "Example: Configuring an Active/Passive Cluster Deployment" on page 5 is that additional design elements must be considered to accommodate recovery of possible failure scenarios.

Full mesh chassis clustering requires you to configure reth interfaces for each node and ensure that they are connected together by one or more switches. In this scenario, shown in Figure 2 on page 29, there are four reth interfaces (reth0, reth1, reth2, and reth3). A reth interface bundles the two physical interfaces (one from each node) together. A reth interface is part of a redundancy group. Only the member that is on the primary node (active) for the redundancy group is active. The member on the secondary (passive) node is completely inactive, that is, it does not send or receive any traffic.

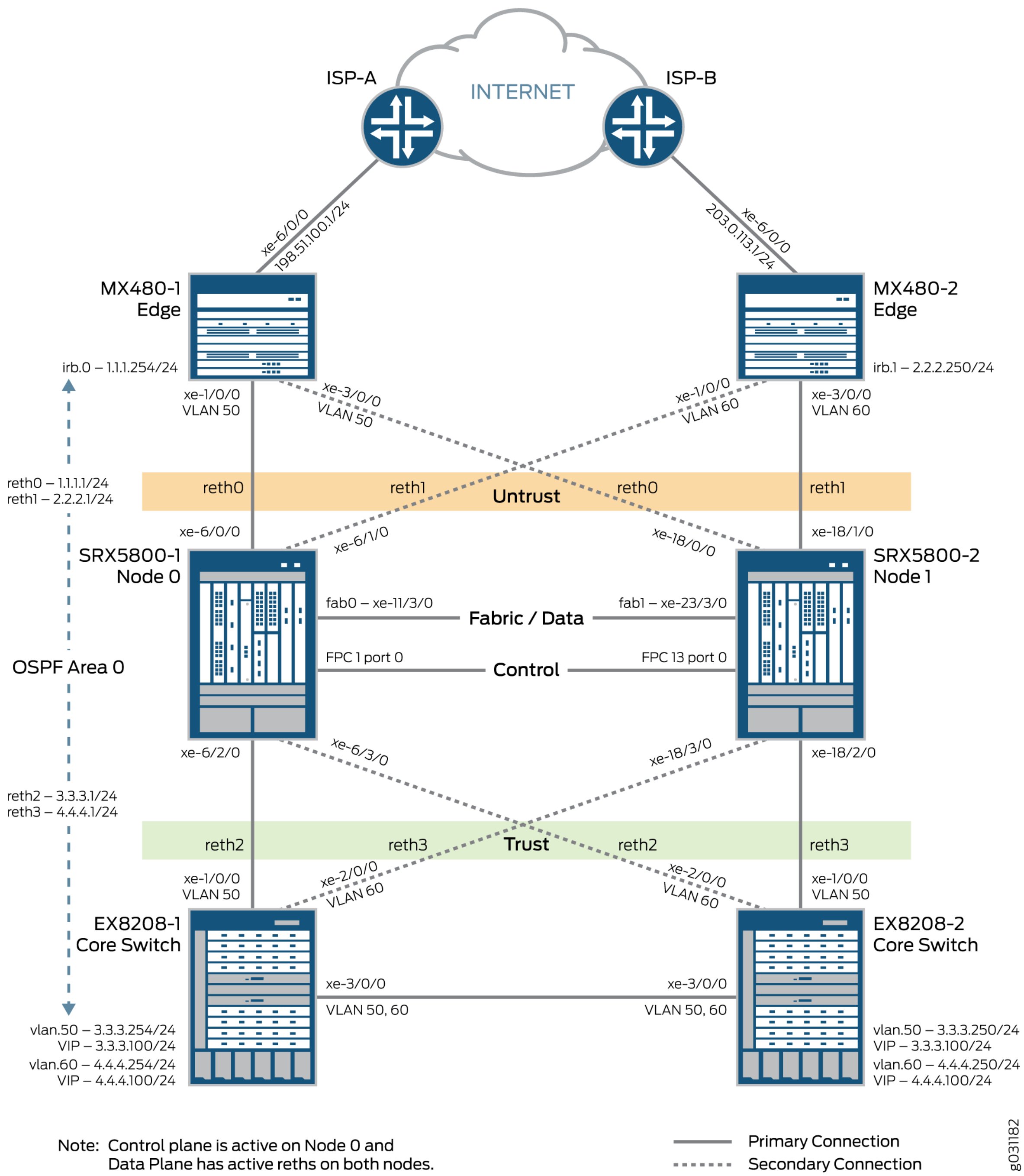
Each reth interface can have one or more logical or subinterfaces (for example, reth 0.0, reth0.1, and so forth). Each must use a different VLAN tag.

The full mesh active/passive chassis cluster consists of two devices:

- One device actively provides routing, firewall, NAT, VPN, and security services, along with maintaining control of the chassis cluster.
- The other device passively maintains its state for cluster failover capabilities should the active device become inactive.

Figure 2 on page 29 shows the topology used in this example.

Figure 2: Full Mesh Active/Passive Chassis Clustering Topology on a Pair of High-End SRX Series Devices



Configuration

IN THIS SECTION

- Verification | 41

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*.

To configure this example, perform the following procedures:

Configuring the Control Ports

Step-by-Step Procedure

Select FPC 1/13, because the central point (CP) is always on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the central point (for this example, use the SPC in slot 1). You must enter the operational mode commands on both devices.

NOTE: Control port configuration is required for SRX5600 and SRX5800 devices. No control port configuration is needed for SRX1400, SRX3400, or SRX3600 devices.

To configure the control port for each device, and commit the configuration:

1. Configure the control port for the SRX5800-1 (node 0) and commit the configuration.

```
user@SRX5800-1# set chassis cluster control-ports fpc 1 port 0
user@SRX5800-1# set chassis cluster control-ports fpc 13 port 0
user@SRX5800-1# commit and-quit
```

2. Configure the control port for the SRX5800-2 (node 1) and commit the configuration.

```
user@SRX5800-2# set chassis cluster control-ports fpc 1 port 0
user@SRX5800-2# set chassis cluster control-ports fpc 13 port 0
user@SRX5800-2# commit and-quit
```

Enabling Cluster Mode

Step-by-Step Procedure

Set the two devices to cluster mode. A reboot is required to enter into cluster mode after the cluster ID and node ID are set. You can cause the system to boot automatically by including the `reboot` parameter in the CLI. You must enter the operational mode commands on both devices. When the system boots, both the nodes come up as a cluster.

> **NOTE**: Since there is only a single cluster on the segments, this example uses cluster ID 1 with Device SRX5800-1 as node 0 and Device SRX5800-2 as node 1.

To set the two devices in cluster mode:

1. Enable cluster mode on the SRX5800-1 (node 0).

```
user@SRX5800-1> set chassis cluster cluster-id 1 node 0 reboot
```

2. Enable cluster mode on the SRX5800-2 (node 1).

```
user@SRX5800-2> set chassis cluster cluster-id 1 node 1 reboot
```

> **NOTE**: If you have multiple SRX device clusters on a single broadcast domain, make sure that you assign different cluster IDs to each cluster to avoid a MAC address conflict.

The cluster ID is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster ID is 1 through 15. Setting a cluster ID to 0 is equivalent to disabling a cluster.

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

Configuring Cluster Mode

Step-by-Step Procedure

> **NOTE**: In cluster mode, the cluster is synchronized between the nodes when you execute a `commit` command. All commands are applied to both nodes regardless of which device the command is configured on.

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

1. Configure the fabric (data) ports of the cluster that are used to pass real-time objects (RTOs) in active/passive mode. Define two fabric interfaces, one on each chassis, to connect together.

```
user@host# set interfaces fab0 fabric-options member-interfaces xe-11/3/0
user@host# set interfaces fab1 fabric-options member-interfaces xe-23/3/0
```

2. Because the SRX Services Gateway chassis cluster configuration is contained within a single common configuration, use the Junos OS node-specific configuration method called groups to assign some elements of the configuration to a specific member only.

   The `set apply-groups ${node}` command uses the node variable to define how the groups are applied to the nodes. Each node recognizes its number and accepts the configuration accordingly. You must also configure out-of-band management on the fxp0 interface of the SRX5800 Services Gateway using separate IP addresses for the individual control planes of the cluster.

> **NOTE**: Configuring the backup router destination address as x.x.x.0/0 is not allowed.

```
user@host# set groups node0
user@host# set groups node1
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 0.0.0.0/0
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 0.0.0.0/0
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 any active/passive mode cluster, only redundancy groups 0 and 1 need to be configured. Use four reth interfaces, all of which are members of redundancy group 1. Besides redundancy groups, you must also define:

   - Redundant Ethernet Interface count—Configure how many redundant Ethernet interfaces (reth) can possibly be configured so that the system can allocate the appropriate resources for it.

   - Priority for control plane and data plane—Define which device has priority (for chassis cluster, high priority is preferred) for the control plane, and which device is preferred to be active for the data plane.

     NOTE: In active/passive or active/active mode, the control plane (redundancy group 0) can be active on a chassis different from the data plane (redundancy group 1+ and groups) chassis. However, for this example, we recommend having both the control and data plane active on the same chassis member. When traffic passes through the fabric link to go to another member node, latency is introduced.

   ```
   user@host# set chassis cluster reth-count 4
   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 occurs with data plane failover. In case of control plane failover, all the daemons are restarted on the new node. Because of this, enabling graceful restart for relevant routing protocols is strongly recommended to avoid losing neighborships with peers. This promotes a seamless transition to the new node without any packet loss.

   Define the following items:

   - Membership information of the member interfaces to the reth interface.

     ```
     user@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-6/2/0 gigether-options redundant-parent reth2
user@host# set interfaces xe-6/3/0 gigether-options redundant-parent reth3
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 xe-18/2/0 gigether-options redundant-parent reth2
user@host# set interfaces xe-18/3/0 gigether-options redundant-parent reth3
```

- Which redundancy group the reth interface is a member of. For this active/passive example, it is always 1.

```
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth2 redundant-ether-options redundancy-group 1
user@host# set interfaces reth3 redundant-ether-options redundancy-group 1
```

- The reth interface information such as the IP address of the interface.

```
user@host# set interfaces reth0 unit 0 family inet address 1.1.1.1/24
user@host# set interfaces reth1 unit 0 family inet address 2.2.2.1/24
user@host# set interfaces reth2 unit 0 family inet address 3.3.3.1/24
user@host# set interfaces reth3 unit 0 family inet address 4.4.4.1/24
```

5. Configure the chassis cluster behavior in case of a failure.

   Each interface is configured with a weight value that is deducted from the redundancy group threshold of 255 upon a link loss. The failover threshold is hard coded at 255 and cannot be changed. You can alter an interface link's weight to determine the impact on the chassis failover.

   When a redundancy group threshold reaches 0, that redundancy group fails over to the secondary node.

   Enter the following commands on the SRX5800-1:

```
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/2/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-6/3/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/2/0 weight 255
```

```
user@SRX5800-1# set chassis cluster redundancy-group 1 interface-monitor xe-18/3/0 weight 255
user@SRX5800-1# set chassis cluster control-link-recovery
```

This step completes the chassis cluster configuration part of the active/passive mode example for the SRX5800. The rest of this procedure describes how to configure the zone, virtual router, routing, the EX8208, and the MX480 to complete the deployment scenario.

Configuring Zones, Routing Options, and Protocols

Step-by-Step Procedure

Configure zones and add the appropriate reth interfaces, and configure OSPF.

To configure zones and OSPF:

1. Configure two zones and add the appropriate reth interfaces .

```
user@host# set security zones security-zone Untrust interfaces reth0.0
user@host# set security zones security-zone Untrust interfaces reth1.0
user@host# set security zones security-zone Trust interfaces reth2.0
user@host# set security zones security-zone Trust interfaces reth3.0
```

2. Permit the appropriate protocols and services to reach interfaces in the Trust zone.

```
user@host# set security zones security-zone Trust host-inbound-traffic protocols ospf
user@host# set security zones security-zone Trust host-inbound-traffic system-services all
```

3. Configure OSPF.

```
user@host# set protocols ospf area 0.0.0.0 interface reth0.0
user@host# set protocols ospf area 0.0.0.0 interface reth1.0
user@host# set protocols ospf area 0.0.0.0 interface reth2.0
user@host# set protocols ospf area 0.0.0.0 interface reth3.0
```

4. Configure a default route and enable graceful restart.

```
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 0.0.0.0/0 next-hop 2.2.2.250
user@host# set routing-options graceful-restart
```

Configuring the EX8208-1

Step-by-Step Procedure

For the EX8208 Ethernet switches, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive full mesh example for the SRX5800 Services Gateway; most notably the VLANs, routing, and interface configuration.

To configure the EX8208-1:

1. Configure the interfaces.

```
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH2
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH3
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching port-mode trunk vlan
members [SRX5800-RETH2 SRX5800-RETH3]
user@host# set interfaces vlan unit 50 family inet address 3.3.3.254/24
user@host# set interfaces vlan unit 60 family inet address 4.4.4.254/24
```

2. Configure VRRP between the two EX switches.

```
user@host# set interfaces vlan unit 50 family inet address 3.3.3.254/24 vrrp-group 1 virtual-
address 3.3.3.100
user@host# set interfaces vlan unit 50 family inet address 3.3.3.254/24 vrrp-group 1 priority
200
user@host# set interfaces vlan unit 50 family inet address 3.3.3.254/24 vrrp-group 1 accept-
data
user@host# set interfaces vlan unit 60 family inet address 4.4.4.254/24 vrrp-group 2 virtual-
address 3.3.3.100
user@host# set interfaces vlan unit 60 family inet address 4.4.4.254/24 vrrp-group 2 priority
100
user@host# set interfaces vlan unit 60 family inet address 4.4.4.254/24 vrrp-group 2 accept-
data
```

3. Configure the VLANs.

```
user@host# set vlans SRX5800-RETH2 vlan-id 50
user@host# set vlans SRX5800-RETH2 l3-interface vlan.50
```

```
user@host# set vlans SRX5800-RETH3 vlan-id 60
user@host# set vlans SRX5800-RETH3 l3-interface vlan.60
```

4. Configure the protocols.

```
user@host# set protocols ospf area 0.0.0.0 interface vlan.50
user@host# set protocols ospf area 0.0.0.0 interface vlan.60
user@host# set protocols rstp interface all
```

5. Configure graceful restart.

```
user@host# set routing-options graceful-restart
```

Configuring the EX8208-2

Step-by-Step Procedure

To configure the EX8208-2:

1. Configure the interfaces.

```
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH2
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH3
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching port-mode trunk vlan
members [SRX5800-RETH2 SRX5800-RETH3]
user@host# set interfaces vlan unit 50 family inet address 3.3.3.250/24
user@host# set interfaces vlan unit 60 family inet address 4.4.4.250/24
```

2. Configure VRRP between the two EX switches.

```
user@host# set interfaces vlan unit 50 family inet address 3.3.3.250/24 vrrp-group 1 virtual-
address 3.3.3.100
user@host# set interfaces vlan unit 50 family inet address 3.3.3.250/24 vrrp-group 1 priority
100
user@host# set interfaces vlan unit 50 family inet address 3.3.3.250/24 vrrp-group 1 accept-
data
```

```
user@host# set interfaces vlan unit 60 family inet address 4.4.4.250/24 vrrp-group 2 virtual-
address 3.3.3.100
user@host# set interfaces vlan unit 60 family inet address 4.4.4.250/24 vrrp-group 2 priority
200
user@host# set interfaces vlan unit 60 family inet address 4.4.4.250/24 vrrp-group 2 accept-
data
```

3. Configure the VLANs.

```
user@host# set vlans SRX5800-RETH2 vlan-id 50
user@host# set vlans SRX5800-RETH2 l3-interface vlan.50
user@host# set vlans SRX5800-RETH3 vlan-id 60
user@host# set vlans SRX5800-RETH3 l3-interface vlan.60
```

4. Configure the protocols.

```
user@host# set protocols ospf area 0.0.0.0 interface vlan.50
user@host# set protocols ospf area 0.0.0.0 interface vlan.60
user@host# set protocols rstp interface all
```

5. Configure graceful restart.

```
user@host# set routing-options graceful-restart
```

Configuring the MX480-1

Step-by-Step Procedure

For the MX480 Edge Routers, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably you must use an IRB interface within a virtual switch instance on the switch.

To configure the MX480-1:

1. Configure the downstream interfaces.

```
user@host# set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family bridge
user@host# set interfaces xe-3/0/0 encapsulation ethernet-bridge unit 0 family bridge
```

2. Configure the upstream interface.

```
user@host# set interfaces xe-6/0/0 unit 0 family inet address 198.51.100.1/24
```

3. Configure the IRB interface.

```
user@host# set interfaces irb unit 0 family inet address 1.1.1.254/24
```

4. Configure a static route and graceful restart.

```
user@host# set routing-options static route 0.0.0.0/0 next-hop 198.51.100.254
user@host# set routing-options graceful-restart
```

5. Configure the bridge domain.

```
user@host# set bridge-domains BD-50 vlan-id 50
user@host# set bridge-domains BD-50 domain-type bridge routing-interface irb.0
user@host# set bridge-domains BD-50 domain-type bridge interface xe-1/0/0
user@host# set bridge-domains BD-50 domain-type bridge interface xe-3/0/0
```

6. Configure OSPF.

```
user@host# set protocols ospf area 0.0.0.0 interface irb.0
```

Configuring the MX480-2

Step-by-Step Procedure

To configure the MX480-2:

1. Configure the downstream interfaces.

```
user@host# set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family bridge
user@host# set interfaces xe-3/0/0 encapsulation ethernet-bridge unit 0 family bridge
```

2. Configure the upstream interface.

```
user@host# set interfaces xe-6/0/0 unit 0 family inet address 203.0.113.1/24
```

3. Configure the IRB interface.

```
user@host# set interfaces irb unit 1 family inet address 2.2.2.250/24
```

4. Configure a static route and graceful restart.

```
user@host# set routing-options static route 0.0.0.0/0 next-hop 203.0.113.254
user@host# set routing-options graceful-restart
```

5. Configure the bridge domain.

```
user@host# set bridge-domains BD-60 vlan-id 60
user@host# set bridge-domains BD-60 domain-type bridge routing-interface irb.1
user@host# set bridge-domains BD-60 domain-type bridge interface xe-1/0/0
user@host# set bridge-domains BD-60 domain-type bridge interface xe-3/0/0
```

6. Configure OSPF.

```
user@host# set protocols ospf area 0.0.0.0 interface irb.1
```

Configuring Miscellaneous Settings

Step-by-Step Procedure

This full mesh chassis clustering example for the SRX5800 does not describe in detail miscellaneous configurations such as how to configure NAT, security policies, or VPNs. They are essentially the same as they would be for standalone configurations.

However, if you are performing proxy ARP in chassis cluster configurations, you must apply the proxy ARP configurations to the reth interfaces rather than the member interfaces because the reth interfaces hold the logical configurations.

You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

Verification

IN THIS SECTION

- Verifying Chassis Cluster Status | 41
- Verifying Chassis Cluster Interfaces | 42
- Verifying Chassis Cluster Statistics | 43
- Verifying Chassis Cluster Control Plane Statistics | 44
- Verifying Chassis Cluster Data Plane Statistics | 45
- Verifying Chassis Cluster Redundancy Group Status | 46
- Troubleshooting with Logs | 47

To confirm that the configuration is working properly, perform these tasks:

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                  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
```

Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

Verifying Chassis Cluster Interfaces

Purpose

Verify information about chassis cluster interfaces.

Action

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

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

Meaning

The sample output shows each interface's status, weight value, and the redundancy group to which that interface belongs.

Verifying Chassis Cluster Statistics

Purpose

Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of real-time objects (RTOs) sent and received for services.

Action

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

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

Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258681
        Probes received: 258681
Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          6            0
    Session create                            161          0
    Session close                             148          0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPSec VPN                                 0            0
    Firewall user authentication              0            0
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
```

```
    RPC ALG                                  0            0
    RTSP ALG                                 0            0
    RAS ALG                                  0            0
    MAC address learning                     0            0
    GPRS GTP                                 0            0
```

Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

Verifying Chassis Cluster Control Plane Statistics

Purpose

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

Action

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

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

Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
```

```
        Probes sent: 258681
        Probes received: 258681
```

Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

Verifying Chassis Cluster Data Plane Statistics

Purpose

Verify information about the number of real-time objects (RTOs) sent and received for services.

Action

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

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

Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          6            0
    Session create                            161          0
    Session close                             148          0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPSec VPN                                 0            0
    Firewall user authentication              0            0
```

```
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
    RPC ALG                                   0            0
    RTSP ALG                                  0            0
    RAS ALG                                   0            0
    MAC address learning                      0            0
    GPRS GTP                                  0            0
```

Meaning

The sample output shows the RTOs sent and received for various services.

Verifying Chassis Cluster Redundancy Group Status

Purpose

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

Action

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

```
{primary:node0}
user@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
```

Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

Troubleshooting with Logs

Purpose

Look at the system logs to identify any chassis cluster issues. You should look at the system log files on both nodes.

Action

From operational mode, enter these `show log` commands.

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

Results

From operational mode, confirm the SRX chassis cluster configuration by entering the `show configuration` command. If the output does not display the intended configuration, repeat the user@host instructions in this example to correct the configuration.

```
version x.xx.x;
groups {
    node0 {
        system {
            host-name SRX5800-1;
            backup-router 10.3.5.254 destination 0.0.0.0/0;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.3.5.1/24;
                    }
                }
            }
        }
    }
```

```
    node1 {
        system {
            host-name SRX5800-2;
            backup-router 10.3.5.254 destination 0.0.0.0/0;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.3.5.2/24;
                    }
                }
            }
        }
    }
}
apply-groups "${node}";
system {
    root-authentication {
        encrypted-password "$ABC123"; ## SECRET-DATA
    }
    name-server {
        8.8.8.8;
    }
    services {
        ssh {
            root-login allow;
        }
        netconf {
            ssh;
        }
        web-management {
            http {
                interface fxp0.0;
            }
        }
    }
}
chassis {
    cluster {
        control-link-recovery;
        reth-count 4;
        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;
                xe-6/2/0 weight 255;
                xe-6/3/0 weight 255;
                xe-18/2/0 weight 255;
                xe-18/3/0 weight 255;
            }
        }
    }
}
interfaces {
    xe-6/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-6/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    xe-6/2/0 {
        gigether-options {
            redundant-parent reth2;
        }
    }
    xe-6/3/0 {
        gigether-options {
            redundant-parent reth3;
        }
```

```
    }
    xe-18/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-18/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    xe-18/2/0 {
        gigether-options {
            redundant-parent reth2;
        }
    }
    xe-18/3/0 {
        gigether-options {
            redundant-parent reth3;
        }
    }
    fab0 {
        fabric-options {
            member-interfaces {
                xe-11/3/0;
            }
        }
    }
    fab1 {
        fabric-options {
            member-interfaces {
                xe-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;
            }
        }
    }
    reth2 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 3.3.3.1/24;
            }
        }
    }
    reth3 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 4.4.4.1/24;
            }
        }
    }
}
routing-options {
    graceful-restart;
    static {
        route 0.0.0.0/0 next-hop 1.1.1.254;
        route 0.0.0.0/0 next-hop 2.2.2.250;
    }
}
security {
    zones {
        security-zone Trust {
```

```
            host-inbound-traffic {
                protocols {
                    ospf;
                }
                system-services {
                    all;
                }
            }
            interfaces {
                reth2.0;
                reth3.0;
            }
        }
        security-zone Untrust {
            interfaces {
                reth0.0;
                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.

RELATED DOCUMENTATION

Chassis Cluster Overview | 3
Example: Configuring an Active/Active Layer 3 Cluster Deployment | 53

Example: Configuring an Active/Active Layer 3 Cluster Deployment

IN THIS SECTION

- Requirements | 53
- Overview | 54
- Configuration | 56

This example shows how to set up basic active/active chassis clustering on high-end SRX Series devices.

Requirements

This example uses the following hardware and software components:

- Two Juniper Networks SRX5800 Services Gateways with identical hardware configurations running Junos OS Release 9.6 or later.
- Two Juniper Networks EX8208 Ethernet Switches running Junos OS Release 9.6 or later.
    - Any EX Series switch can be used here.

Before you begin:

- Physically connect the two SRX Services Gateways (back-to-back for the fabric and control ports).

NOTE: This configuration example has been tested using the software release listed and is assumed to work on all later releases.

Overview

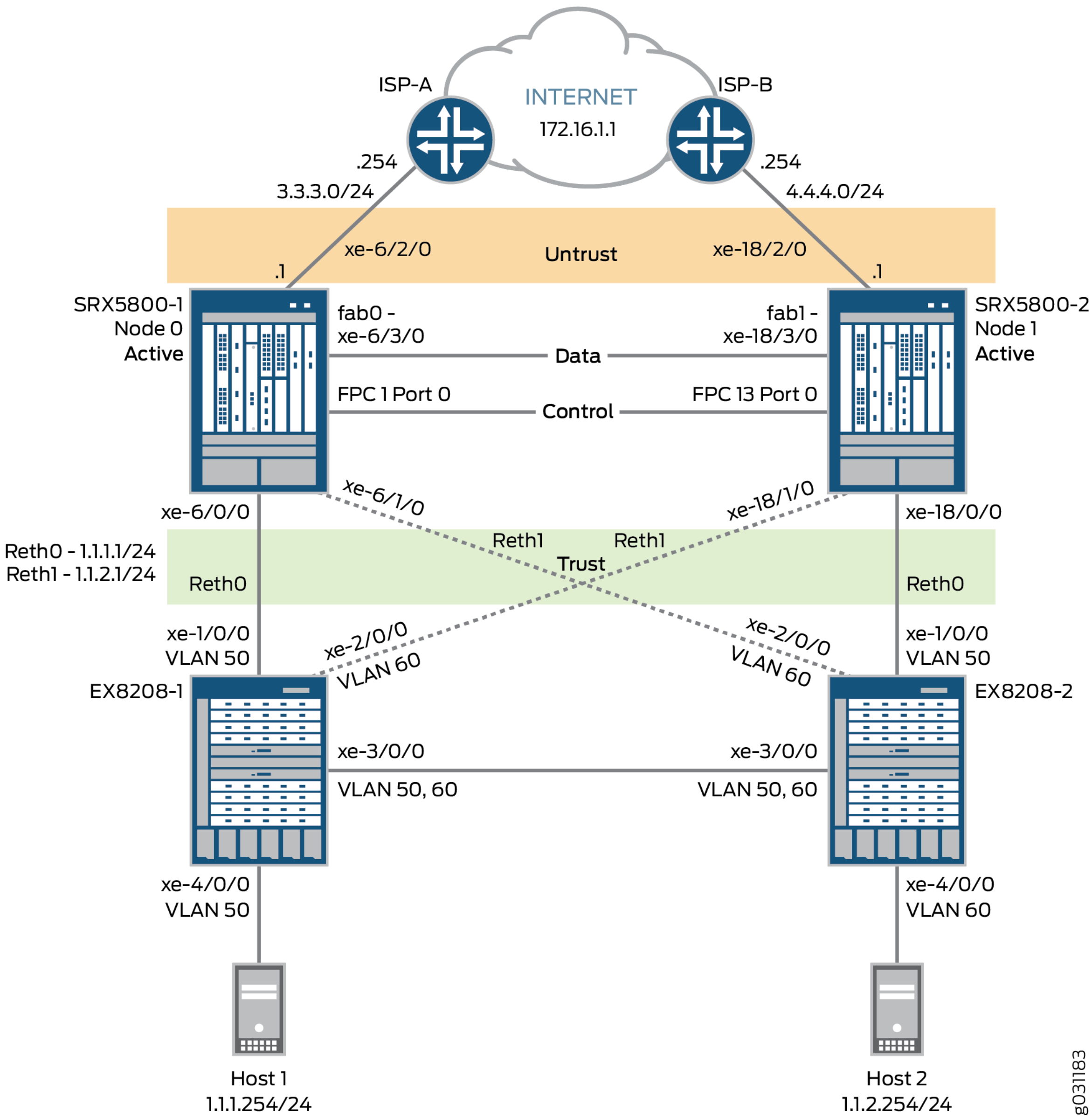
A chassis cluster consists of two SRX Series devices with identical hardware. Active/active clustering on SRX Series devices is supported for those environments that want to maintain traffic on both chassis cluster members whenever possible. In an active/active deployment, only the data plane is in active/active mode; the control plane is in active/passive mode. This allows one control plane to control both chassis members as a single logical device, allowing the control plane to fail over to the other device in case of failure. Having only the data plane in active/active mode allows the data plane to failover independently of the control plane.

Active/active configuration also allows ingress interfaces to be on one cluster device and the egress interfaces to be on the other. When the ingress and egress interfaces are set up on different devices, the data traffic must pass through the data fabric to the other cluster device and out the egress interface.

This active/active chassis clustering example requires you to configure two redundant Ethernet (reth) interfaces—reth0 and reth1—for each node and ensure they are connected together by one or more switches. A reth interface bundles the two physical interfaces (one from each node) together. A reth interface is assigned to a redundancy group.

Figure 3 on page 55 shows the topology used in this example.

Figure 3: Active/Active Layer 3 Chassis Clustering Topology on a Pair of High-End SRX Series Devices



Configuration

IN THIS SECTION

- Verification | 63

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*.

To configure this example, perform the following procedures:

Configuring the Control Ports

Step-by-Step Procedure

Configure the control port for each device.

Select FPC 1 and FPC 13, because the central point (CP) is always on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the central point (this example uses the SPC in slot 1).

> **NOTE**: Control port configuration is required only for SRX5600 and SRX5800 devices. No control port configuration is needed for SRX1400, SRX3400, and SRX3600 devices because they use a fixed control port.

1. Configure the control ports and commit the configuration:

```
user@host# set chassis cluster control-ports fpc 1 port 0
user@host# set chassis cluster control-ports fpc 13 port 0
user@host# commit and-quit
```

Enabling Cluster Mode

Step-by-Step Procedure

Assign a cluster ID and node ID to each device.

Set the two devices to cluster mode by adding a cluster ID and node ID on each and rebooting. You can configure the system to boot automatically by including the `reboot` parameter in the set command.

> **NOTE**: Since there is only a single cluster on the segments, this example uses cluster ID 1 with Device SRX5800-1 as node 0 and Device SRX5800-2 as node 1.

To set the two devices in cluster mode:

1. Enable cluster mode on SRX5800-1 (node 0).

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

2. Enable cluster mode on SRX5800-2 (node 1).

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

> **NOTE**: If you have multiple SRX device clusters on a single broadcast domain, make sure that you assign different cluster IDs to each cluster to avoid MAC address conflicts.

When the system reboots, the nodes come up as a cluster. From this point forward, configuration of the cluster is synchronized between the node members, and the two separate devices function as one device.

Configuring Cluster Parameters

Step-by-Step Procedure

> **NOTE**: In cluster mode, all commands and configuration are applied to both nodes.

To configure chassis cluster settings:

1. Configure a fabric (data) port on each device to enable traffic to pass from one device to the other for cases where traffic arrives on an ingress interface on one node but leaves on another node.

NOTE: A 10-Gigabit Ethernet connection is recommended for active/active deployments.

```
user@host# set interfaces fab0 fabric-options member-interfaces xe-6/3/0
user@host# set interfaces fab1 fabric-options member-interfaces xe-18/3/0
```

2. Configure each device's fxp0 interface for out-of-band management. Assign a separate IP address for each device (control plane) of the cluster.

   Because the SRX Services Gateway chassis cluster configuration is contained within a single common configuration, to assign some elements of the configuration to a specific member only, use the Junos OS node-specific configuration method called groups. The `set apply-groups ${node}` command uses the node variable to define how the groups are applied to the nodes. Each node recognizes its number and accepts the configuration accordingly.

```
user@host# set groups node0 system host-name SRX5800-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24
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 apply-groups ${node}
```

3. Configure redundancy groups for chassis clustering.

   Each node has interfaces in a redundancy group. Redundancy group 0 controls the control plane, it defines which node will be the primary. Redundancy group 1+ controls the data plane and includes the data plane ports. This active/active clustering mode example uses 2 reth interfaces with redundancy groups 0, 1, and 2.

   As part of redundancy group configuration, you must also define the priority for control plane and data plane—which device is preferred for the control plane, and which device is preferred for the data plane. (For chassis clustering, higher priority is preferred.)

NOTE: The control plane (redundancy group 0) and data plane (redundancy group 1+) can each be active on a different chassis. However, for this example, we recommend having both the control and data plane active on the same chassis member. Redundancy group 0 (RG0) and redundancy group 1 (RG1) default to the active state on node 0, whereas, redundancy group 2 (RG2) defaults to active state on node 1.

```
user@host# set chassis cluster redundancy-group 0 node 0 priority 129
user@host# set chassis cluster redundancy-group 0 node 1 priority 128
user@host# set chassis cluster redundancy-group 1 node 0 priority 129
user@host# set chassis cluster redundancy-group 1 node 1 priority 128
user@host# set chassis cluster redundancy-group 2 node 0 priority 128
user@host# set chassis cluster redundancy-group 2 node 1 priority 129
```

4. Configure the data interfaces on the platform so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly.

   Define the following items:

   - The maximum number of reth interfaces for the cluster, so that the system can allocate the appropriate resources for them.

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

   - The reth interface information such as the IP address of the interface.

   ```
   user@host# set interfaces reth0 unit 0 family inet address 1.1.1.1/24
   user@host# set interfaces reth1 unit 0 family inet address 1.1.2.1/24
   ```

   - Membership information of the member interfaces to reth interfaces.

   ```
   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
   ```

   - The mapping of reth interfaces to redundancy groups.

   ```
   user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
   user@host# set interfaces reth1 redundant-ether-options redundancy-group 2
   ```

5. Configure the behavior in case of a failure.

   Each interface is configured with a weight value that is deducted from the redundancy group threshold of 255 upon a link loss. When a redundancy group threshold reaches 0, that redundancy group fails over to the secondary node.

   NOTE: If the `control-link-recovery` feature is not enabled, a manual reboot is required to bring the secondary node back into sync with the primary node.

   ```
   user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
   user@host# set chassis cluster redundancy-group 2 interface-monitor xe-6/1/0 weight 255
   user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
   user@host# set chassis cluster redundancy-group 2 interface-monitor xe-18/1/0 weight 255
   user@host# set chassis cluster control-link-recovery
   ```

   NOTE: Individual VLANs on an interface are not monitored. Only interfaces as a whole are monitored.

   This step completes the chassis cluster configuration.

6. Configure other interfaces that do not belong to the reth interfaces. These are the upstream interfaces towards the ISPs.

   ```
   user@host# set interface xe-6/2/0 unit 0 family inet address 3.3.3.1/24
   user@host# set interface xe-18/2/0 unit 0 family inet address 4.4.4.1/24
   ```

   The following sections describe how to configure zones, security policies, NAT, routing, and the EX8208 Core Switches to complete the deployment scenario.

Configuring Zones, Policies, NAT, and Routes

Step-by-Step Procedure

Configure and connect the reth interfaces to the appropriate zones and define a security policy that permits outbound traffic. Also, for this example we will use a default route and NAT to enable end hosts to reach the Internet.

To configure zones, policies, NAT, and routes:

1. Assign the interfaces to the appropriate zones.

```
user@host# set security zones security-zone trust interfaces reth0.0
user@host# set security zones security-zone trust interfaces reth1.0
user@host# set security zones security-zone untrust interfaces xe-6/2/0.0
user@host# set security zones security-zone untrust interfaces xe-18/2/0.0
```

2. Configure a policy to allow traffic from the hosts in the trust zone to the Internet.

```
user@host# set security policies from-zone trust to-zone untrust policy allow match source-
address any
user@host# set security policies from-zone trust to-zone untrust policy allow match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy allow match
application any
user@host# set security policies from-zone trust to-zone untrust policy allow then permit
```

3. Configure source NAT for outbound traffic.

```
user@host# set security nat source rule-set internet from zone trust
user@host# set security nat source rule-set internet to zone untrust
user@host# set security nat source rule-set internet rule rule1 match source-address
1.1.0.0/16
user@host# set security nat source rule-set internet rule rule1 then source-nat interface
```

4. Define a default static route to enable hosts to reach the Internet.

```
user@host# set routing-options static route 0.0.0.0/0 next-hop 3.3.3.254
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 4.4.4.254 preference
7
```

Configuring the EX8208-1

Step-by-Step Procedure

For the EX8208, the following commands provide configuration only as it pertains to this active/active example for the SRX5800, most notably the VLANs, routing, and interface configuration.

1. Configure the interfaces.

```
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH0
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH1
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching port-mode trunk vlan
members [SRX5800-RETH1 SRX5800-RETH0]
user@host# set interfaces xe-4/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH0
```

NOTE: The end host is sending untagged traffic.

2. Configure the VLANs.

```
user@host# set vlans SRX5800-RETH0 vlan-id 50
user@host# set vlans SRX5800-RETH1 vlan-id 60
```

3. Enable RSTP.

```
user@host# set protocols rstp interface all
```

NOTE: In this example, RSTP is not strictly required as there is no Layer 2 loop. However, a typical environment would likely have more switches, which would require the protocol to be enabled.

Configuring the EX8208-2

Step-by-Step Procedure

To configure the EX8208-2:

1. Configure the interfaces.

```
user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH0
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH1
user@host# set interfaces xe-3/0/0 unit 0 family ethernet-switching port-mode trunk vlan
members [SRX5800-RETH1 SRX5800-RETH0]
user@host# set interfaces xe-4/0/0 unit 0 family ethernet-switching port-mode access vlan
members SRX5800-RETH1
```

NOTE: The end host is sending untagged traffic.

2. Configure the VLANs.

```
user@host# set vlans SRX5800-RETH0 vlan-id 50
user@host# set vlans SRX5800-RETH1 vlan-id 60
```

3. Enable RSTP.

```
user@host# set protocols rstp interface all
```

NOTE: In this example, RSTP is not strictly required as there is no Layer 2 loop. However, a typical environment would likely have more switches, which would require the protocol to be enabled.

Verification

IN THIS SECTION

- Verifying Chassis Cluster Status | 64
- Verifying Chassis Cluster Interfaces | 65
- Verifying Chassis Cluster Statistics | 66

- Verifying Chassis Cluster Control Plane Statistics | 68
- Verifying Chassis Cluster Data Plane Statistics | 68
- Verifying Chassis Cluster Redundancy Group Status | 69
- Troubleshooting with Logs | 70

Confirm that the configuration is working properly.

Verifying Chassis Cluster Status

Purpose

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

Action

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

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

Monitor Failure codes:
    CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  129      primary        no      no       None
node1  128      secondary      no      no       None

Redundancy group: 1 , Failover count: 1
```

```
node0  129      primary       no      no       None
node1  128      secondary     no      no       None

Redundancy group: 2 , Failover count: 0
node0  128      secondary     no      no       None
node1  129      primary       no      no       None
```

Meaning

The sample output shows the status of the primary and secondary nodes and that there are no manual fail overs.

Verifying Chassis Cluster Interfaces

Purpose

Verify information about chassis cluster interfaces.

Action

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

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

Control link status: Up

Control interfaces:
    Index   Interface   Monitored-Status   Internal-SA
    0       fxp1        Up                 Disabled

Fabric link status: Up

Fabric interfaces:
    Name    Child-interface    Status
                               (Physical/Monitored)
    fab0    xe-6/3/0           Up   / Up
    fab0
    fab1    xe-18/3/0          Up   / Up
    fab1
```

```
Redundant-ethernet Information:
    Name         Status      Redundancy-group
    reth0        Up          1
    reth1        Up          2

Redundant-pseudo-interface Information:
    Name         Status      Redundancy-group
    lo0          Up          0

Interface Monitoring:
    Interface         Weight    Status    Redundancy-group
    ge-6/0/0           255       Up        1
    ge-18/0/0          255       Up        1
    ge-6/1/0           255       Up        2
    ge-18/1/0          255       Up        2
```

Meaning

The sample output provides status information about the control and fabric links. It also shows each reth interface's status, weight value, and redundancy group.

Verifying Chassis Cluster Statistics

Purpose

Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of real-time objects (RTOs) sent and received for services.

Action

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

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

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

```
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258681
        Probes received: 258681
Services Synchronized:
    Service name                              RTOs sent    RTOs received
    Translation context                       0            0
    Incoming NAT                              0            0
    Resource manager                          6            0
    Session create                            161          0
    Session close                             148          0
    Session change                            0            0
    Gate create                               0            0
    Session ageout refresh requests           0            0
    Session ageout refresh replies            0            0
    IPSec VPN                                 0            0
    Firewall user authentication              0            0
    MGCP ALG                                  0            0
    H323 ALG                                  0            0
    SIP ALG                                   0            0
    SCCP ALG                                  0            0
    PPTP ALG                                  0            0
    RPC ALG                                   0            0
    RTSP ALG                                  0            0
    RAS ALG                                   0            0
    MAC address learning                      0            0
    GPRS GTP                                  0            0
```

Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

Verifying Chassis Cluster Control Plane Statistics

Purpose

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

Action

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

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

Control link statistics:
    Control link 0:
        Heartbeat packets sent: 258689
        Heartbeat packets received: 258684
        Heartbeat packets errors: 0
Fabric link statistics:
    Child link 0
        Probes sent: 258681
        Probes received: 258681
```

Meaning

Use the sample output to:

- Verify that the **Heartbeat packets sent** is incrementing.
- Verify that the **Heartbeat packets received** is a number close to the number of **Heartbeats packets sent**.
- Verify that the **Heartbeats packets errors** is zero.

This verifies that the heartbeat packets are being transmitted and received without errors.

Verifying Chassis Cluster Data Plane Statistics

Purpose

Verify information about the number of real-time objects (RTOs) sent and received for services.

Action

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

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

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

Meaning

The sample output shows the number of RTOs sent and received for the various services.

Verifying Chassis Cluster Redundancy Group Status

Purpose

Verify the status of a redundancy group.

Action

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

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

Monitor Failure codes:
    CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 1 , Failover count: 1
node0  129      primary        no      no       None
node1  128      secondary      no      no       None
```

Meaning

The sample output shows that redundancy group 1 is functioning normally, with no preemptions, manual fail overs, or other failures.

Troubleshooting with Logs

Purpose

Look at the system log files to identify any chassis cluster issues. You should look at the system log files on both nodes.

Action

From operational mode, enter these `show log` commands.

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

Results

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

```
groups {
    node0 {
        system {
            host-name SRX5800-1;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.3.5.1/24;
                    }
                }
            }
        }
    }
    node1 {
        system {
            host-name SRX5800-2;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.3.5.2/24;
```

```
                    }
                }
            }
        }
    }
}
apply-groups "${node}";
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-18/0/0 weight 255;
            }
        }
        redundancy-group 2 {
            node 0 priority 128;
            node 1 priority 129;
            interface-monitor {
                xe-6/1/0 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-6/2/0 {
        unit 0 {
            family inet {
                address 3.3.3.1/24;
            }
        }
    }
    xe-18/0/0 {
        gigether-options {
            redundant-parent reth0;
        }
    }
    xe-18/1/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    xe-18/2/0 {
        unit 0 {
            family inet {
                address 4.4.4.1/24;
            }
        }
    }
    fab0 {
        fabric-options {
            member-interfaces {
                xe-6/3/0;
            }
        }
    }
    fab1 {
        fabric-options {
            member-interfaces {
                xe-18/3/0;
            }
        }
    }
```

```
    reth0 {
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 0 {
            family inet {
                address 1.1.1.1/24;
            }
        }
    }
    reth1 {
        redundant-ether-options {
            redundancy-group 2;
        }
        unit 0 {
            family inet {
                address 1.1.2.1/24;
            }
        }
    }
}
routing-options {
    static {
        route 0.0.0.0/0 {
            next-hop 3.3.3.254;
            qualified-next-hop 4.4.4.254 {
                preference 7;
            }
        }
    }
}
security {
    nat {
        source {
            rule-set internet {
                from zone trust;
                to zone untrust;
                rule rule1 {
                    match {
                        source-address 1.1.0.0/16;
                    }
                    then {
                        source-nat {
```

```
                            interface;
                        }
                    }
                }
            }
        }
    }
    policies {
        from-zone trust to-zone untrust {
            policy allow {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
    }
    zones {
        security-zone trust {
            interfaces {
                reth0.0;
                reth1.0;
            }
        }
        security-zone untrust {
            interfaces {
                xe-6/2/0.0;
                xe-18/2/0.0;
            }
        }
    }
}
```

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

RELATED DOCUMENTATION

Chassis Cluster Overview | 3

Example: Configuring an Active/Passive Cluster Deployment | 5
Example: Configuring an SRX Series Services Gateway as a Full Mesh Chassis Cluster | 26